Efficient numerical methods for the instationary solution of laminar reacting gas flow problems

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties, in het openbaar te verdedigen
op vrijdag 13 februari 2009 om 12:30 uur

door

Sander VAN VELDHUIZEN
wiskundig ingenieur

geboren te Rotterdam

Dit proefschrift is goedgekeurd door de promotoren:
prof.dr.ir. C. Vuik
prof.dr.ir. C.R. Kleijn

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
prof.dr.ir. C. Vuik,	Technische Universiteit Delft, promotor
prof.dr.ir. C.R. Kleijn,	Technische Universiteit Delft, promotor
prof.dr. R.J. Kee,	Colorado School of Mines
prof.dr. C. Cavallotti,	Politecnico di Milano
prof.dr. J.G. Verwer,	Universiteit of Amsterdam
prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft
prof.dr.ir. B.J. Boersma	Technische Universiteit Delft

Efficient numerical methods for the instationary solution of laminar reacting gas flow problems.
Dissertation at Delft University of Technology.

TU Delft

The work described in this thesis was financially supported by the Delft Centre for Computational Science and Engineering.

ISBN 978-90-9023967-5

Summary

Efficient numerical methods for the instationary solution of laminar reacting gas flow problems

Sander van Veldhuizen

Production processes of high-purity, high performance solid materials in the form of a thin solid film or a powder are of significant importance in various industries, such as the fabrication of micro-electronics, optical and mechanical coatings, and solar cells. Numerous techniques to produce these layers are available, e.g. sputtering, evaporation and Chemical Vapor Deposition (CVD). CVD distinguishes itself by involving chemistry in the process, whereby its greatest advantage is the capability to deposit layers of uniform thickness on highly irregularly shaped surfaces.

Numerical simulations are widely used to design CVD reactors and to optimize the process itself. Over the last decades many researchers have been developing mathematical models to describe the various physical and chemical processes in a CVD reactor. For gas flow with heat and mass transfer in CVD reactors, models based on continuum equations are generally used, having the advantage of being applicable to a wide range of reactor geometries.

Up till recent times, the total process times were large compared to the transient start up and shut-down cycli, such that it was sufficient to perform steady state simulations of these processes. However, with the deposited films getting thinner and thinner, process times are reduced and transient times become more important. Further, besides the classical (steady state) CVD processes, more attention is going to inherently transient processes such as Atomic Layer Deposition and Rapid Thermal CVD.

At the same time, many research groups have been developing computer codes to compute steady state solutions of the traditional processes. The emphasis has always been on the modeling and validition of these models. Much less attention is paid to the computational efficiency of the codes.

However, the solution of the involved mathemical equations is difficult, due to the stiffness caused by the modeled chemistry. Many commercial CFD codes, which are sometimes tailored for these applications, therefore have often great problems to compute the solution. Solutions computed by various codes have been reported to differ a lot and the computational times needed to find solutions are generally excessive. Where most commercial CFD codes already have problems to compute the steady state solution, similar or worser problems are expected for time accurate simuations.

Equivalent difficulties on the numerical modeling of laminar reacting gas flows are found in other applications. Examples are the numerical modeling of laminar combustion and Solid Oxide Fuel Cells. The mathematical models describing the physics involved in all these applications are constantly under development, and create a need for efficient solvers capable of rapid and robust solution of the model equations.

In this thesis a rigorous mathematical approach has been applied to these problems, with the aim to reduce computational times. Besides stability issues due to the stiff reaction term, non-negativity of the species concentrations is very important to enable stable time integration. To fulfill this extra constraint is in general very difficult, and for unconditionally stable higher order time integration impossible. The discretization techniques proposed in this thesis are non-negativity conserving on the level of spatial discretization, time integration and iterative solvers.

Positive spatial discretization techniques are widely known. However, at the reacting boundaries we proposed a discretization that conserves non-negativity, where straightforward techniques lack that property. For higher order time integration it appears that for reacting gas flow problems this property is impossible to fulfill. Of course, the inherent stiffness of the involved chemistry causes parts of the equations to be integrated implicitly. Consequently, we use the implicit Euler Backward time integration method, which is proven to be unconditionally positive and unconditionally stable.

On the level of nonlinear solvers Projected Newton methods are introduced. In the field of constrained optimization such techniques are widely known, but they are unknown in the field of PDEs and reacting flows. To gain computational efficiency, Krylov Subspace methods are used to approximate the solution of the interior linear algebra problem. The stiff reaction terms in the transport equations cause the linear systems to be ill-conditioned, such that inaccurate solution and slow convergence of these methods are observed. This problem is tackled by incorperating effective preconditioning techniques. Various techniques are reviewed and adapted to make them suitable for the applications considered in this thesis.

Choosing the best preconditioners combined with our Projected Newton methods enables us to perform instationary, multi-dimensional gas flow simulations with multi-species, multi-reaction CVD chemistry from inflow conditions until steady state in a computationally efficient way.

Samenvatting

Efficiënte numerieke methoden voor de instationare oplossing van reagerende laminaire stromingen

Sander van Veldhuizen

De productie processen van geavanceerde en zuivere materialen in de vorm van een poeder of een dunne film vorm zijn onmisbaar in verschillende industrieën, zoals in het fabricageproces van micro-electronica, optische en mechanische coatings, en zonnecellen. Er zijn verschillende technieken om deze films, of poeders, te produceren. Voorbeelden zijn sputtering, opdamping en Chemical Vapor Deposition (CVD). De chemische reacties in CVD onderscheiden deze productietechniek van de overige. Een van de belangrijkste voordelen van CVD is dat films van uniforme dikte kunnen worden gedeponeerd op onregelmatige oppervlakten.

Voor zowel het ontwerp als de optimalisatie van CVD reactoren en processen zijn numerieke simulaties een belangrijk stuk gereedschap. In de laatste decennia zijn veel wiskundige modellen ontwikkeld om de fysische en chemische processen in een CVD reactor te beschrijven. Voor gasstromen met warmte en massa transport worden in het algemeen modellen gebaseerd op continue beschrijvingen gebruikt. Deze hebben het voordeel dat ze toepasbaar op uiteenlopende reactoren.

De totale duur van CVD processen uit het verleden is lang in vergelijking met de transiente opstart en afsluit-cycli. In dit geval kon worden volstaan met *steady state* simulaties van het CVD proces. Echter, de te deponeren films worden dunner en dunner, zodat de tijdsduur van het CVD proces korter wordt, en transiente verschijnselen steeds belangrijker worden. Ten tweede beweegt de technologie zich meer en meer naar inherent transiente CVD processen, zodat steady state simulaties niet langer zullen voldoen. Twee voorbeelden van transiente CVD technologien zijn *Atomic Layer Deposition* en *Rapid Thermal CVD*.

Door onderzoekers wereldwijd zijn er computercodes ontwikkeld om

de stationaire oplossing te berekenen van 'traditionele' CVD processen. Hierbij ging de meeste aandacht uit naar de modelering en validatie van deze modellen, en minder naar de rekenkundige efficientie van de computer codes. Echter, in het algemeen is het vinden van de oplossing van de onderliggende wiskundige vergelijkingen in deze specifieke toepassing geen eenvoudige taak. De stijve chemie term in de transportvergelijkingen voor de chemische componenten (stofjes) in het gasmengsel maakt deze vergelijkingen moeilijk op te lossen. Commerciële CFD software pakketten, soms specifiek voor deze toepassing geschreven, hebben vaak problemen om de oplossing te berekenen. Naast dat oplossingen berekend door verschillende codes veel van elkaar verschillen, zijn vaak de rekentijden excessief groot.

De meeste commerciële CFD pakketten al problemen hebben met het berekenen van de steady state oplossing, worden dezelfde of ergere moeilijkheden verwacht om tijdsnauwkeurige oplossingen te berekenen.

In de numerieke modellering van laminaire reagerende gasstromingen van gerelateerde toepassingen zoals laminaire verbranding en Solid Oxide Fuel Cells spelen dezelfde moeilijkheden een rol. De onderliggende fysische modellen zijn voortdurend in ontwikkeling, en mede daardoor is er vraag naar een algemene efficiente computer code voor dit type problemen.

Het onderwerp van deze dissertatie is het reduceren van de rekentijden voor instationaire simulaties van laminair reagerende gasstromingen. Naast stabiliteitseisen voor tijdsintegratie, hetgeen belangrijk is door de stijve chemie bronterm in de transportvergelijkingen, is het behoud van niet-negativiteit van chemische concentraties belangrijk. Het is onmogelijk voor onvoorwaardelijke stabiele hogere orde tijdsintegratie methoden te voldoen aan deze positiviteitseis te voldoen. Alle in deze dissertatie voorgestelde oplossingstechnieken zijn positviteit behoudend op zowel het niveau van plaats-discretizatie, tijds-discretizatie en iteratieve solvers.

Projected Newton methoden zijn voorgesteld om de oplossingen op het tijdsniveau positief te houden. Voor 'traditionele' Newton methoden is dit niet noodzakelijk waar, hetgeen met numerieke experimenten is aangetoond. Om een hogere efficiency te halen worden Krylov deelruimte methoden toegepast om de Newton stap te benaderen. De stijve reactietermen zorgen echter voor een slecht-geconditioneerd lineair systeem, met als gevolg onnauwkeurige oplossingen en een trage convergentie. Met effectieve preconditionering technieken is dit probleem opgelost. Verschillende technieken zijn getest en zonodig aangepast voor de huidige toepassing.

Met de Projected Newton solver, gecombineerd met de beste preconditioner, zijn we in staat om instationaire, multi-dimensionale simulaties van gasstromingen met een multi-component, multi-reactie CVD chemie model door te rekenen van de instroom condities tot de stationaire oplossing is bereikt op een rekenkundig efficiente manier.

Acknowledgements

Nine years ago I started to study applied mathematics at the Delft University of Technology. Five years later this study was completed and under supervision of prof. Vuik and prof. Kleijn I started my PhD studies in November 2004. In this four year period of research many people were involved, either from the field of research, or related activities. This part of the thesis is the appropriate place to acknowledge them.

Financial support for this study was provided by the Delft Centre for Computational Science and Engineering. Additionally, thanks are addressed to the Delft University of Technology for sponsoring the thesis printing costs.

First of all, I would like to express my gratitude to my advisors prof. Kees Vuik and prof. Chris Kleijn for their guidance and support in the past four years. I would like to thank them for the many fruitful discussions, their reading and useful criticism on various papers and the draft versions of this thesis.

From the start of this PhD project the glassgroup of TNO Science and Industry, represented by Adriaan Lankhorst and Philip Simons, was involved as well. I like to thank them for the useful input. In particular, many thanks are owed to Philip for performing the computations of the three-dimensional flow problem.

In the fall of 2007 I had the oppertunity to visit prof. Kee's group at the Colorado School of Mines. I am indebted to prof. Bob Kee for giving me this opportunity and hosting me. Special thanks are also owed to Huayang Zhu for the pleasant colloboration.

It was a pleasure to work in the numerical analysis group. I thank all (former) members for creating the typical casual, friendly and pleasant atmosphere. Of course, the coffee breaks with its many (interesting) conversations are unforgettable.

The importance of a skilled secretary is often underestimated. Up till september 2008 Diana Droog has been the secretary of the numerical analysis group. Her professionality was put forward by making burocracy

dissapear. As well, Diana was invaluable in printing the booklets for the PhDays. I like to thank her for all the work, support and, especially, our many chats at the end of the working day.

For the past four years office HB07.250 was, probably, one of the noisiest of the 7th floor. As roommates, Etel Javierre, John Brusche, Tijmen Collignon, Reijer Idema and Ibbi, have contributed to many moments of joy, many interesting discussions and a lot of fun. I truly enjoyed sharing the office with them. Hopefully, in the near future HB 07.250 remains the noisiest, funniest, . . . office of the 7th floor.

Lunch was for a small group within the numerical analysis department an integral part of the working day. The unwritten rule not to talk about work led to many hilarious discussions, of which some I will never forget. Fred, Sander 1, John, Jok, Jelle, Miriam, Tijmen, Reijer, Jennifer, Paulien, Domenico and Duncan, thanks for your company during lunch.

Many new friendships were formed in the Dutch-Flanders numerical analysis community during the PhDays 2005 – PhDays2008. In 2006 and 2007 it was my pleasure to organize this weekend event. Without my co-organizers Arthur van Dam, Tammo Jan Dijkema, Ward van Aerschot, Liesbeth Vanherpe, Yves Frederix, Liesbet Roose and Hendrik Speleers it would have been impossible to make both weekends to a great succes. Thanks for that.

Being the TU Delft and numerical analysis representative in the JM Burgerscentrum PhD contact group was a marvelous experience. The semi-anually meetings were always very enjoyable, as well as the outings. Thanks to all members of the contact group. In 2007 Peter Lucas and I had the privilege to organize the outing, which was, as far as I can remember, very succesful. Thanks are also addressed to Ilse Hoekstijn-Philips for the administrative assistence.

Further, I would like to thank my dearest friends for all the great times besides work. The holidays, dinners, nights out with you gave a welcome relief after work. A special word of thank to my parents for their unconditional support over the last 28 years. Last but not least, my warmest thanks are for my dearest Hoi Wah, whose support was, is and remains invaluable.

Sander van Veldhuizen
Delft, November 2008

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 General Problem Description

Nowadays mathematical modeling is a common resource for the design and/or optimization of industrial processes and equipment. Well known examples are casting of metals, production of ceramics, electrolysis and thin film deposition. In this study we aim to develop efficient numerical software for the simulation of laminar chemically reacting gas flow processes and equipment. The applications considered in this study are Chemical Vapor Deposition and Solid Oxide Fuel Cells.

Most research performed in this study has been devoted to the time accurate simulation of Chemical Vapor Deposition processes. In the last chapter we report on the numerical results with respect to Solid Oxide Fuel Cell modeling.

For both applications finding the solution of the mathematical equations describing the physical processes is generally difficult due to the inherent mathematical stiffness of the equations describing the chemistry. Time scales of the various chemical reactions may differ orders of magnitude from each other depending on the process conditions such as pressure, temperature, species concentrations, flowrate and reactor or fuel cell geometry. Further, time scales of the transport of species and the gas phase chemistry may differ orders of magnitude from each other, and cause stiffness as well. Hence, the presence of stiffness is all around in the reacting flow models.

In this study we assume the mathematical models for the laminar reacting gas flow process to be given. The emphasis is fully on efficient computational methods for the numerical solution of the model equations. For Chemical Vapor Deposition, the computational era started in the early

1980s by the work of Wahl (1977), Jensen & Graves (1983) and Coltrin et al. (1984). The mathematical models for CVD development since then consist of sets of mathematical equations describing the macro- and microscopic physical and chemical processes in the gas phase and at the deposition surface. Ideally, the models are applicable to a wide range of (reactor) geometries and processes.

For the CVD processes considered in this thesis, mathematical models are used that have been developed in the early 1990s, and are currently still in use. These models consist of a set of partial differential equations describing the transport phenomena and production/destruction rates of species due to chemical reactions at the macroscopic scale. Usually, the transport properties apearring in these equations are evaluated through various submodels. This is also the case for the chemical reaction rates. Various codes have been developed over the last decades to compute the *steady state* solution of such CVD models. Examples are the SANDIA codes CHEMKIN and SPIN, see Coltrin et al. (1993), Kee et al. (1989) and Coltrin et al. (1996), Phoenics-CVD of CHAM, see Phoenics-CVD (1995), the CVD modules of Fluent, see Fluent (1995), and the CVDMODEL code developed at the Delft University of Technology, see Kleijn et al. (1989), Kleijn (1991), Kuijlaars et al. (1995) and Kleijn (2000).

However, there is an increasing need to compute time accurate transient solutions of these models. Such simulation results give, for example, insights in start-up and shut-down cycli of CVD processes. Moreover, time dependent simulations are indispensable for inherently transient CVD process such as Atomic Layer Deposition, see for instance Lankhorst et al. (2007), and Rapid Thermal Chemical Vapor Deposition, see for instance Bouteville (2005).

Most commercial Computational Fluid Dynamics (CFD) codes have great problems to combine multi-dimensional CFD modeling and detailed chemistry modeling. Although some commercial CFD codes claim to be able to handle stiff chemistry, no succesfull attempts to model multi-dimensional gas-flow with multi-species, multi-reaction CVD chemistry using commercial CFD codes have been reported in literature.

The numerical stiffness of the discretized transport and chemistry terms leads generally to poor convergence and the obtained results can be unreliable. Geyling (1994) performed a study to the differences in simulation results produced by various CVD-tailored CFD codes. As was reported in Geyling (1994), differences of an order of magitude were observed in species concentrations computed by the various codes. Lastly, the computation times are excessive, in particular for time accurate solutions.

The Solid Oxide Fuel Cells (SOFC's) considered in this thesis are modeled by means of macroscopic continuum models of the composite electrodes. More specifically, the distributed charge-transfer model developed by Zhu & Kee (2008), and extended for so-called segemented-in-series

SOFCs in Kee et al. (2008) is briefly discussed. Over the last years significant progress has been made in the development of numerical models for SIS-cells, see Costamagna et al. (2004) and Haberman & Young (2008), in which fluid and mass transport are coupled with chemical and electrochemical processes. Compared to prior literature, the model used in this thesis makes significant advances in the fundamental representation of chemistry and electrochemistry. As for the used CVD models, most attention in literature was on their validation, and much less on their efficient numerical solution.

Finding time dependent solutions for the mentioned CVD and SOFC models involves a range of difficulties. First of all, the advection-diffusion-reaction equations describing the species transport and chemical interactions are stiffly and nonlinearly coupled through the reaction terms. Due to stability requirements, it is necessary to integrate the stiff reaction terms implicitly. Secondly, the solution represents a set of physical quantities, and must reflect the physical properties of these quantaties. The most important in this respect is the non-negativity of species concentrations. It appears that, in particular, during time integration it is extremely hard to preserve this property. Thirdly, partly or fully implicit time integration implies that per time step one or more nonlinear systems have to be solved. Doing that in an computationally efficient way is not a straightforward task. Further, the usage of approximation techniques for the solution of the involved nonlinear equations does not necessarily guarantee the conservation of physical properties such as non-negativity.

Besides on the number of spatial dimensions and the number of mesh points in each spatial dimension, the number of unknowns in these problems depends also on the number of reactants in the model. For models containing a large number (typically several dozens) of reactants the solution of the resulting nonlinear systems becomes very expensive. Moreover, the conservation of the physical properties of the solution returned from standard nonlinear solver techniques is not guaranteed.

In this study it has been chosen to use Newton-type methods to solve the system of nonlinear algebraic equations. The efficiency of such methods is mainly determined by the computational costs of solving the interior linear algebra problem. Since the linear systems are always large and sparse, iterative methods are ideal candidates to solve these problems. The computational efficiency of these methods is mainly determined by effective preconditing. Of course, the stiffness of the system of PDEs that model the reacting flow can have quite some effect on the convergence behavior and accuracy of iterative solution techniques.

The main difficulty in all laminar reacting gas flow simulations is the solution of the system of advection-diffusion-reaction equations describing the transport and their conversion due to chemical reactions of all species in the gas mixture. Compared to the solution of these stiff systems of transport

equations, the solution of the accomponying hydrodynamics problem is a relatively trivial task. Therefore, the emphasis of this thesis is on the numerical solution of systems of advection-diffusion-reaction equations.

The aims of this study are summarized as follows:

(i) to develop robust and efficient numerical methods to perform time accurate transient simulations;

(ii) guarantee non-negative species concentrations without clipping, and thus conserve mass for all species.

(iii) to study the influence of stiffness on the linear solutions, and develop efficient and robust iterative linear solvers. In particular, efficient preconditioners need to be constructed.

As mentioned before, the major part of this study has been devoted to the time accurate transient simulation of Chemical Vapor Deposition. Consequently, more details on this specific application are provided than on Solid Oxide Fuel Cell modeling.

1.2 Outline of the Thesis

This dissertation is organized as follows.

- The mathematical framework, in which the transport phenomena and homogeneous and heterogeneous chemical reactions in thermal Chemical Vapor Deposition reactors are being described, is presented in Chapter 2. This model is generally applicable to a large variety of thermal Chemical Vapor Deposition processes and reactors, and also to other reacting flow processes such as laminar combustion.

- In Chapter 3, the use of the Method of Lines approach, as well as the method itself, are discussed. In particular, attention is paid to a discretization of the reacting surface boundary condition which preserves positivity.

- In Chapter 4, the notion of positivity is discussed in detail. Positivity of the species concentrations in the gas mixture is of great importance to avoid blow up of the solution. Some mathematical results on this subject are formulated.

- In Chapter 5, a review of various stiff ODE methods from literature is presented. The emphasis is on both positivity and computational efficiency. Details of the implementation are given if necessary. Finally, these ODE methods are applied to a benchmark problem, of which the numerical results are presented.

- Chapter 6 is devoted to the solution of large systems of nonlinear algebraic equations. A review of Inexact Newton methods is presented. The key feature of such solution techniques is that the internal linear systems are solved in an approximated manner. Essential for the performance of these methods is the choice of the forcing term, which determines the accuracy of the linear solver. Further, an extension of Newton methods is proposed, which conserves the positivity property when needed.

- In Chapter 7 suitable iterative solution techniques are discussed for the approximation of the solution of the Newton equation in Newton's method. Various preconditioning techniques are reviewed, or extended to make them suitable for the applications considered in this dissertation. Typically, for these applications the number of unknowns depends on the number of grid points and the number of species. Essential for the performance of iterative solution techniques for linear systems is the ordering of unknowns. Two orderings of unknowns are discussed. Numerical experiments are used to show that one of these orderings is most effective.

- In Chapter 8 the numerical results of transient simulations on Chemical Vapor Deposition are presented. Two chemistry models and two reactors configurations are studied. If possible, the numerical results are benchmarked against results obtained by other well known simulation codes. Transient results are presented for both two and three-dimensional computational domains. In particular, the computational costs are evaluated for the numerical methods proposed in preceeding chapters.

- Chapter 9 is devoted to the numerical modeling of Solid Oxide Fuel Cells. The essential parts of the mathematical model are briefly presented, as well as the numerical results for an illustrative example.

- Finally, in Chapter 10, the results from the previous chapters are reviewed and some general conclusions are formulated.

CHAPTER 2

Chemical Vapor Deposition

Thin solid films are widely used in many technological areas with applications varying from insulating and (semi-)conducting layers in microelectronics and photovoltaics, to optical, mechanical and/or decorative coatings on various materials. The production of such thin layers can be done by various deposition processes, e.g. sputtering, evaporation and Chemical Vapor Deposition (CVD). The fact that it involves chemical reactions clearly distinguishes CVD from the other production technologies, whereby the most important advantage is its capability of depositing films of uniform thickness on highly irregularly shaped surfaces, see for instance Hitchman & Jensen (1993).

Basically, a CVD system is a chemical reactor in which precursor gases containing the atoms to be deposited are introduced, usually diluted in an inert carrier gas. Furthermore, the reactor chamber contains substrates on which the deposition takes place. In this study it is assumed that the energy to drive the (gas phase and surface) reactions is thermal energy, provided by external heat sources.

In numerical simulation the following six steps occuring in every CVD process have to be mathematically modeled:

1. Convective and diffusive transport of reactants from the reactor inlet to the reaction zone within the reactor chamber,

2. Chemical reactions in the gas phase leading to a multitude of new reactive species and byproducts,

3. Diffusive transport of the initial reactants and the reaction products from the homogeneous reactions to the susceptor surface, where they are adsorped on the susceptor surface,

4. Surface diffusion of adsorbed species over the surface and heterogeneous surface reactions catalyzed by the surface, leading to the formation of a solid film,

5. Desorption of gaseous reaction products, and their diffusive transport away from the surface,

6. Convective and/or diffusive transport of reaction products away from the reaction zone to the outlet of the reactor.

For fully heterogeneous CVD processes the second step in the above enumeration does not take place. Steps one to six are illustrated in Figure 2.1.

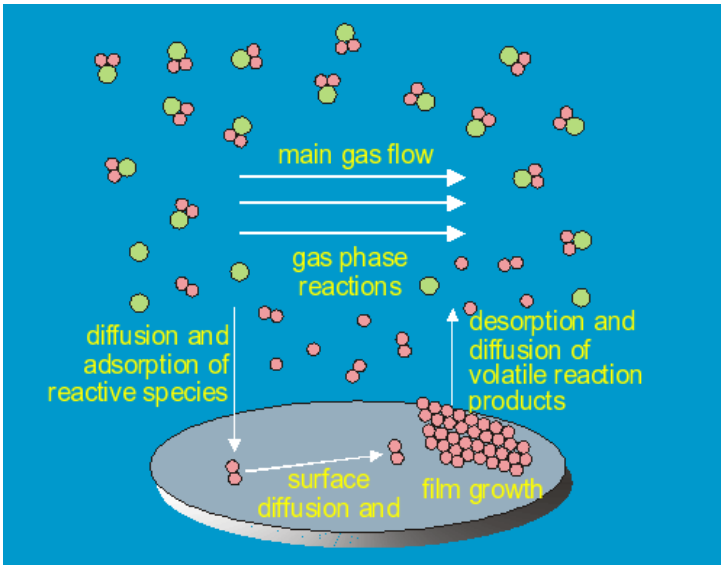


Figure 2.1: Schematic representation of the six basic steps in CVD after Jensen (1988).

2.1 Basic Assumptions

To mathematically model a CVD process, the gas flow, the transport of thermal energy, the transport of species and the chemical reactions in the reactor have to be described. We assume that the gas mixture in the reactor behaves as a continuum, as an ideal gas and in accordance with Newton's law of viscosity. The gas flow in the reactor is assumed to be laminar.

The continuum approach can be safely used when the Knudsen number Kn is below 0.01, see Kleijn (1991). The Knudsen number Kn is the

ratio of the mean free path length ξ of the molecules in the reactor and a characteristic dimension L of the reactor, i.e.,

$$\mathrm{Kn} = \frac{\xi}{L}. \tag{2.1}$$

For pressures larger than 100 Pa and typical reactor dimensions larger than 1 cm it is safe to use the continuum approach, see Kleijn (1991).

2.2 Gas Species Concentrations

Before the partial differential equations of the model are formulated, we introduce the notion of mass fractions and molar fractions. The composition of the N component gas mixture is described in terms of the dimensionless mass fractions, which are defined as

$$\omega_i = \frac{\rho_i}{\rho}, \tag{2.2}$$

where ρ_i is the partial mass density of species i in the gas mixture and ρ the mass density of the gas mixture. The mass density of the gas mixture, defined as

$$\rho = \sum_{i=1}^{N} \rho_i, \tag{2.3}$$

is in general a function of temperature T, pressure P and composition of the gas mixture. From (2.2) and (2.3) if follows that the mass fractions have the property to sum up to one, i.e.,

$$\sum_{i=1}^{N} \omega_i = 1. \tag{2.4}$$

In order to describe the chemical processes in the gas mixture, it is more convenient to describe the gas mixture composition in terms of mole fractions. The mole fraction of species i, denoted as f_i, is the number of moles of species i in a volume divided by the total number of moles in the volume. The mass fractions and molar fractions are related through

$$\omega_i = \frac{f_i m_i}{m}. \tag{2.5}$$

In (2.5) the molar mass of species i is denoted as m_i, whereas m denotes the average molar mass. The latter can be computed from

$$m = \sum_{i=1}^{N} f_i m_i, \tag{2.6}$$

or,

$$m = \left(\sum_{i=1}^{N} \frac{\omega_i}{m_i} \right)^{-1}. \tag{2.7}$$

2.3 Model for Fluid Flow and Heat Transfer

Conservation of total mass, momentum and heat are described respectively by the continuity equation, i.e.,

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}), \tag{2.8}$$

the Navier-Stokes equations, i.e.,

$$\frac{\partial (\rho \mathbf{v})}{\partial t} = -(\nabla \rho \mathbf{v}) \cdot \mathbf{v} + \nabla \cdot \left[\mu \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^{\mathbf{T}} \right) - \frac{\mathbf{2}}{\mathbf{3}} \mu (\nabla \cdot \mathbf{v}) \mathbf{I} \right] - \nabla \mathbf{P} + \rho \mathbf{g}, \tag{2.9}$$

and the transport equation for thermal energy

$$\begin{aligned} c_p \frac{\partial (\rho T)}{\partial t} &= -c_p \nabla \cdot (\rho \mathbf{v} T) + \nabla \cdot (\lambda \nabla T) + \\ &\quad + \nabla \cdot \left(RT \sum_{i=1}^{N} \frac{\mathbb{D}_i^T}{M_i} \frac{\nabla f_i}{f_i} \right) + \sum_{i=1}^{N} \frac{H_i}{m_i} \nabla \cdot \mathbf{j}_i \\ &\quad - \sum_{i=1}^{N} \sum_{k=1}^{K} H_i \nu_{ik} R_k^g, \end{aligned} \tag{2.10}$$

Here, ρ denotes the gas mixture density, $\mathbf{v}$ the mass averaged velocity vector, μ the viscosity, $\mathbf{I}$ the unit tensor, $\mathbf{g}$ the vector of gravitational acceleration, c_p specific heat, λ the thermal conductivity, R the universal gas constant, $\mathbb{D}_i^T$ the thermal diffusion coefficient of species i, H_i the molar enthalpy of species i, and $\mathbf{j}_i$ the diffusive mass flux. The stoichiometric coefficient of the i^{th} species in the k^{th} gas-phase reaction with net molar reaction rate R_k^g is denoted as ν_{ik}. In Section 2.4 the exact definitions of the diffusive mass flux, thermal diffusion coefficients and net molar reaction rate are presented.

Under the assumption that the gas mixture behaves as an ideal gas, the system of equations (2.8) - (2.10) is closed by the ideal gas law

$$Pm = \rho RT. \tag{2.11}$$

The third term on the right-hand side of (2.10) is due to the *Dufour effect*, or diffusion-thermo effect. The Dufour effect is the "inverse" process of thermal diffusion, which is described in Section 2.4.2. The Dufour effect causes an energy flux due to concentration gradients in the gas mixture. The fourth term on the right represents the transport of heat associated

with the inter-diffusion of the chemical species. Both terms have found to be not important in CVD, see Kleijn (1995).

The fifth term on the right-hand side describes the consumption and production of heat due to the chemical reactions. For most CVD systems, especially when the reactants are highly diluted in an inert carrier gas, the heat of reactions has a negligible influence on the gas temperature distribution. For such systems, the computation of the laminar flow and the temperature field is a relatively trivial task. The difficulty, however, lies in solving the set of highly nonlinear and strongly coupled species equations.

2.4 Model for Species Transport and Chemical Reactions in the Gas Phase

The transport of species is formulated in terms of mass fractions and mass fluxes. The convective mass flux of species i is $\rho\omega_i\mathbf{v}$. The mass diffusion flux $\mathbf{j}_i$ of species i is composed of ordinary diffusion $\mathbf{j}_i^C$, which is a result of concentration gradients in the gas mixture, and thermal diffusion $\mathbf{j}_i^T$, which is the result of a temperature gradient, i.e.,

$$\mathbf{j}_i = \mathbf{j}_i^C + \mathbf{j}_i^T. \tag{2.12}$$

2.4.1 Ordinary Diffusion

For a multicomponent gas mixture there are several approaches to model ordinary diffusion. The Stefan-Maxwell equations give an exact, general expression for ordinary diffusion fluxes, see Kee et al. (2003). An approximate approach, which is used in this study, is to model the ordinary diffusive mass fluxes according to Fick's Law, making use of effective multicomponent diffusion coefficients. The ordinary diffusion flux is then computed as

$$\mathbf{j}_i^C = \rho\mathbb{D}_i'\nabla\omega_i, \tag{2.13}$$

with effective multicomponent diffusion coefficient

$$\mathbb{D}_i' = (1 - f_i)\left(\sum_{j=1, j\neq i}^{N} \frac{f_j}{D_{ij}}\right)^{-1}. \tag{2.14}$$

For gas mixtures, in which the reactants are highly diluted in a carrier gas, i.e.,

$$f_1, \ldots, f_{N-1} \ll 1, \tag{2.15}$$

the approximate approach of equations (2.13) and (2.14) is identical to the exact Stefan-Maxwell approach.

2.4.2 Thermal diffusion

The Soret effect, or the effect of thermal diffusion, separates an initially homogeneous gas mixture under the influence of a temperature gradient. Compared with ordinary diffusion the Soret effect is in general small. However, for CVD systems in which large temperature gradients are present this effect may be important. A cold wall CVD reactor is an example where large temperature gradients, up to several hundreds Kelvin per centimeter, can be found. In general, thermal diffusion causes a movement of relatively large and heavy molecules to 'colder' regions and a movement of relatively smaller and lighter molecules to hotter parts of the reactor chamber.

The thermal diffusive mass flux is modeled as

$$\mathbf{j}_i^T = -\mathbb{D}_i^T \nabla(\ln T). \tag{2.16}$$

In expression (2.16) $\mathbb{D}_i^T$ is the multi-component thermal diffusion coefficient for species i. In general, $\mathbb{D}_i^T$ is a function of the temperature T and the composition of the gas mixture, but independent of the pressure. For large and heavy molecules we have that $\mathbb{D}_i^T > 0$, whereas for small and light molecules we have $\mathbb{D}_i^T < 0$. Furthermore,

$$\sum_{i=1}^{N} \mathbb{D}_i^T = 0. \tag{2.17}$$

2.4.3 Balance Equations for Gas Species Concentrations

We assume that K reversible chemical reactions take place in the gas phase, with a net molar reaction rate R_k^g ($k = 1, \ldots, K$) and stoichiometric coefficients ν_{ik}, which are further discussed in the next section. The balance equation for the i^{th} gas species, $i = 1, \ldots, N$, in terms of mass fractions and diffusive mass fluxes is then given as

$$\frac{\partial(\rho\omega_i)}{\partial t} = -\nabla \cdot (\rho \mathbf{v} \omega_i) - \nabla \cdot \mathbf{j}_i + m_i \sum_{k=1}^{K} \nu_{ik} R_k^g. \tag{2.18}$$

In equation (2.18) the left-hand side accounts for the transient variations in concentrations, whereas the first and second term on the right-hand side account for the convective and diffusive species transport. The total diffusive mass flux $\mathbf{j}_i$ of species i is composed of ordinary diffusion and thermal diffusion, see expression (2.12). The third term on the right-hand side of equation (2.18) represents the creation and destruction of gaseous species due to homogeneous gas-phase reactions.

2.4.4 Reaction Rates for Gas-Phase Reactions

Under the assumption that K reversible gas-phase reactions of the form

$$\sum_{i=1}^{N} \nu'_{ik} \mathcal{A}_i \underset{k^g_{k,\text{backward}}}{\overset{k^g_{k,\text{forward}}}{\rightleftarrows}} \sum_{i=1}^{N} \nu''_{ik} \mathcal{A}_i \tag{2.19}$$

take place, the net molar reaction rate R^g_k for the k^{th} reaction, see the last term on the right hand side of equation (2.18), is defined as

$$R^g_k = k^g_{k,\text{forward}} \prod_{i=1}^{N} \left(\frac{P \omega_i m}{RT m_i} \right)^{\nu'_{ik}} - k^g_{k,\text{backward}} \prod_{i=1}^{N} \left(\frac{P \omega_i m}{RT m_i} \right)^{\nu''_{ik}}. \tag{2.20}$$

In (2.19), $\mathcal{A}_i$ are the species in the gas mixture, ν'_{ik} the forward stoichiometric coefficient for species i in reaction k, ν''_{ik} the backward stoichiometric coefficient for species i in reaction k. The net stoichiometric coefficient ν_{ik} is then defined as $\nu_{ik} = \nu''_{ik} - \nu'_{ik}$. In equation (2.20), P is the pressure, T the temperature, R the universal gas constant, m_i the molar mass of species i and m the average molar mass, computed as in formula (2.7).

The values of $k^g_{k,\text{forward}}$ and $k^g_{k,\text{backward}}$ depend strongly on the temperature, and are independent of the pressure for sufficiently high pressures. At lower pressures, the so-called pressure fall-off regime, they may also depend on the pressure. For more details we refer to Kleijn (1991) and Kleijn (1995). Usually, the forward reaction rate constant $k^g_{k,\text{forward}}$ is fitted according to a modified Arrhenius expression:

$$k^g_{k,\text{forward}}(T) = A_k T^{\beta_k} e^{\frac{-E_k}{RT}}, \tag{2.21}$$

where A_k, β_k and E_k are fit parameters. The backward reaction rate constants $k^g_{k,\text{backward}}$ are computed self-consistently from the forward reaction rate constants and reaction thermo chemistry, as

$$k^g_{\text{backward}}(T) = \frac{k^g_{\text{forward}}(T)}{K^g(T)} \left(\frac{RT}{P^0} \right)^{\sum_{i=1}^{N} \nu_{ik}}, \tag{2.22}$$

where the reaction equilibrium constant is given by

$$K^g(T) = \exp\left(-\frac{\Delta H^0_k(T) - T \Delta S^0_k(T)}{RT} \right), \tag{2.23}$$

with

$$\Delta H^0_k(T) = \sum_{i=1}^{N} \nu_{ik} H^0_i(T) \quad \text{and} \quad \Delta S^0_k(T) = \sum_{i=1}^{N} \nu_{ik} S^0_i(T). \tag{2.24}$$

In formula (2.22) the atmospheric pressure is denoted as P^0. In expression (2.24) $H_i^0(T)$ is the standard heat of formation as function of temperature and $S_i^0(T)$ the standard entropy as function of temperature. For more details we refer to Kleijn (1991) and Kleijn (1995).

Typically, the forward and backward rate constants of the fastest and slowest reactions can differ many orders of magnitude. For example, in the classical 17 species and 26 reactions chemistry model of the CVD process of silicon from silane, developed by Coltrin et al. (1989), the slowest and fastest reactions differ some 25 orders of magnitude at a temperature of 1000 K. Due to these huge differences the set of species equations (2.18) is extremely stiff.

2.5 Modeling surface chemistry

Usually, heterogeneous surface reactions are characterized by complicated reaction mechanisms that consist of a number of steps. The surface reaction rate will therefore depend on the partial pressures of gaseous species, the rate constants of the individual steps (as functions of local temperature), temperature, surface concentrations and other surface properties. However, there is in general little or no information available on the individual reaction steps and rate constants.

In this study we are not interested in the fundamental modeling of surface chemistry; we will make use of published surface reaction models in which it is assumed that at the wafer surface S irreversible surface reactions take place transforming gaseous reactants into solid products and gaseous byproducts. The s-th transformation of gaseous reactants into solid and gaseous reaction products is of the form

$$\sum_{i=1}^{N} \sigma'_{is} \mathcal{A}_i \xrightarrow{R_s^S} \sum_{i=1}^{N} \sigma''_{is} \mathcal{A}_i + \sum_{j=1}^{M} \chi_{js} \mathcal{B}_j, \tag{2.25}$$

with $\mathcal{A}_i$ as before, $\mathcal{B}_j$ the solid reaction products, M the number of solid reaction products, σ'_{is} and σ''_{is} the stoichiometric coefficients for gaseous species i in surface reaction s and χ_{is} the stoichiometric coefficient for the solid species. Again, the net stoichiometric coefficient σ_{is} is defined as $\sigma_{is} = \sigma''_{is} - \sigma'_{is}$. For further details on surface reaction modeling we refer to Kleijn (1991).

If the surface reaction rate R_s^S is known, then the growth rate $\mathcal{G}_j$ of solid species j is defined as

$$\mathcal{G}_j = \frac{m_j}{\rho_j} \sum_{s=1}^{S} R_s^S \chi_{js}, \tag{2.26}$$

with m_j the molecular mass of solid species j and ρ_j the density of solid species j.

2.6 Boundary Conditions

Within the reactor we have inflow and outflow boundaries, nonreacting solid walls and one or more reacting surface(s). At the inlet and outlet the usual boundary conditions are supplied, i.e.,

- at the inlet there is a prescribed temperature, prescribed species mass fractions and a prescribed inflow velocity, and,
- at the outflow there are homogeneous Neumann conditions for all unknowns.

For adiabatic nonreacting solid walls we impose zero normal temperature gradients, whereas for isothermal walls the temperature is prescribed. For the velocities at nonreacting walls the no-slip and impermeability conditions hold. For the species mass fractions the total mass flux vector normal to a nonreacting surface is equal to zero for each species, i.e.,

$$\mathbf{n} \cdot \mathbf{j}_i = 0, \tag{2.27}$$

where $\mathbf{n}$ is a unity vector normal to the surface of the wall. Note that, due to the presence of thermal diffusion this does not imply a zero normal gradient for the species concentrations.

Due to the irreversible surface reactions (2.25) there is a net mass consumption rate $\mathcal{P}_i$ of gaseous species i at the wafer surface according to

$$\mathcal{P}_i = m_i \sum_{s=1}^{S} \sigma_{is} R_s^S. \tag{2.28}$$

For the velocity component in normal direction we have

$$\mathbf{n} \cdot \mathbf{v} = \frac{1}{\rho} \sum_{i=1}^{N} \mathcal{P}_i, \tag{2.29}$$

while for all other components of the velocity the no-slip condition holds. The temperature on the wafer surface is fixed. The total mass flux of species i normal to the wafer is equal to $\mathcal{P}_i$., i.e.,

$$\mathbf{n} \cdot (\rho \omega_i \mathbf{v} + \mathbf{j}_i) = \mathcal{P}_i. \tag{2.30}$$

The reactants in the CVD processes considered in this thesis are highly diluted in a carrier gas. Therefore, it is justified to assume that

- the velocity-, temperature-, density- and pressure fields are in steady state and not influenced by the transient chemistry, and,
- the velocity component normal to the wafer surface is negligibly small.

Thus, in the simulations presented in this thesis we only account for boundary condition (2.30), whereas for the steady state flow field boundary condition (2.29) is replaced by

$$\mathbf{n} \cdot \mathbf{v} = 0. \tag{2.31}$$

CHAPTER 3

Methods of Lines Approach

Most numerical solvers for time dependent problems follow the popular Method of Lines (MOL) approach, in which space and time discretizations are considered separately. The popularity of this approach is based on its simple concept, flexibility, the fact that various discretizations can easily be combined and that nowadays many well developed ODE methods exist.

Here, the spatial discretization of the stiff system of species equations (2.18) is done in a Finite Volume (FV) setting, yielding a semi-discrete system

$$w'(t) = F(t, w(t)), \quad t \geq 0, \tag{3.1}$$

with the initial value $w(0) = w_0$ given. According to the MOL approach, fully discrete approximations are obtained by applying a suitable time integration method with time step size τ for the time levels $t_n = n\tau, n = 1, 2, \ldots$

Furthermore, we want that the natural property of species mass fractions being non-negative to be conserved in the spatial discretization. In Chapter 4 we go into further details on positivity for the numerical model of CVD. The focus of Chapter 4 is on positivity conserving time integration methods. The emphasis of Section 3.1 will be on positivity conservation of the FV discretization.

Special attention is needed for the FV discretization of the boundary condition at the reacting surface. The reacting surface flux is modeled in such a way that from the mathematical point of view, mass is extracted from the system. If this boundary condition is discretized in a straightforward way, then positivity of the species concentrations along the reacting surface is not guaranteed. In Section 3.4 a positive FV discretization of the particular boundary condition is presented.

3.1 Hybrid Finite Volume Discretization

By defining a computational grid in either two or three spatial dimensions, a Finite Volume semi-discretization can be built for the system of species equations (2.18). In this study the computational grid is always a set of adjoining rectangular control volumes. The unknowns are located in the control volume centers. Each of those grid cells is surrounding one grid point in which all scalar variables, i.e., pressure P, temperature T and the species mass fractions ω_i, are computed. The vector quantities, i.e., the velocity $\mathbf{v}$ and the mass diffusion fluxes $\mathbf{j}_i$, are evaluated at the cell boundaries leading to a staggered grid arrangement.

In this section we shortly present the Finite Volume discretization for the two-dimensional case in cylindrical coordinates. The control-volume surrounding cell center C with cell faces n, e, s, w and corresponding grid points indicated by N(orth), E(ast), S(outh) and W(est), is illustrated in Figure 3.1. The species equations (2.18) are written in the general two-dimensional axisymmetric form

$$\frac{\partial(r\rho\phi)}{\partial t} = -\nabla\cdot(r\rho\mathbf{v}\phi) + \nabla\cdot(r\Gamma\nabla\phi) + rS, \tag{3.2}$$

with ϕ as the unknown, ρ the density, $\mathbf{v}$ the velocity, Γ the diffusion coefficient, r the radial coordinate and S the reaction term. The two-dimensional cartesian version of the species equations (2.18) is retrieved by setting $r = 1$ in equation (3.2). Integrating equation (3.2) over the control-volume $\Delta r\Delta z$ surrounding cell center C and applying the Gauss Divergence Theorem gives

$$\frac{\partial(r_C\rho\phi_C)}{\partial t}\Delta r\Delta z = \sum_{i=n,e,s,w}\int_{S_i}(r\rho\mathbf{v}\phi + r\Gamma\nabla\phi)\cdot\mathbf{n}dS + r_C S_C\Delta r\Delta z. \tag{3.3}$$

The Finite Volume formulation is completed by approximating ϕ and its first derivative on the cell walls. In the literature several methods are proposed to approximate both quantities, see for instance Patankar (1980). In this study we approximate them by the central scheme if possible and by the first order upwind scheme if neccesary.

This is illustrated for the two-dimensional case on the n-wall. Define the cell-Péclet number on the n-wall as

$$\mathrm{Pe}_n = \frac{\rho_n v_n \Delta z_n}{\Gamma_n}. \tag{3.4}$$

The hybrid scheme approximates ϕ_n as

$$\phi_n = \begin{cases} \phi_N & \text{for} \quad \mathrm{Pe}_n < -2 \\ \frac{1}{2}(\phi_N + \phi_C) & \text{for} \quad |\mathrm{Pe}_n| \le 2 \\ \phi_C & \text{for} \quad \mathrm{Pe}_n > 2 \end{cases}, \tag{3.5}$$

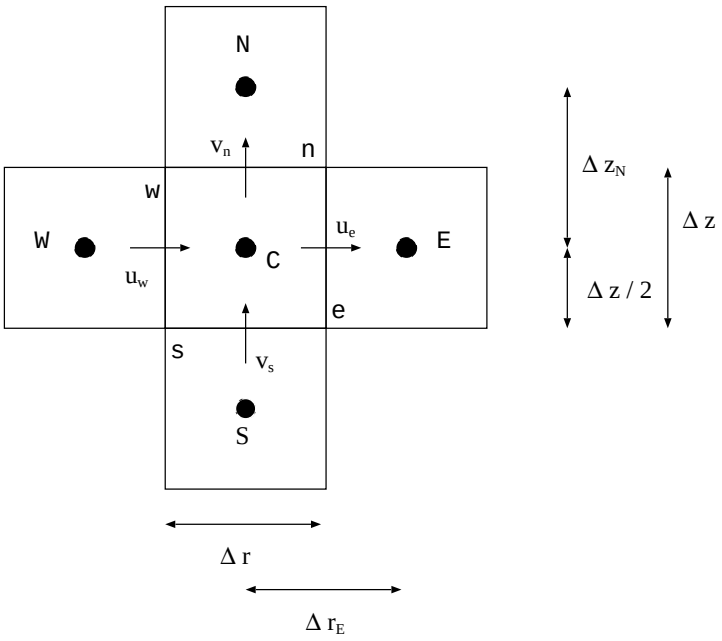


Figure 3.1: Grid cells

and its first partial derivative with respect to z on the n-wall as

$$\left.\frac{\partial \phi}{\partial z}\right|_n = \frac{\phi_N - \phi_C}{\Delta z_n}, \tag{3.6}$$

for all Pe_n.

It is generally known that the second order accurate central scheme is stable for cell-Péclet numbers in absolute value less than two, i.e., $|\mathrm{Pe}| \leq 2$, and becomes unstable otherwise. The first order accurate upwind scheme, on the other hand, is stable for all cell-Péclet numbers. Its price is the loss of accuracy with respect to the central scheme, and its large numerical diffusivity, see for instance Hundsdorfer & Verwer (2003). To summarize, combining these two schemes gives a stable and positive spatial Finite Volume discretization of the species equations (2.18), which is second order accurate if possible.

3.2 Higher Order Upwinding

Instead of first order upwinding one could also use higher order upwinding schemes to discretize the advection flux. In this section we show that one has to be careful when using higher order upwinding. It is known that such schemes are not positivity preserving for all mesh sizes. For example,

consider the constant coefficient one-dimensional advection equation

$$\frac{\partial \phi}{\partial t} + a\frac{\partial \phi}{\partial x} = 0, \tag{3.7}$$

on the spatial domain $0 \leq x \leq 1$ with $a > 0$ a constant advection coefficient. At $x = 0$ we impose the Dirichlet boundary condition

$$\phi(0, t) = 1, \quad t \geq 0, \tag{3.8}$$

whereas at $x = 1$ a homogeneous Neumann boundary condition is imposed.

For $a > 0$ the third-order upwind-biased scheme reads

$$\frac{\partial \phi_j}{\partial t} = \frac{a}{h}\left(-\frac{1}{6}\phi_{j-2}(t) + \phi_{j-1}(t) - \frac{1}{2}\phi_j(t) - \frac{1}{3}\phi_{j+1}(t)\right), \tag{3.9}$$

where h is the mesh-width. In vector notation the semi-discrete system (3.9) reads

$$\phi'(t) = A\phi(t) + b, \tag{3.10}$$

where the entries of A are determined through (3.9) and b is the vector corresponding to the Dirichlet boundary condition at $x = 0$. The exact solution of (3.10) is given by

$$\phi(t) = e^{tA}\left(y_0 + A^{-1}b\right) - A^{-1}b. \tag{3.11}$$

In Figure 3.2 the exact solution (3.11) is presented for the third-order upwind-biased scheme with $h = 1/50$ and $a = 1$. The solution is stable, but it has negative values for $0.55 \leq x \leq 0.65$. For the second order central scheme and the first order upwinding scheme applied to equation (3.7) the exact solutions of the resulting semi-discrete system are shown as well. From Figure 3.2 it can be seen that for $h = 1/50$ the second order central schemes returns unstable solutions. The first order upwind scheme produces neither oscillations nor negative values, but it has the drawback to damp the solution, see also Hundsdorfer & Verwer (2003).

In the CVD processes considered in this study some species have concentration profiles with steep gradients. In the case that the $|\mathrm{Pe}| < 2$ condition is not satisfied, we would like to have a discretization such that positivity is ensured. The second order central scheme and the third-order upwind-biased scheme clearly do not preserve positivity of the solution for all mesh-sizes, whereas the first order upwinding scheme does. The damping of the solution due to the local use of this scheme is further discussed in the next section.

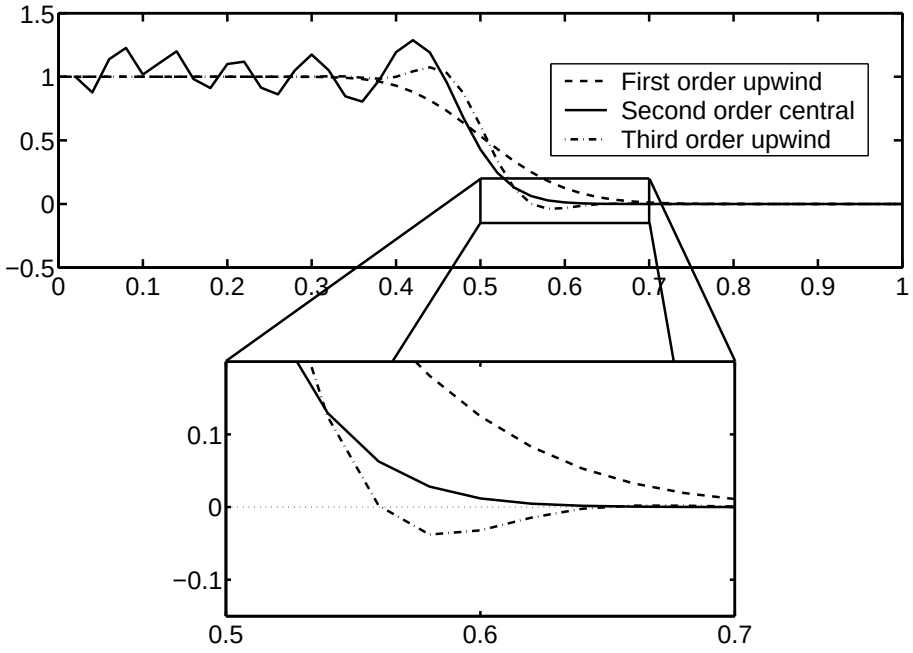


Figure 3.2: Numerical tests for linear advection equation (3.7) with mesh-width $h = 1/50$. Dashed: First order upwinding. Dash-dot: Third order upwind-biased scheme (3.9). Solid: Second order central scheme.

3.3 Damping of the Hybrid Scheme for Low Reynolds Number CVD Flows

Typically, for the CVD processes in this study the Reynolds numbers

$$\mathrm{Re} = \frac{UL}{\nu}, \tag{3.12}$$

where L is a characteristic length, U is a characteristic velocity and ν is the kinematic viscosity, are below 100. Consequently, cell Péclet numbers

$$\frac{U\Delta x}{\mathbb{D}_i'}, \tag{3.13}$$

for the species equations are below 2 when

$$\frac{\Delta x}{L}\frac{\nu}{\mathbb{D}_i'} < 0.02. \tag{3.14}$$

Since for gases

$$\frac{\mathbb{D}_i'}{\nu} \sim 1, \tag{3.15}$$

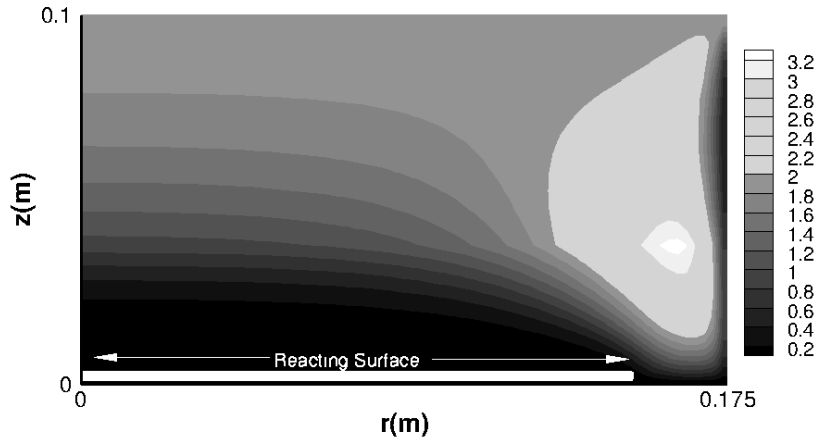
we have that

$$\Delta x < 0.02L. \tag{3.16}$$

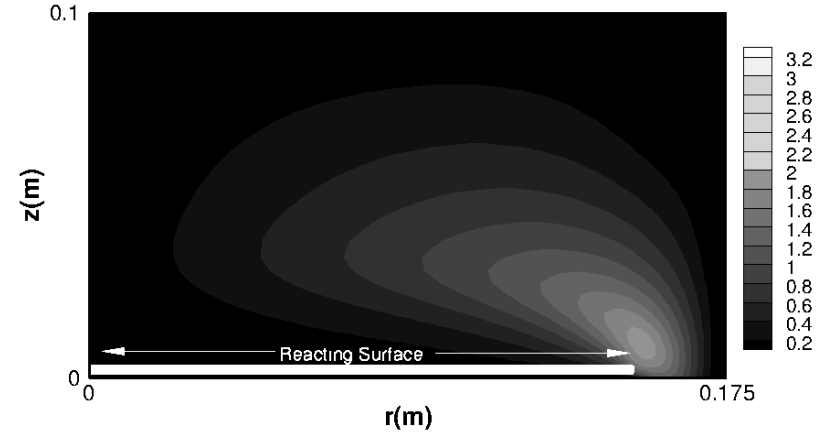
For this reason, on meshes with ~ 50 or more cells per direction, the hybrid scheme (3.5) - (3.6) usually results in a central differencing scheme.

In Figure 3.3 the cell-Péclet numbers are presented for the numerical simulations on the finest mesh in Chapter 8 of this thesis. Indeed, from Figure 3.3(a) it can be concluded that there is only a small region in the computational domain where the cell Péclet number in the vertical direction is larger than two in absolute value. In this small region of the reactor there are no gas phase reactions, nor is it in the region of interest above the reacting surface. The cell Péclet numbers in horizontal direction, see Figure 3.3(b), are less than two in the entire computational domain. Thus, for this reactor configuration, the hybrid scheme (3.5) - (3.6) is almost everywhere in the computational domain second order accurate.

To conclude, the damping of the first order upwinding in the hybrid FV scheme (3.5) - (3.6) is usually not occuring for the CVD problems considered in this study. If first order upwinding is needed to ensure stability and positivity, then it is only needed in areas of the computational domain that are not critical to the accuracy of the solution. The fact that the hybrid scheme (3.5) - (3.6) is unconditionally positive for all mesh sizes is more important.



(a) Vertical mesh-Péclet numbers



(b) Horizontal mesh-Péclet numbers

Figure 3.3: Cell Péclet numbers on the finest mesh for the reactor geometry used in Kleijn (2000), van Veldhuizen et al. (2008b) and Chapter 8 of this thesis. In Figure 3.3(a) the cell Péclet numbers in vertical direction are given. In Figure 3.3(b) the cell Péclet numbers in horizontal direction are given.

3.4 Discretization of the Surface Reaction Flux

Recall that on the boundary which corresponds to the reacting wafer surface the following boundary condition is imposed: The total mass transport flux of species i in the outward normal direction is equal to the mass production rate $\mathcal{P}_i$ of species i (see equation (2.28)). For dilute systems, with $\mathbf{n} \cdot \mathbf{v} = 0$ at the reacting surface, this condition mathematically is denoted as

$$\mathbf{n} \cdot \mathbf{j}_i = \mathcal{P}_i. \tag{3.17}$$

On the cell wall of the control volume that corresponds to the reacting surface as illustrated in Figure 3.4, the diffusion flux in normal direction is approximated as

$$\mathbf{n} \cdot \mathbf{j}_i = \rho_{\text{wall}} \mathbb{D}'_i \frac{(\omega_{i,\text{wall}} - \omega_{i,\text{center}})}{1/2\Delta z} + \frac{\mathbb{D}^T}{T_{\text{wall}}} \frac{T_{\text{wall}} - T_{\text{center}}}{1/2\Delta z}, \tag{3.18}$$

where

- ρ_{wall} denotes the density of the gas mixture at the wafer,
- $\mathbb{D}'_i$ denotes the effective diffusion coefficient, see equation (2.14),
- $\mathbb{D}^T$ is the effective thermal diffusion coefficient for species i, see equation (2.16),
- $1/2\Delta z$ denotes the distance from the reacting surface to the cell center of the corresponding control volume, see Figure 3.4,
- $\omega_{i,\text{center}}$ is the mass fraction of species i at the cell center of the corresponding control volume,
- $\omega_{i,\text{wall}}$ is the mass fraction of species i at the wafer,
- T_{center} is the temperature at the cell center of the corresponding control volume, and,
- T_{wall} is the temperature at the wafer.

Remark that the species mass fraction $\omega_{i,\text{wall}}$ of species i at the wafer is an unknown in equation (3.17). However, we are not interested in computing the mass transport flux, but in computing the mass production rate $\mathcal{P}_i$ of species i, which is a function of the species mass fraction at the wafer. Therefore, an approximation of $\omega_{i,\text{wall}}$ is needed. This approximation should satisfy the requirements of a mass fraction being positive and less than or equal to one.

3.4.1 Extrapolating Cell Centered Species Mass Fractions

The most straightforward way to approximate ω_{wall} is by linear extrapolation. This approach, amongst others, is followed in Kleijn (1991) and Kleijn (2000). In the situation illustrated in Figure 3.4, ω_{wall} would be approximated as

$$\omega_{\text{wall}} = \omega_{\text{center}} + \frac{1/2\Delta z}{\Delta z_N}(\omega_{\text{center}} - \omega_N). \tag{3.19}$$

Then, ω_{wall} is neither guaranteed to be positive nor to be less or equal than one. For dilute systems as studied in this thesis, species mass fractions are not likely to be larger than one, even when inaccurate extrapolation is applied (remember that the mass fraction of the carrier gas is computed as one minus the sum of the other mass fractions). However, there is a serious danger that inaccurately extrapolated mass fractions of species that are consumed at a wall will become negative. This is confirmed by numerical simulations in which equation (3.19) was applied. We will therefore focus on a method to compute wall mass fractions of consumed species in a way that preserves non-negativity.

3.4.2 A Positive Approximation of ω_{wall}

All surface reactions considered in this study are unimolecular decomposition reactions, see Section 2.5. Therefore, the mass consumption rate at the surface is linearly proportional to the species molar concentration, and consequently, for dilute systems, also proportional to the species mass fraction. The mass consumption rate $\mathcal{P}_i$, see expression (2.28), can be written as

$$\mathcal{P}_i = -m_i K_i \omega_{i,\text{wall}}, \tag{3.20}$$

with K_i a positive constant (K_i depends on local surface temperature and local pressure).

In the dilute mixture approach, see for instance Kleijn (1995), the effective thermal diffusion coefficient $\mathbb{D}_i^T$ for species i is written as

$$\mathbb{D}_i^T = \alpha_{\text{TD}} \omega_i \rho \mathbb{D}_i', \tag{3.21}$$

with α_{TD} the thermal diffusion factor of species i in the carrier gas. Then, for these species, $\omega_{i,\text{wall}}$ can be computed from expressions (3.17), (3.18), (3.20) and (3.21) as

$$\omega_{i,\text{wall}} = \frac{\omega_{i,\text{center}}}{1 + \alpha_{\text{TD}} \frac{T_{\text{wall}} - T_{\text{center}}}{T_{\text{wall}}} - \frac{\Delta z m_i K_i}{2\rho_{\text{wall}} \mathbb{D}_i'}}. \tag{3.22}$$

For sufficiently fine meshes along the reacting boundary $|T_{\text{wall}} - T_{\text{center}}| \ll T_{\text{wall}}$. Further, $|\alpha_{\text{TD}}| = O(1)$.

To summarize, $\omega_{i,\text{wall}}$, computed via expression (3.22), is a mass fraction (meaning that $0 \leq \omega_{i,\text{wall}} \leq 1$) as long as $\omega_{i,\text{center}}$ is a mass fraction, because

1. if $\omega_{i,\text{center}} < 1$, then also $\omega_{i,\text{wall}} < 1$, and,
2. if $\omega_{i,\text{center}}$ is positive, then also $\omega_{i,\text{wall}}$ is positive.

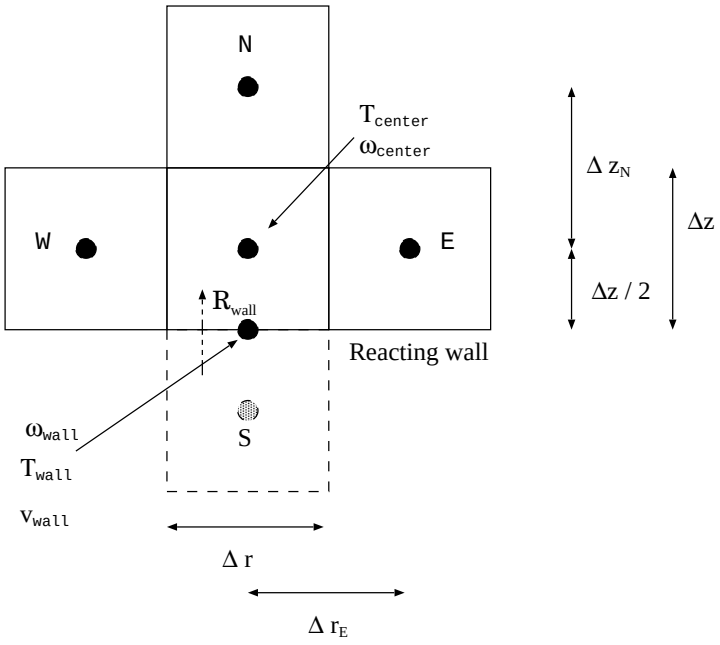


Figure 3.4: Grid cells along the reacting boundary. The south cell, with cell center S, is a virtual cell. The species mass fraction ω_{wall} is computed according to expression (3.22), whereas T_{wall} is prescribed (see Section 2.6).

CHAPTER 4

Positivity

The physical interpretation of the solutions ω_i of the system of species equations (2.18) tells us that

$$\omega_i(\mathbf{x}, 0) > 0 \text{ for all } x \text{ implies } \omega_i(\mathbf{x}, t) > 0 \text{ for all } x \text{ and } t > 0.$$

This property is called positivity, which is an abbreviation of 'non-negativity preserving for the species concentrations in the solution vector'. Looking closer to this property, we remark that first of all the mathematical model proposed in Kleijn (1991) and discussed in Chapter 2 should be positive. Obviously, the advection and diffusion parts are non-negativity preserving. Positivity of the reaction terms (2.20) in the species equations (2.18) is discussed in Section 4.1.

As we have seen in Chapter 3, it is in general not guarantueed that spatial discretizations preserve non-negativity. The hybrid Finite Volume discretization of the species equations (2.18) introduced in Chapter 3 is stable and positive for all mesh sizes. Following the MOL approach, the obtained positive semi-discretization should be integrated in time. Again, we would like to have one or more critera that tell us when positivity is preserved. It appears that this extra condition on time integration methods, besides stability, is much more restrictive towards the time step size than stability.

In Section 4.1 we discuss positivity of semi-discretizations and/or ODE systems. Thereafter, in Section 4.2, positivity for time integration methods is considered. Relations between positivity and other monoticity properties like Total Variation Diminishing (TVD) are addressed in Section 4.3.

4.1 Positive Semi-Discretizations

In this section we investigate positivity for ODE systems

$$w'(t) = F(t, w(t)). \tag{4.1}$$

In this thesis the ODE system (4.1) represents a semi-discrete system of the time dependent species equations (2.18), which is obtained through discretization in space by means of the hybrid Finite Volume method of Chapter 3.

Throughout this chapter we assume that the semi-discrete system (4.1) consists of $m = n \cdot N$, with N the number of species and n the number of spatial grid points, time dependent ODEs,

$$\begin{bmatrix} w_1'(t) \\ \vdots \\ w_m'(t) \end{bmatrix} = \begin{bmatrix} F_1(t, w(t)) \\ \vdots \\ F_m(t, w(t)) \end{bmatrix}. \tag{4.2}$$

Further, by $w(t) \geq 0$ it is meant that this inequality is satisfied component-wise, i.e.,

$$w(t) \geq 0 \quad \Longrightarrow \quad w_1(t) \geq 0, \ldots, w_m(t) \geq 0. \tag{4.3}$$

Definition 4.1. *An ODE system $w'(t) = F(t, w(t))$, $t \geq 0$, is called positive, or non-negativity preserving, if $w(0) \geq 0$ (component-wise) $\Longrightarrow$ $w(t) \geq 0$, for all $t > 0$.*

The next theorem provides a simple criterion on $F(t, w(t))$ to test whether the ODE system $w'(t) = F(t, w(t))$, $t \geq 0$, is positive. For a proof we refer to Hundsdorfer & Verwer (2003).

Theorem 4.2. *Suppose that $F(t, w)$ is continuous and satisfies a Lipschitz condition with respect to w. Then the ODE system $w'(t) = F(t, w(t))$, $t \geq 0$, is positive if and only if for any vector $w \in \mathbb{R}^m$ and all $i = 1, \ldots, m$, and $t \geq 0$ yields*

$$w \geq 0 \quad (\textit{componentwise}), \quad w_i = 0 \Longrightarrow F_i(t, w) \geq 0. \tag{4.4}$$

The goal of this section is to investigate positivity for semi-discrete systems. Consider, for instance, the one dimensional linear advection-diffusion equation

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}\left(a(x,t)u(x,t)\right) = \frac{\partial}{\partial x}\left(d(x,t)\frac{\partial}{\partial x}u(x,t)\right), \tag{4.5}$$

with periodic boundary conditions, and where $a(x, t)$ is the space and time dependent advection coefficient, and $d(x, t) > 0$ the space and time dependent diffusion coefficient. Application of Theorem 4.2 shows that Finite

Volume discretization by means of central differences gives a positive semi-discretization if and only if the cell Péclet numbers, defined as ah/d, satisfy

$$\max_{x,t} \frac{|a(x,t)|h}{d(x,t)} \leq 2. \tag{4.6}$$

Discretizing the advection part by means of first order upwind, and second order central differences for the diffusive part, gives an unconditionally positive semi-discretization.

In Chapter 3 an equivalent 2D approach is presented for a Finite Volume discretization. Positivity of the semi-discrete system is achieved when the spatially discretized reaction terms (2.20) are also positivity preserving.

The reaction terms (2.20) can be written in the production-loss form

$$R_k^g(t,w) = p(t,w) - L(t,w)w, \tag{4.7}$$

where $p(t,w) \geq 0$ (componentwise) is a vector and $L(t,w) \geq 0$ (componentwise) a diagonal matrix. The components $p_i(t,w)$ of $p(t,w)$ and $L_i(t,w)$ of the diagonal matrix $L(t,w)$ are of polynomial type in w with non-negative coefficients. These coefficients are easily found for practical examples.

Addition of reaction terms (2.20), which can be written in the production-loss form (4.7), to the advection-diffusion equation (4.5) and applying Theorem 4.2 gives a positive semi-discretization for the one dimensional advection-diffusion-reaction equation if and only if $p(t,w) \geq 0$, see also Hundsdorfer & Verwer (2003).

The one-dimensional results above are easily generalized to higher dimensions and to FV schemes, as has been done in Chapter 3. In Chapter 3 it has already been remarked that the hybrid FV scheme preserves positivity, whereas in this section a mathematical foundation is presented for the derivation of these conditions.

Further, note that by establishing the positivity of the (discretized) reaction terms (2.20) implies that the mathematical model of the gas-phase chemistry is positive.

4.2 Positive Time Integration

Definition 4.3. *A time integration method $w_{n+1} = \varphi(w_n)$ is called positive if for all $n \geq 0$ holds, $w_n \geq 0 \Longrightarrow w_{n+1} \geq 0$.*

The positivity requirement restricts the choice of time integration methods. In this section we will present results for general ODE systems, i.e., non-linear systems $w'(t) = F(t, w(t))$. First, we start exploring the positivity property for the Euler Forward and Euler Backward time integration methods.

4.2.1 Positivity for Euler Forward and Euler Backward

Suppose that the right hand side of the non-linear semi-discretization $w'(t) = F(t, w(t))$ satisfies:

Condition 4.4. *There is an $\alpha > 0$, depending on $F(t, w)$, such that for a time step τ holds:*

$$\text{if } \alpha\tau \leq 1, \text{ then } w + \tau F(t, w) \geq 0 \text{ for all } t \geq 0 \text{ and } w \geq 0.$$

Provided that $w_n \geq 0$, Condition 4.4 guarantees positivity for w_{n+1} computed via Euler Forward. For linear semi-discrete systems $w'(t) = Aw(t)$ with entries $A_{ij} \geq 0$ for $i \neq j$, $A_{ii} \geq -\zeta$ for all i and $\zeta > 0$ fixed, Condition 4.4 is easily illustrated. Application of Euler Forward to this system gives a positive solution if $1 + \tau A_{ii} \geq 0$ for all i. This will hold if $\zeta\tau \leq 1$. To write down such an expression for α for equation (2.18) is undoable, because of the complicated structure of the chemical source terms. At least, α should be such that Euler Forward gives stable numerical solutions.

Secondly, assume that $F(t, w(t))$ satisfies :

Condition 4.5. *For any $v \geq 0, t \geq 0$ and $\tau > 0$ the equation*

$$w = v + \tau F(t, w), \tag{4.8}$$

has a unique solution w that depends continuously on τ and v.

According to the following theorem we have unconditional positivity for Euler Backward. The proof is taken from Hundsdorfer & Verwer (2003).

Theorem 4.6. *Conditions 4.4 and 4.5 imply positivity for Euler Backward for any time step size τ.*

Proof. For given t, v and with a chosen τ, we consider the equation $w = v + \tau F(t, w)$ and we call its solution $w(\tau)$. We have to show that $v \geq 0$ implies $w(\tau) \geq 0$ for all positive τ. By continuity it is sufficient to show that $v > 0$ implies $w(\tau) \geq 0$. This is true because if we assume that $w(\tau) > 0$ for $\tau \leq \tau_0$, except for the i^{th} component $w_i(\tau_0) = 0$, then $0 = w_i = v_i + \tau_0 F_i(t, w(\tau_0))$. According to Condition 4.4 we have $F_i(t, w(\tau_0)) \geq 0$ and thus $v_i + \tau_0 F_i(t, w(\tau_0)) > 0$, which is a contradiction. □

Remark 4.7. *Application of Euler Backward to the nonlinear semi-discretization $w'(t) = F(t, w(t))$ needs the solution of the nonlinear vector equation*

$$w_{n+1} - \tau F(t_n, w_{n+1}) = w_n. \tag{4.9}$$

Theorem 4.6 ensures for every time step size τ positivity of the exact solution of (4.9). In practice, however, the solution of (4.9) is approximated by an iterative solver, and thus, it is not guaranteed to be positive.

4.2.2 Higher Order Positive Time Integration

Implicit time integration methods are useful in the sense that they eliminate the time-step restriction associated with stability. Therefore, unconditionally positive schemes can be implicit schemes only. We would like to preserve the unconditional positivity of Euler Backward in higher order time integration methods.

Unfortunately, this is not possible. Look for instance to the second order Runge-Kutta method

$$w^{(1)} = w^n + \beta_1 \tau F(t_n + \beta_1 \tau, w^{(1)}) \tag{4.10}$$

$$w^{n+1} = \alpha_1 w^n + \alpha_2 w^{(1)} + \beta_2 \tau F(t_{n+1}, w^{n+1}). \tag{4.11}$$

Note that (4.10) - (4.11) contains no explicit terms in order to avoid time-step restrictions for stability and positivity requirements. Second order accuracy requires the coefficients in (4.10) - (4.11) to satisfy

$$\alpha_2 = \frac{1}{2\beta_1(1-\beta_1)}, \quad \alpha_1 + \alpha_2 = 1, \quad \text{and,} \quad \beta_2 = \frac{1-2\beta_1}{2(1-\beta_1)}. \tag{4.12}$$

Under the assumption that $w_n \geq 0$, Theorem 4.6 guarantees $w^{(1)} \geq 0$. Further, Condition 4.4 and 4.5 and Theorem 4.6 imply w^{n+1} to be positive when $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$. Elementary calculations show that either $\alpha_2 \in [2, \infty)$, or $\alpha_2 \in (-\infty, 0)$, which implies that either α_1 is negative, or α_2 is negative. Thus, we have shown that the second order implicit Runge-Kutta scheme (4.10) - (4.11) cannot be unconditionally positive. In fact, we have proven that it cannot be positive for any time step size τ.

The simple analysis on the second order implicit Runge-Kutta scheme (4.10) - (4.11) presented above is perfectly generalized for all higher order time integration methods in the following result, due to Bolley & Crouzeix (1973).

Theorem 4.8. *Any unconditionally positive time integration method has order* $p \leq 1$.

For a proof we refer to Bolley & Crouzeix (1973). The consequence is that the only well-known method having unconditionally positivity is Euler Backward. Finally, we remark that for higher order methods the need to preserve positivity may necessitate the use of impractically small time steps.

4.3 Positivity and TVD

Like positivity, Total Variation Diminishing (TVD) is a form of super stability. The TVD property is developed for studying the properties of numerical

schemes to solve hyperbolic conservation laws, see, for instance, Gottlieb et al. (2001), Hundsdorfer & Verwer (2003), Hundsdorfer et al. (2003), LeVeque (2002) and Wesseling (2001). If a system of ODEs

$$w'(t) = F(t, w(t)), \tag{4.13}$$

with an appropriate initial condition $w(0) = w_0$, stands for a semidiscretization of a hyperbolic conservation law, then it is important that the fully discrete process is monotonic in the sense that

$$\|w_n\| \leq \|w_{n-1}\|, \tag{4.14}$$

for a certain norm $\|\cdot\|$. If a numerical scheme satisfies the monoticity property (4.14), where for $\|\cdot\|$ the seminorm

$$|y|_{\mathrm{TV}} = \sum_{j=1}^{n} |y_j - y_{j-1}|, \quad \text{with} \quad y_0 = y_n, \quad \text{for} \quad y \in \mathbb{R}^n, \tag{4.15}$$

is used, then such a scheme is called Total Variation Diminishing (TVD). If a numerical scheme satisfies

$$|w_n|_{\mathrm{TV}} \leq |w_{n-1}|_{\mathrm{TV}}, \tag{4.16}$$

then localized over- and undershoots are prevented. In the case that $w_{n-1} \geq 0$, then inequality (4.16) implies that $w_n \geq 0$.

Conditions on time integration methods to ensure positivity or TVD are derived in the same way, see for instance Hundsdorfer et al. (2003) and van Veldhuizen et al. (2008b). For certain implicit schemes these conditions are identical. For instance, the Euler Backward method is both unconditionally positive and unconditionally TVD. Also for Diagonal Implicit Runge-Kutta methods the conditions for positivity and TVD are the same, see Hundsdorfer & Verwer (2003) and van Veldhuizen et al. (2006b).

As for positivity, it is shown that higher order unconditional TVD time integration methods do not exist. In Gottlieb et al. (2001) this has been proven for implicit Runge-Kutta schemes and for implicit multi-step methods. Recall that explicit schemes always have to fulfill a CFL condition in order to be TVD, or positive, see Gottlieb et al. (2001).

Higher order time integration methods satisfying the TVD property (4.14) must have explicit stages, see Gottlieb et al. (2001) and van Veldhuizen et al. (2008b). Addition of explicit stages to an implicit higher order time integration method could retrieve the TVD, or positivity property, see Gottlieb et al. (2001) or van Veldhuizen et al. (2006b). However, due to the huge stiffness of the species equations (2.18) explicit time integration is ruled out by stability requirements.

4.4 Conclusions

Clearly, for time accurate simulations of reacting gas flows higher order time integration is desired. On the other hand, negative species are absolutely undesired, because they cause blow up of the solution in finite time. Conservation of non-negativity is therefore essential. However, as has been shown in this chapter, unconditionally positive time integration methods can be first order accurate only. Euler Backward is the only known time integration method being unconditionally positive.

Higher order time integration of the stiff species equations will require a tight restriction on the time step in order to maintain positivity. Other alternatives like time splitting or IM(plicit)-EX(plicit) time integration combined with higher order schemes could also be considered. In the next chapter we will amongst others discuss the advantages and the disadvantages of such methods.

Since in practice the solutions of the nonlinear vector equations arising from implicit time discretizations are approximated by iterative methods, their non-negativity is not even ensured when Euler Backward time integration is used (compare with Remark 4.7). In Chapter 6 and 7 we discuss positivity issues for Newton's method and for Krylov methods, respectively.

CHAPTER 5

Comparison of Some Stiff ODE Methods

From the previous chapter it can be concluded that Euler Backward is an almost ideal time integration method. It has the advantages of being unconditionally stable and positive. Disadvantages are the first order consistency and its damping. In this chapter, we will discuss a selection of higher order time integration methods that are suitable to integrate equation (2.18) from a theoretical point of view.

This chapter is organized as follows. First we start off by recalling some basic notions on stability, operator splitting and variable time step size selection. All ODE schemes which have been tested are equipped with a variable time step selector, as is usual in the ODE field. Detailed descriptions on variable time stepping can be found in Hairer & Wanner (1996), Hundsdorfer & Verwer (2003), van Veldhuizen et al. (2006a) and van Veldhuizen et al. (2007b).

In Sections 5.2 - 5.5, the actual implemented ODE schemes are presented. For each particular scheme stability issues, positivity conditions and implementation details are provided. This chapter is concluded with numerical results.

If in this chapter, and subsequent chapters, the norm $\|\cdot\|$ is not specified, then the L_2 norm is used.

5.1 Basic Notions

In this section we introduce the notation used throughout this chapter, as well as some notions on stability of time integration methods. Further, advantages and disadvantages of splitting methods are shortly discussed.

To obtain an efficient code, the implementation of a time step size selector is indispensable. This issue is addressed in Section 5.1.3.

5.1.1 Stability

Consider the Dahlquist test equation, see Hairer & Wanner (1996),

$$w'(t) = \lambda w(t), \tag{5.1}$$

with $\lambda \in \mathbb{C}$. Application of a one-step method, like for instance a Runge-Kutta method, to equation (5.1) gives the recursion

$$w_{n+1} = \mathcal{R}(z) w_n, \quad z = \tau\lambda, \tag{5.2}$$

with τ the time step size $\tau = t_{n+1} - t_n$. The function $\mathcal{R}(z)$ is called the stability function of the particular method. The stability region of this particular method is the set $\mathcal{S} \in \mathbb{C}$ defined as

$$\mathcal{S} = \{z \in \mathbb{C} : |\mathcal{R}(z)| \leq 1\}. \tag{5.3}$$

A time integration method is called A-stable if the left half plane $\mathbb{C}^-$,

$$\mathbb{C}^- = \{z \in \mathbb{C} : \mathrm{Re}(z)| \leq 0\}, \tag{5.4}$$

is contained in $\mathcal{S}$, i.e.,

$$\mathbb{C}^- \subset \mathcal{S}. \tag{5.5}$$

Further, a time integration method is called L-stable if this method is A-stable and $\mathcal{R}|\infty| = 0$. Unconditional stability of a time integration method is obtained when $\mathrm{Re}(\lambda) < 0$ and the time integration method is A-stable.

In order to derive the stability region for linear multistep methods extra notions are needed. Since the scope of this study is not on the derivation of such results, we restrict ourselves to referring to the comprehensive descriptions, for instance Hairer et al. (1987), Hairer & Wanner (1996), Hundsdorfer & Verwer (2003) and van Veldhuizen (2005).

5.1.2 Splitting Methods

For a general advection-diffusion-reaction problem

$$\frac{\partial w}{\partial t} + \nabla \cdot (\mathbf{a} w) = \nabla \cdot (\mathbf{D} \nabla w) + R(w), \tag{5.6}$$

it is generally inefficient to apply the same time integration method to different parts of the system. If the reaction terms $R(u)$ are stiff, as in the case of the species equations (2.18), then that part of equation (5.6) calls for an implicit time integration method. Discretized advection terms are often more suitable to be integrated explicitly.

Solving the spatially discretized system of ODEs (5.6) by means of a simple implicit integration rule results in a large system of nonlinear algebraic equations. Due to the underlying spatial connectivity of the discretized advection and diffusion terms the nonlinear systems becomes large, and hence computationally expensive to solve. The basic idea behind splitting is to treat each term in equation (5.6) separately, such that each term is efficiently integrated in time.

Operator Splitting

First, we shortly discuss a splitting technique called operator-, or time splitting. Consider a general ODE system

$$w'(t) = F(t, w(t)), \tag{5.7}$$

with a two term splitting

$$F(t, w) = F_1(t, w) + F_2(t, w). \tag{5.8}$$

We illustrate the time splitting method by the first order splitting

$$\begin{array}{rclcrcl} \frac{d}{dt}w^*(t) & = & F_1(t, w^*(t)) & \text{for } t_n < t \le t_{n+1} & \text{with} & w^*(t_n) & = & w_n, \\ \frac{d}{dt}w^{**}(t) & = & F_2(t, w^{**}(t)) & \text{for } t_n < t \le t_{n+1} & \text{with} & w^{**}(t_n) & = & w^*(t_{n+1}), \end{array}$$

giving $w_{n+1} = w^{**}(t_{n+1})$ as the next approximation.

The splitting error can be derived by Taylor expansions of $w^*(t_{n+1})$ and $w^{**}(t_{n+1})$ around $t = t_n$. It is equal to

$$\rho_n = \frac{1}{2}\tau \left[\frac{\partial F_1}{\partial w} F_2 - \frac{\partial F_2}{\partial w} F_1 \right] (t_n, w(t_n)) + O(\tau^2). \tag{5.9}$$

If the bracketed term equals zero the splitting error is of order τ^2, but this is generally not true.

Following the idea of Strang (1968) to use symmetry in splitting methods, a second order splitting can be obtained. Multiple application of the second order Strang splitting operator, see Strang (1968), gives higher order splittings. For more comprehensive descriptions of these methods we refer to Hundsdorfer & Verwer (2003).

Remark 5.1. *For linear ODE problems with a two term splitting as in equation (5.8), where we assume that $\|F_1\|$ is bounded and F_2 has an eigenvalue equal to $1/\varepsilon$, with $1 \gg \varepsilon > 0$, Sportisse (2000) and Verwer & Sportisse (1998) have proven that the first order splitting remains first order accurate. The second order Strang splitting gives, in general, also a first order accurate splitting. Thus, in the stiff case we can expect that first order splitting is, in general, the most accurate splitting one can obtain. First order splitting for nonlinear hyperbolic equations with stiff source terms has been analyzed by Tang (1998). Further references and discussion can, for instance, be found in Hundsdorfer & Verwer (2003).*

Remark 5.2. *Operator splitting always gives a splitting error; for the first order splitting (5.8) this error is equal to expression (5.9). This implies that steady states are not returned exactly. For codes that compute a time accurate transient solution until a steady state is reached, this property is not desired.*

IMEX

IM(plicit)-EX(plicit) methods are methods that are a suitable mix of implicit and explicit methods. The concept of IMEX can be applied to both Runge-Kutta type and multistep type of time integration methods. The concept is illustrated by combining Euler Forward and Euler Backward to the general ODE (5.7), where $F(t, w)$ has a two term splitting (5.8). Further, assume that $F_1(t, w)$ is a non-stiff term, and $F_2(t, w)$ is too stiff to be integrated explicitly. The IMEX approach is then

$$w_{n+1} = w_n + \tau F_1(t_n, w_n) + \tau F_2(t_{n+1}, w_{n+1}). \tag{5.10}$$

By Taylor series expension we obtain for the truncation error

$$\rho_n = -\frac{1}{2}\tau w''(t_n) + \tau F_1'(t_n, w_n) + O(\tau^2). \tag{5.11}$$

With respect to stability the following is derived. If we assume that the implicit part of method (5.10) is stable, then the stability region for the explicitly integrated $F_1(t_n, w_n)$ is equal to the stability region of Euler Forward. The stability region of Euler Forward is illustrated in Figure 5.1.

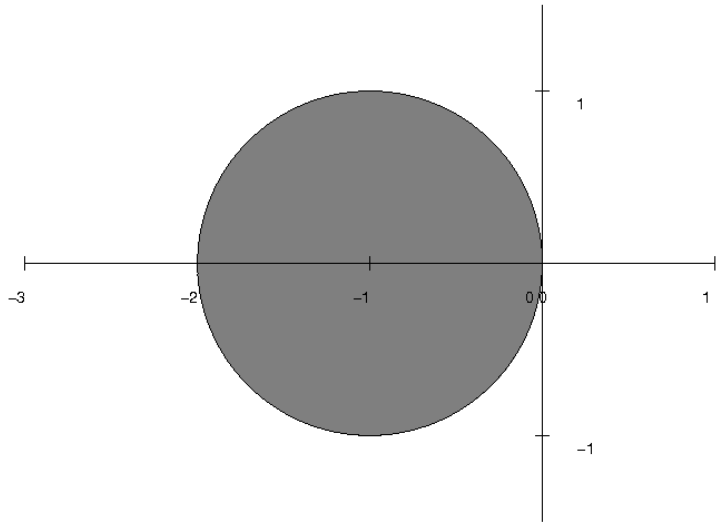


Figure 5.1: Stability region of Euler Forward

On the other hand, if we assume that the explicit part of method (5.10) is stable, then it can be derived that the implicit part is unconditionally stable. For a derivation of these results we refer to van Veldhuizen (2005).

Remark 5.3. *It is straightforward to show that steady states are returned exactly for the IMEX method (5.10).*

5.1.3 Variable Time Step Selection

Nowadays, many 'off the shelve' ODE codes and MOL solvers for PDEs integrate with variable time step sizes. Usually, users need to specify a certain (relative) tolerance and norm, and then the code automatically adjusts the time step size to the local variation in the solution to meet a certain local error tolerance in that norm. In this section we summarize the variable time step controller as it is implemented in our code. A more detailed description can be found in Hundsdorfer & Verwer (2003). More comprehensive descriptions are found in Hairer et al. (1987), Hairer & Wanner (1996) and Shampine (1994).

Consider an attempted step τ_n in time from t_n to $t_{n+1} = t_n + \tau_n$ that is performed by a p^{th} order time integration method. Suppose an estimate D_n of order $\bar{p} \leq p$ of the local truncation error is available. Define the parameter Tol as a user specified tolerance for the local error.

An attempted time step is accepted when $D_n \leq \text{Tol}$. Rejection of the time step takes place when $D_n > \text{Tol}$, and redone with a halved time step size, i.e.,

$$\tau_n \leftarrow \frac{1}{2}\tau_n. \tag{5.12}$$

In the case of an accepted time step the new time step size τ_{n+1} is computed as

$$\tau_{n+1} = r \cdot \tau_n, \tag{5.13}$$

with

$$r = \left(\frac{\text{Tol}}{D_n}\right)^{1/(\bar{p}+1)}. \tag{5.14}$$

Since estimates are used and additional control on decrease and increase of the time step size is desirable, the expression for the new (trial) time step size τ_{n+1} is implemented as

$$\tau_{n+1} = \min(r_{\max}, \max(r_{\min}, \zeta r))\tau_n. \tag{5.15}$$

In (5.15), $r_{\max}$ and $r_{\min}$ are the maximal and minimal growth factor, respectively, and $\zeta < 1$ serves to make the estimate conservative so as to avoid repeated rejections. Typical values are $\zeta \in [0.7, 0.9]$, $r_{\min} \in [0.1, 0.5]$ and $r_{\max} \in [1.5, 10]$.

Besides checking for meeting the error criteria and estimating the new time step size, we also check whether the solution is positive (componentwise) and whether Newton's method is converged. When either of these conditions is not met, we halve the time step size and redo the time step.

For each time integration method discussed further in this chapter, an estimate of the local error, which is needed for the time step controller, is given in each of the accompanying sections.

5.2 Euler Backward

In order to be self-contained, the Euler Backward method is shortly discussed in this section. The Euler Backward scheme belongs to the family of implicit Runge-Kutta schemes, is first order accurate and given as

$$w_{n+1} = w_n + \tau F(t_{n+1}, w_{n+1}). \tag{5.16}$$

It is easy to show that Euler Backward is L-stable. Further, as seen in Chapter 4, this ODE scheme is the only known scheme to be unconditionally positive.

Implementing the method of equation (5.16) is straightforward. In order to solve the system of nonlinear algebraic equations (5.16) a Newton-type method has to be implemented.

When a variable time step size controller is used, then the local truncation error for non-stiff systems is given by

$$\rho_n = -\frac{1}{2}\tau^2 w''(t_n) + O(\tau^3), \tag{5.17}$$

which can be estimated by

$$D_n = -\frac{1}{2}\|w_{n+1} - w_n - \tau F(t_n, w_n)\|. \tag{5.18}$$

For stiff systems the local truncation error is

$$\rho_n = -\frac{1}{2}\tau^2 (I - \tau A_n)^{-1} w''(t_n) + O(\tau^3), \tag{5.19}$$

with A_n an integrated Jacobian matrix. Expression (5.19) can be approximated by

$$(I - \tau \tilde{A}_n)^{-1} D_n, \quad \text{with} \quad \tilde{A}_n = F'(w_n), \tag{5.20}$$

and D_n as in expression (5.18). The solution of the linear system with matrix $(I - \tau F'(w_n))$ is relatively cheap, since the same matrix is involved in Newton's method to solve the implicit relation. Often, an LU decomposition or a good preconditioner is available. If not, then the estimator (5.18) gives good results in practice as well. For a derivation we refer to Hundsdorfer & Verwer (2003).

5.3 Rosenbrock Methods

Rosenbrock methods are linearly implicit Runge-Kutta type methods for stiff ODEs, which have proven to be effective for low to moderate accuracy for various stiff problems, see, for instance, Hairer & Wanner (1996) and Hundsdorfer & Verwer (2003). In literature different forms of these schemes have been used. These methods are named after Rosenbrock (1963), who was the first to propose schemes of this kind. Nowadays, Rosenbrock schemes are understood to solve an autonomous ODE system $w'(t) = F(w(t))$ by means of the s-stage one step formula presented in Definition 5.4.

Definition 5.4. *An s-stage Rosenbrock method is defined as*

$$k_i = \tau F\left(w_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + \tau J_F \sum_{j=1}^{i} \gamma_{ij} k_j, \quad i = 1, \ldots, s, \tag{5.21}$$

$$w_{n+1} = w_n + \sum_{i=1}^{s} b_i k_i, \tag{5.22}$$

where J_F is the Jacobian $F'(w(t))$.

The number of stages s and the coefficients b_{ij}, α_{ij} and γ_{ij} define a particular method and are selected to obtain a desired level of consistency and stability.

Remark that to compute an approximation w_{n+1} from w_n, in each stage (5.21) a linear system of algebraic equations with the matrix $(\mathbf{I} - \gamma_{ii}\tau J_F)$ has to be solved. To save computing time the coefficients γ_{ii} are taken constant, e.g., $\gamma_{ii} = \gamma$. Then, in every time-step the matrix $(\mathbf{I} - \gamma_{ii}\tau J_F)$ is identical, such that the LU factorization can be re-used. In the case that a preconditioned iterative linear solver is used, the preconditioner, e.g., an incomplete factorization of $(\mathbf{I} - \gamma_{ii}\tau J_F)$, can be re-used.

Define the coefficients β_{ij}, c_i and d_i as

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}, \quad c_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \text{and,} \quad d_i = \sum_{j=1}^{i-1} \beta_{ij}. \tag{5.23}$$

Using the coefficients β_{ij}, c_i and d_i, defined in (5.23), the order conditions for Rosenbrock schemes of order $p \leq 3$, a maximum number of stages $s \leq 4$ and $\gamma_{ii} = \gamma =$ constant can easily be derived. They are presented in Table 5.1. For a derivation of these conditions we refer to either Hairer & Wanner (1996), or Hundsdorfer & Verwer (2003).

Of particular interest is the second order Rosenbrock scheme ROS2

$$w_{n+1} = w_n + b_1 k_1 + b_2 k_2, \tag{5.24}$$

$$k_1 = \tau F(t_n, w_n) + \gamma \tau J_F k_1, \tag{5.25}$$

$$k_2 = \tau F(t_n + \alpha_{21}\tau, w_n + \alpha_{21} k_1) + \gamma_{21} \tau J_F k_1 + \gamma \tau J_F k_2, \tag{5.26}$$

Table 5.1: Order conditions of Rosenbrock methods with $\gamma_{ii} = \gamma$ for $s \leq 4$ and $p \leq 3$.

order p	order conditions
1	$b_1 + b_2 + b_3 + b_4 = 1$
2	$b_1 d_2 + b_3 d_3 + b_4 d_4 = 1/2 - \gamma$
3	$b_2 c_2^2 + b_3 c_3^2 + b_4 d_4^2 = 1/3$
	$b_3 \beta_{32} d_2 + b_4(\beta_{42} d_2 + \beta_{43} d_3) = 1/6 - \gamma + \gamma^2$

with coefficients

$$b_1 = 1 - b_2, \quad \alpha_{21} = \frac{1}{2b_2} \quad \text{and} \quad \gamma_{21} = -\frac{\gamma}{b_2}. \tag{5.27}$$

In method (5.24) - (5.26) J_F is the Jacobian of $F(t_n, w_n)$ with respect to w_n. This method is of order two for arbitrary γ as long as $b_2 \neq 0$. The stability function is given as

$$R(z) = \frac{1 + (1 - 2\gamma)z + (\gamma^2 - 2\gamma + \frac{1}{2})z^2}{(1 - \gamma z)^2}. \tag{5.28}$$

The method is A-stable for $\gamma \geq 1/4$ and L-stable if $\gamma = 1 \pm 1/2\sqrt{2}$.

5.3.1 Positivity of ROS2

By selecting for γ the larger value $\gamma_+ = 1 + 1/2\sqrt{2}$, we have the property that $R(z) \geq 0$, for all negative real z. For diffusion-reaction problems, which have negative real eigenvalues, this property ensures positivity of the solution. In the case that advection is added to the system, the matrix has eigenvalues with negative real parts and relatively small imaginary parts. Then, the positivity property is no longer guaranteed. It appears that the second order Rosenbrock method performs quite well with respect to the positivity property, as has been experienced in Verwer et al. (1999). An explanation for this behavior is lacking, but for a linearized chemical system the following property can be derived.

Consider the nonlinear chemical kinetics system $w'(t) = f(w)$, with $f(w)$ a production-loss form (see equation (4.7))

$$f(w) = P(w) - L(w)w. \tag{5.29}$$

Recall that $P(w) > 0$ contains the production terms for all species, and $L(w)w$ represents the destruction terms of all species. Suppose that for species k

the production and destruction terms are constant. Thus, we consider the k^{th} ordinary differential equation

$$w_k' = P_k - L_k w_k, \tag{5.30}$$

with $P_k \geq 0$ and $L_k \geq 0$ both constant. Applying the ROS2 scheme (5.24) - (5.26) to the ordinary differential equation (5.30) gives the approximation

$$w_k^{n+1} = R(z) w_k^n + \frac{R(z)-1}{z} \tau P_k, \quad \text{with} \quad z = -\tau L_k. \tag{5.31}$$

Hence, w_k^{n+1} can be negative when $R(z) < 0$. If, on the other hand, $0 \leq R(z) \leq 1$, then the positivity of w_k^{n+1} is guaranteed. For nonlinear systems the reasoning above does not hold. However, for species concentrations that are close to their steady state concentration, the linear reasoning comes close to what happens in the actual computation in the Rosenbrock scheme (5.24) - (5.26) to the ordinary differential equation (5.30). For further details we refer to Verwer et al. (1999).

5.3.2 Implementation Details

This section is concluded with a remark on the implementation of the second order Rosenbrock scheme (5.24) - (5.26). In our code it is implemented with the parameters

$$b_1 = b_2 = \frac{1}{2} \quad \text{and,} \quad \gamma = 1 + \frac{1}{2}\sqrt{2}. \tag{5.32}$$

The matrix-vector multiplication in the second stage of (5.26) is avoided by introducing

$$\tilde{k}_1 = k_1, \quad \text{and,} \quad \tilde{k}_2 = k_2 - k_1. \tag{5.33}$$

The ROS2 scheme is then implemented as

$$\begin{aligned} w_{n+1} &= w_n + {}^3/_2 \tilde{k}_1 + {}^1/_2 \tilde{k}_2, & (5.34)\\ \tilde{k}_1 &= \tau F(w_n) + \gamma \tau J_F \tilde{k}_1, & (5.35)\\ \tilde{k}_2 &= \tau F(w_n + \tilde{k}_1) - 2\tilde{k}_1 + \gamma \tau J_F \tilde{k}_2. & (5.36) \end{aligned}$$

5.3.3 Local Error Estimation

Note that within ROS2 the intermediate approximation

$$\tilde{w}_{n+1} = w_n + k_1, \tag{5.37}$$

is first order consistent. Since it is directly available within the solver, $\tilde{w}_{n+1}$ can be used to provide a cheap local error estimation as

$$D_n = w_{n+1} - \tilde{w}_{n+1}. \tag{5.38}$$

5.4 Backward Differentiation Formulas

In computational chemistry applications the Backward Differentiation Formulas belong to the most widely used methods to solve stiff species equations (2.18). Mainly, this is due to their favorable stability properties, but other properties also play a role. The Backward Differentation Formulas, usually denoted as BDFs, belong to the class of linear multistep methods.

Definition 5.5. *The linear k-step method is defined as*

$$\sum_{j=0}^{k} \alpha_j w_{n+j} = \tau \sum_{j=0}^{k} \beta_j F(t_{n+j}, w_{n+j}). \tag{5.39}$$

Note that the most advanced level is t_{n+k} instead of t_{n+1} and that the k past values $w_n, \ldots, w_{n+k-1}$ are used to compute w_{n+k}. The method (5.39) is explicit when $\beta_k = 0$, and implicit otherwise.

The order conditions for the *k*-step linear multistep method (5.39) are summarized as:

The method (5.39) is of order p if and only if

$$\sum_{j=0}^{k} \alpha_j = 0, \quad \sum_{j=0}^{k} \alpha_j^i = i \sum_{j=0}^{k} \beta_j j^{i-1} \quad \text{for} \quad i = 1, 2, \ldots, p. \tag{5.40}$$

These conditions are easily derived by Taylor series expansion, see, for instance, Hundsdorfer & Verwer (2003) and Hairer & Wanner (1996).

Definition 5.6. *The k-step Backward Differentation Formulas, usually called BDFs, are implicit linear multistep methods. For the coefficients β_j, $j = 0, \ldots, k$, holds that*

$$\beta_k = 1 \quad \text{and} \quad \beta_j = 0 \quad \text{for} \quad j = 0, \ldots, k-1. \tag{5.41}$$

The coefficients α_j, $j = 0, \ldots, k$, are chosen such that the order is optimal, which is k for a k-step BDF method.

The 1-step BDF method is Euler Backward,i.e.,

$$w_{n+1} - w_n = \tau F(t_{n+1}, w_{n+1}). \tag{5.42}$$

Applying the order conditions (5.40) for second order accuracy lead to the BDF-2 method

$$\frac{3}{2} w_{n+2} - 2w_{n+1} + \frac{1}{2} w_n = \tau F(t_{n+2}, w_{n+2}), \tag{5.43}$$

whereas the order conditions for third order accuracy gives the BDF-3

$$\frac{11}{6} w_{n+3} - 3w_{n+2} + \frac{3}{2} w_{n+1} - \frac{1}{3} w_n = \tau F(t_{n+3}, w_{n+3}). \tag{5.44}$$

Of practical interest is the BDF-2 method, since for this method positivity conditions can be derived.

5.4.1 Stability for BDFs

Unlike Runge-Kutta methods, there are not many A-stable linear multistep methods. Dahlquist (1963) derived that an A-stable linear multistep method is of order equal to or less than two. This result is also known as the second Dahlquist barrier.

Indeed it can be derived that the BDF-1 and BDF-2 methods are A-stable, see Hundsdorfer & Verwer (2003) and Hairer & Wanner (1996). For $k > 2$ up to $k = 6$ the BDF methods are $A(\alpha)$-stable, which means that the set

$$\{z \in \mathbb{C} : z = 0, \infty \quad \text{or} \quad |\arg(-z) \leq \alpha|\} \tag{5.45}$$

is contained in the stability region of that particular method. For BDF-3, BDF-4, BDF-5 and BDF-6 the angle α in degrees depends on k as:

k	3	4	5	6
α	86°	73°	51°	17°

The BDF methods are unstable for $k > 6$, see Hairer et al. (1987). For $3 \leq k \leq 6$ the stability regions of the BDF-k methods are illustrated in Figure 5.2.

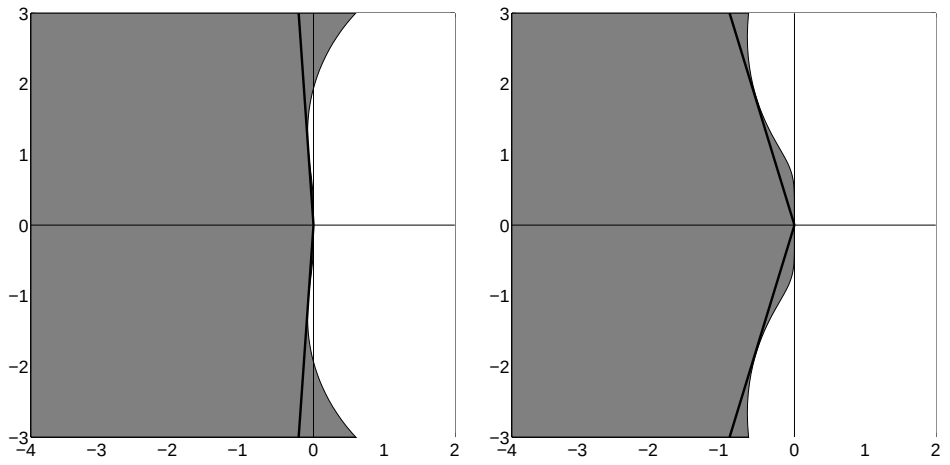
Popularity of the BDF methods is due to the good absolute stability properties. At infinity, the stability properties are surpassed since the zeros of the stability polynomial

$$\sum_{j=0}^{k} \alpha_j z^j + \beta_k \tau\lambda z^k, \tag{5.46}$$

tend to zero when $|\tau\lambda| \to \infty$. One of the roots of polynomial (5.46) approximates $e^{\tau\lambda}$ up to order $p + 1$ for $|\tau\lambda| \to 0$. That particular root is called the principle root and the remaining $(k-1)$ roots are called spurious roots. Thus, for BDF methods applied to stiff problems, the $(k-1)$ spurious roots do not cause oscilatory behavior of the solution. This means that the time step size τ can be increased without any risk of generating spurious oscillations.

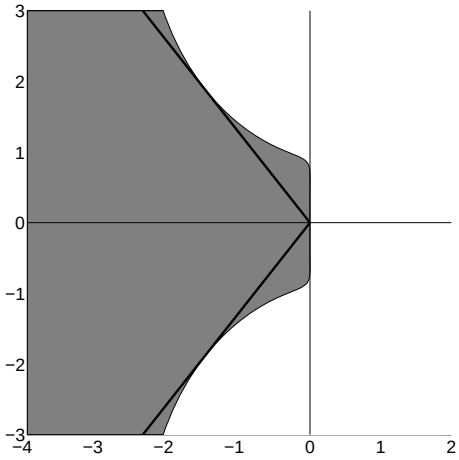
5.4.2 Implementation

When implementing a linear multistep scheme, and in particular a BDF-k scheme, one has to take into account that the first $(k-1)$ approximations cannot be computed with this scheme. Specifically, one also has to take care that the starting procedure returns stable solutions. A possible, and often used, solution is to use the BDF-1 scheme to compute w_1, the BDF-2 scheme to compute $w_2, \ldots$, and the BDF-$(k-1)$ scheme to compute w_{k-1}. Another solution is to compute $w_1, \ldots, w_{k-1}$ by means of a Runge-Kutta method.

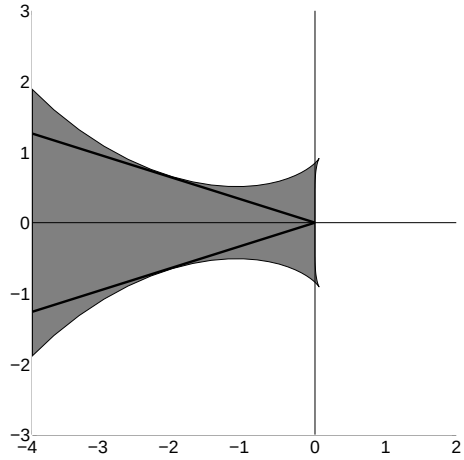


(a) BDF-3

(b) BDF-4



(c) BDF-5



(d) BDF-6

Figure 5.2: Stability regions for BDF-k, $3 \leq k \leq 6$. The boundaries of the $A(\alpha)$-stability regions are illustrated by the bold lines.

5.4.3 Positivity

As for Runge-Kutta methods, the requirement of positivity does place a severe time step size restriction on BDF methods. For the BDF-2 methods we will derive conditions for which positivity is ensured. The BDF-2 scheme (5.43) can be rewritten as

$$w_{n+2} - \frac{2}{3}\tau F(t_{n+2}, w_{n+2}) = \frac{4}{3}w_{n+1} - \frac{1}{3}w_n. \tag{5.47}$$

Based on the analysis carried out in Hundsdorfer et al. (2003) for the Total Variation Diminishing property of BDF-2 type schemes, positivity of the BDF-2 method (5.47) will be considered.

Since no claims can be made on the positivity of the right hand side of formula (5.47), no conditions for positivity of w_{n+2} can be obtained. In order to circumvent this, the BDF-2 scheme (5.47) will be rewritten up to and including its starting values w_1 and w_0. Let $\theta \geq 0$ be a parameter, which will be specified later. Then, for $n \geq 3$, formula (5.47) can be written as

$$\begin{aligned} w_{n+2} - \frac{3}{2}\tau F(t_{n+2}, w_{n+2}) = \\ \left(\frac{4}{3} - \theta\right)w_{n+1} + \theta\frac{2}{3}\tau F(t_{n+1}, w_{n+1}) + \left(\theta\frac{4}{3} - \frac{1}{3}\right)w_n + \theta\frac{1}{3}w_{n-1}. \end{aligned} \tag{5.48}$$

Continuing this way of subtracting and adding $\theta^j w_{n-j}$ and substituting in formula (5.47) gives for $n \geq 3$

$$\begin{aligned} w_{n+2} - \frac{3}{2}\tau F(t_{n+2}, w_{n+2}) = \left(\frac{4}{3} - \theta\right)w_{n+1} + \theta\frac{2}{3}\tau F(t_{n+1}, w_{n+1}) + \\ \sum_{j=4}^{n} \theta^{j-4}\left(\left(-\frac{1}{3} + \theta\frac{4}{3} - \theta^2\right)w_{n-j+2} + \theta^2\frac{2}{3}\tau F(t_{n-j+2}, w_{n-j+2})\right) \\ \theta^{n-3}\left(\left(-\frac{1}{3} + \theta\frac{4}{3}\right)w_1 - \theta\frac{1}{3}w_0 + \theta\frac{2}{3}\tau F(t_0, w_0)\right). \end{aligned} \tag{5.49}$$

We assume that $w_0 \geq 0$, and that w_1 is computed via an appropriate starting procedure such that $w_1 \geq 0$. Further, assume as well that

$$\left(-\frac{1}{3} + \theta\frac{4}{3}\right)w_1 - \theta\frac{1}{3}w_0 + \theta\frac{2}{3}\tau F(t_0, w_0) \geq 0. \tag{5.50}$$

Then, by applying Condition 4.5, w_{n+2} is positive when the right-hand side of equation (5.49) is positive. By inequality (5.50) we have that the right-hand side of equation (5.49) is positive if and only if the scaled Euler Forward steps

$$\left(\frac{4}{3} - \theta\right)\left(w_{n+1} - \tau\frac{\frac{2}{3}\theta}{\frac{4}{3} - \theta}F(t_{n+1}, w_{n+1})\right), \tag{5.51}$$

and

$$\left(-\frac{1}{3}+\frac{4}{3}\theta-\theta^2\right)\left(w_{n-j+2}-\tau\frac{\frac{2}{3}\theta^2}{-\frac{1}{3}+\frac{4}{3}\theta-\theta^2}F(t_{n-j+2},w_{n-j+2})\right), \tag{5.52}$$

are positive. Thus, expressions (5.51) and (5.52) are positive when $\tau \leq r(\theta)\tau_{EF}$, where τ_{EF} is the time step size such that Euler Forward is positive (see Condition 4.4), and

$$r(\theta) = \min\left(\frac{\frac{4}{3}-\theta}{\frac{2}{3}\theta}, \frac{-\frac{1}{3}+\frac{4}{3}\theta-\theta^2}{\frac{2}{3}\theta^2}\right). \tag{5.53}$$

For practical use it is useful to find the maximum of $r(\theta)$. After some tedious computations we obtain that

$$\max_{\theta\geq 0} r(\theta) = \frac{1}{2}. \tag{5.54}$$

Thus, we find that BDF-2 is positive, when $\alpha\tau \leq 1/2\tau_{EF}$. This means that the implicit and unconditionally stable BDF-2 scheme has a time step size restriction which is 2 times tighter than for Euler Forward in order to be positive.

5.4.4 Local Error Estimation

Multi-step methods use information from at least two previous time levels. When using variable time step sizes, the formula coefficients need to be adjusted for maintaining the order consistency. Let $\tau_{n+1} = t_{n+2} - t_{n+1}$ and

$$r = \frac{\tau_{n+1}}{\tau_n}. \tag{5.55}$$

The variable step size version of the BDF-2 scheme is then given as

$$w_{n+2} - \frac{(1+r)^2}{1+2r}w_{n+1} + \frac{r^2}{1+2r}w_n = \frac{1+r}{1+2r}\tau F(t_{n+2},w_{n+2}). \tag{5.56}$$

For a derivation of this scheme we refer to Hairer et al. (1987). Following Hundsdorfer & Verwer (2003), the following first order local error estimator is available

$$D_n = \frac{1+r}{1+2r}\left(w_{n+2} + (r^2-1)w_{n+1} - r^2 w_n - (1+r)\tau_n F(t_{n+1},w_{n+1})\right). \tag{5.57}$$

For the first step the Euler Backward method with its first order local error estimator

$$D_0 = \frac{1}{2}\left(w_1 - w_0 - \tau_0 F(t_0,w_0)\right), \tag{5.58}$$

is used.

5.5 IMEX Runge-Kutta Chebyshev Methods

The IMEX extension of the class of Runge-Kutta Chebyshev (RKC) methods, developed by Verwer and co-workers (see Verwer & Sommeijer (2004) and Verwer et al. (2004)), is designed to solve stiff systems of ODEs that represent semi-discrete advection-diffusion-reaction equations. In this IMplicit-EXplicit RKC integration method the advection and diffusion terms are treated simultaniously and explicitly, whereas the highly stiff reaction terms are integrated implicitly.

The RKC methods belong to the class of stabilized explicit Runge-Kutta methods. Whereas the principal goal of Runge-Kutta methods is to achieve the highest order of accuracy possible for a given number of stages s, stabilized explicit RK methods use a few stages to achieve a low order of accuracy such that the additional stages are exploited to increase the stability region. The RKC method is stable on a segment of the negative real axis, which is bounded by the origin and the stability boundary $\beta(s)$.

Definition 5.7. *The stability boundary $\beta(s)$ is the number $\beta(s)$ such that $[-\beta(s), 0]$ is the largest segment of the negative real axis contained in the stability region*

$$\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \leq 1\}.$$

The method has a greater applicability when this strip is wider, but for diffusion dominated flow problems larger stability bounds $\beta(s)$ are more important. For the RKC method discussed in this section the size of $\beta(s)$ increases quadratically in s.

In this thesis we will not go into the full details on the construction of the RKC scheme. We only consider schemes of order 2, because they are believed to be more efficient than the first order schemes, see Verwer & Sommeijer (2004).

First, we will give the scheme, and thereafter we briefly discuss its stability function. The second order explicit RKC formula has the form

$$\begin{aligned}
w_{n0} &= w_n, && (5.59)\\
w_{n1} &= w_n + \tilde{\mu}_1 \tau F(t_n, w_{n0}), && (5.60)\\
w_{nj} &= (1 - \mu_j - \nu_j) w_n + \mu_j w_{n,j-1} + \nu_j w_{n,j-2} + \\
&\quad + \tilde{\mu}_1 \tau F(t_n + c_{j-1}\tau, w_{n,j-1}) + \tilde{\gamma}_j \tau F(t_n, w_{n0}), && (5.61)\\
w_{n+1} &= w_{ns}, && (5.62)
\end{aligned}$$

with $j = 2, \ldots, s$. For $s \geq 2$ all coefficients in formulas (5.59) - (5.62) are available in analytical expressions as:

$$\omega_0 = 1 + \frac{\varepsilon}{s^2}, \quad \omega_1 = \frac{T_s'(\omega_0)}{T_s''(\omega_0)}, \tag{5.63}$$

$$b_j = \frac{T_j''(\omega_0)}{(T_j'(\omega_0))^2}, \quad c_j = \frac{T_s'(\omega_0)}{T_s''(\omega_0)} \frac{T_j''(\omega_0)}{T_j'(\omega_0)} \approx \frac{j^2-1}{s^2-1}, \tag{5.64}$$

$$\tilde{\mu}_1 = b_1\omega_1, \quad \mu_j = \frac{2b_j\omega_0}{b_{j-1}}, \quad \nu_j = -\frac{b_j}{b_{j-2}}, \tag{5.65}$$

$$\tilde{\mu}_j = \frac{2b_j\omega_1}{b_{j-1}}, \quad \tilde{\gamma}_j = -a_{j-1}\tilde{\mu}_j, \quad \text{and} \quad a_j = 1 - b_j T_j(\omega_0). \tag{5.66}$$

In formulas (5.63), (5.64) and (5.66) $T_j(z)$ is the j^{th} order Chebyshev polynomial of the first kind. For $z \in \mathbb{C}$ it is recursively defined as

$$T_j(z) = 2zT_{j-1}(z) - T_{j-2}(z), \tag{5.67}$$

with $T_0(z) = 1$ and $T_1(z) = z$. The stability function of the RKC scheme (5.59) - (5.62) is

$$\mathcal{R}(z) = a_s + b_s T_s(\omega_0 + \omega_1 z). \tag{5.68}$$

For derivations of the stability function and the method itself we refer to Sommeijer et al. (1997), Hundsdorfer & Verwer (2003), Verwer & Sommeijer (2004) and Verwer et al. (2004).

The parameter $\varepsilon \geq 0$ in ω_0 in formula (5.63) is called a damping parameter. For $\varepsilon = 0$ stability regions as in Figure 5.3(a) are obtained. For practical reasons a stability region as in Figure 5.3(a) is undesirable, i.e., along the negative real axis we do not have that $\mathcal{R}(z)$ is strictly less than one. The corresponding stability bound $\beta(s)$ is

$$\beta(s) = \frac{3}{2}(s^2 - 1), \tag{5.69}$$

see Verwer & Sommeijer (2004). In van der Houwen (1996) it can be found that for second order stability polynomials the optimal stability bound $\beta(s)$ increases quadratically with s as s increases by means of the approximation

$$\beta(s) = 0.82s^2. \tag{5.70}$$

Thus, for $\varepsilon = 0$ the stability polynomial (5.68) generates about 80% of the optimal stability bound (5.70).

For $\varepsilon > 0$ we see that along the negative real axis the strip becomes wider. This is clearly illustrated in Figure 5.3(b) for $s = 5$ and $\varepsilon = 0.1$. Under the assumption that ε is small the real stability bound $\beta(s)$ is given as

$$\beta(s) = \frac{3}{2}(s^2 - 1)\left(1 - \frac{2}{15}\varepsilon\right). \tag{5.71}$$

For more details we refer to Verwer & Sommeijer (2004).

The IMEX extension of the above scheme is as follows. Suppose we have an ODE system $w'(t) = F(t, w(t))$, where $F(t, w)$ can be split as

$$F(t, w) = F_E(t, w) + F_I(t, w) \tag{5.72}$$

with $F_I(t,w)$ the part of $F(t,w)$ which is too stiff to be integrated explicitly, i.e., in our case the reaction terms. The term $F_E(t,w)$ is the moderately stiff part of F that can be integrated explicitly by the RKC method, which are the advection and diffusion terms. Then, the IMEX extension of the explicit RKC scheme (5.59) - (5.62) from Verwer et al. (2004) reads

$$
\begin{aligned}
w_{n0} &= w_n, && (5.73)\\
w_{n1} &= w_n + \tilde{\mu}_1\tau F_E(t_n + c_0\tau, w_{n0}) + \tilde{\mu}_1\tau F_I(t_n + c_1\tau, w_{n1}), && (5.74)\\
w_{nj} &= (1-\mu_j-\nu_j)w_n + \mu_j w_{n,j-1} + \nu_j w_{n,j-2} && (5.75)\\
&\quad +\tilde{\mu}_j\tau F_E(t_n + c_{j-1}\tau, w_{n,j-1}) + \tilde{\gamma}_j\tau F_E(t_n + c_0\tau, w_{n0})\\
&\quad +(\tilde{\gamma}_j - (1-\mu_j-\nu_j)\tilde{\mu}_1)\tau F_I(t_n + c_0\tau, w_{n0})\\
&\quad -\nu_j\tilde{\mu}_1\tau F_I(t_n + c_{j-2}\tau, w_{n,j-2}) + \tilde{\mu}_1\tau F_I(t_n + c_j\tau, w_{nj}) && (5.76)\\
w_{n+1} &= w_{ns}. && (5.77)
\end{aligned}
$$

Note that the highly stiff part $F_I(t,w)$ of $F(t,w)$ is treated implicitly. If the stiff reaction term $F_I(t,w)$ is absent, then the explicit scheme (5.59)-(5.62) is recovered.

For the IMEX-RKC scheme the implicit part is unconditionally stable as long as the eigenvalues of the Jacobian of $F_I(t,w)$ are real, whereas the stability condition for the explicit part remains unchanged, see Verwer & Sommeijer (2004).

Steady states are returned exactly, which is not true for other operator splittings, see Hundsdorfer & Verwer (2003). Unconditional positivity is not guaranteed; the exact condition is not known to the author.

5.5.1 Implementation details

We conclude with some remarks on the implementation of the IMEX-RKC solver. In each of the s stages in the IMEX-RKC scheme (5.73) - (5.77) a system of nonlinear algebraic equations

$$w_{nj} - \tilde{\mu}_1\tau F_I(t_n + c_j\tau, w_{nj}) = v_j, \tag{5.78}$$

with v_j given and w_{nj} a vector of unknowns, has to be solved. For efficiency reasons it is beneficial that $\tilde{\mu}_1$ is independent of j. Further, a modified Newton iteration is used to solve the nonlinear equation (5.78), where as initial guess w_{n0} is taken. Consequently, the Jacobian of $F_I(t_n, w_{n0})$ has to be computed once per time step. The LU factorization needed within the modified Newton iteration is then identical over the s stages.

With respect to the Jacobian of the left-hand side of equation (5.78) the following can be noted. The reaction terms have no underlying spatial grid connectivity. If the unknown species concentrations are ordered per grid point, then this Jacobian consists of a number (i.e., the number of grid

points) of decoupled small sized subsystems with dimension equal to the number of species. The LU factorization of such a matrix is easily obtained.

Details of the variable time step controller, which tests the current solution for accuracy and the explicitly integrated part for stability, can be found in Verwer et al. (2004). This controller also adjusts the number of stages s, depending on the time step size and the conditions for stable integration of advection and diffusion (called IRKC(full)) or for stable integration of diffusion only (called IRKC(fly)).

Conditions for stable explicit integration of advection and diffusion are obtained via von Neumann stability analysis. The approach of Wesseling (1996) is followed in which time step size conditions are given to guarantee that the eigenvalues emerging from van Neumann stability analysis to lie inside geometric figures like squares, ellipses and ovals. This approach is described in Wesseling (1996) and in Chapter 5 of Wesseling (2001). For the explicit integration of advection and diffusion via the RKC method ovals are used, because they give a better fit near the origin. For $s = 5$ an inscribed oval in the stability region of the RKC method is illustrated in Figure 5.4. Technical details can be found in Verwer et al. (2004).

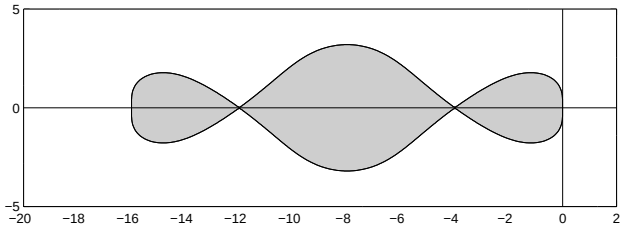
5.5.2 Local Error Estimation

Shampine et al. (2005) comprehensively describe how the local error estimation is derived and implemented in the computer code.

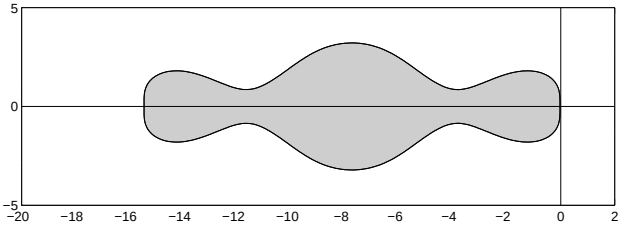
5.6 Numerical Results

The ODE methods presented in the above section have been tested on the benchmark problem of Kleijn (2000). All specific details of this two dimensional Chemical Vapor Deposition process are discussed in Chapter 8. The number of gaseous species in the used chemistry model for this Chemical Vapor Deposition process is 17, of which 16 participate in the reaction mechanism consisting of 26 gas-phase reactions. Furthermore, the surface chemistry reaction model with 14 surface reactions as described in Section 8.1.2 is included. The reactor configuration for all simulations is described in Section 8.2.1 and illustrated in Figure 8.1. Because of axisymmetry, the computational domain is two-dimensional.

The simulation runs from the the instant that the reactor is completely filled with helium carrier gas and a mixture of helium and silane starts to enter the reactor, until steady state. The spatial computational grid consists of 35 equidistant grid points in radial direction, and 32 non-equidistant grid points in axial direction. The grid spacing in axial direction gradually decreases towards the wafer surface. In our experiments steady state is



(a) The undamped case



(b) The damped case with $\varepsilon = 0.1$

Figure 5.3: Stability regions of the second order shifted Chebyshev polynomial (5.68) with $s = 5$.

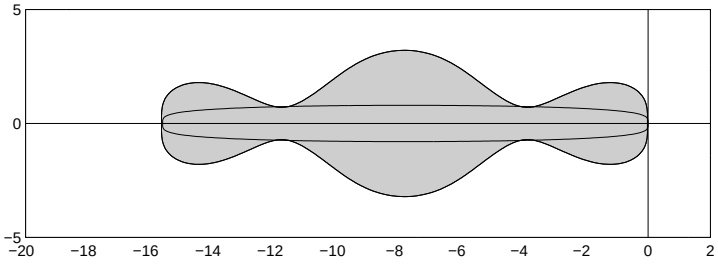


Figure 5.4: Stability region of (5.61) with inscribed oval

assumed to be obtained when for a certain time step t_n the inequality

$$\frac{||w_{n+1} - w_n||_2}{||w_n||_2} \leq 10^{-6}, \tag{5.79}$$

holds, where w_n is the numerical solution of the semi-discretization $w'(t) = F(t, w)$ on time $t = t_n$. The validation and interpretation of the results is done in Chapter 8. For now, we are only interested in the performance of the various ODE integrators.

We compared the unconditionally positive Euler Backward method, the ROS2 scheme, the conditionally positive BDF-2 and the IMEX RKC scheme. We have to remark that the BDF-2 scheme is implemented in such a way that when negative solutions are obtained, first the time step is halved and redone. If it then again returns negative species solutions, then it switches back to the BDF-1 scheme. After a succesful BDF-1 step it tries to return to the BDF-2 scheme.

In both the Euler Backward and the BDF solver, a system of nonlinear algebraic equations has to be solved for each time step. We have done experiments with both a full Newton method, and a modified Newton method. In Full Newton the Jacobian is evaluated in each Newton iteration. If the initial guess is in a neighborhood of a solution, then quadratric convergence is obtained.

In the Euler Backward and/or BDF solver it is also possible to update the Jacobian occasionally, such that Newton's method becomes a modified Newton iteration. Define the convergence rate of the Newton iteration as

$$\Theta_n = \frac{||d_n||}{||d_{n-1}||}, \quad n \geq 1, \tag{5.80}$$

where d_n is the Newton update. Then, we do not recompute the Jacobian in the next time step when (i) the Newton process converges in one iteration, or, (ii) the convergence rate in the last Newton iteration was very small, e.g., $\Theta_n \leq 10^{-3}$, which means that the last Newton iteration gives fast convergence, see also, for instance, Hairer & Wanner (1996). By updating the Jacobian only occasionally, the quadratic convergence behavior of the full Newton iteration is lost; usually linear convergence is retained.

The linear systems are solved directly by means of an LU factorization of the Jacobian-matrix. To reduce the amount of work to factorize the Jacobian, the unknown species mass fractions are ordered per grid point. As has been pointed out in van Veldhuizen et al. (2007b), this ordering gives the smallest bandwith in the Jacobian. As is commonly known, the smaller the bandwidth is, the less work is needed to compute the LU-factors in the Jacobian-matrix.

In Table 5.2, 5.3 and 5.4 numerical results are given for the various time integration methods, with either the full or modified Newton iteration to

solve the nonlinear systems. Also shown are the relative errors in the L_2 norm with respect to a time accurate ODE solution, on some fixed times. We used relative errors, because the solution contains relatively small components. The user-specified quantity TOL (see Section 5.1.3) to monitor the local truncation error is taken equal to 10^{-3}. For the time accurate ODE solution this value was set to 10^{-6}. We observe that for the global errors as presented in Table 5.2, 5.3 and 5.4, the behavior is as expected.

For the unconditionally positive Euler Backward time integration scheme the modified Newton (see above) influences the positivity of the solution, i.e., the number of rejected time steps due to negative species increases (compare Tables 5.2 and 5.3), from 1 to 31. Rejected time steps due to negative entries in the solution vector should be redone with smaller time steps, resulting in a larger number of F evaluations (the number of Jacobian evaluations is approximately equal). Thus, as a result of an increasing number of Newton iterations, the total computational costs increase.

For the BDF2 scheme (compare Tables 5.2 and 5.3), application of modified Newton strategy, as explained above, gives more satisfying results. From Table 5.3 it can be concluded that for BDF2 an increasing number of cheaper Newton iterations is computationally cheaper than factorizing the Jacobian in every Newton iteration.

With respect to the other higher order time integration schemes (see Table 5.4), we note the following. ROS2 is the cheapest higher order time integrator for this Chemical Vapor Deposition process. For the IMEX-RKC scheme we see that both versions perform equally well. Since there is no gain in efficiency by using 'on the fly' stability conditions for the explicit part, the more robust fully CFL-protected IMEX-RKC(full) is preferred.

With respect to positivity of the solution during transient simulations we note the following. Omission of the reacting surface and thermal diffusion in the reaction Jacobian gives very poor Newton convergence. We also observed that in this case the solution conserves positivity for very small time steps only, even for Euler Backward. We conclude that for this Chemical Vapor Deposition problem it is required to use the exact Jacobian, in which also the derivatives of the reacting surface and thermal diffusion are included.

From the integration statistics presented in Tables 5.2 - 5.4 it is concluded that for long time steady state simulations Euler Backward is, in spite of its first order accuracy, the most efficient time integrator. In van Veldhuizen et al. (2008b) it is concluded that the unconditional postivity of Euler Backward is preferred over the conditional higher order methods present in this section.

The conclusions drawn in van Veldhuizen et al. (2008b) are based on time accurate numerical simulations from inflow conditions until steady state. For highly accurate time dependent simulations over a smaller time frame the integration statistics are different. Again, the Euler Backward

Table 5.2: Integration statistics for EB and BDF-2, with full Newton solver

Number of	EB	BDF-2
F	190	757
F'	94	417
Linesearch	11	0
Newton iters	94	417
Rej. time steps	1	10
Acc. time steps	38	138
CPU Time	6500	30500
Relative error on $t = 1.6$ s	$6.8 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
on $t = 3.2$ s)	$7.9 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$

Table 5.3: Integration statistics for EB and BDF2, with modified Newton.

Number of	EB	BDF-2
F	720	1786
F'	84	163
Linesearch	39	33
Newton iters	463	1441
Rej. time steps	31	33
Acc. time steps	88	121
CPU Time	10800	17000
Relative error on $t = 1.6$ s	$6.8 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
on $t = 3.2$ s	$7.9 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$

Table 5.4: Integration statistics for ROS2, IRKC(fly), where stability for the explicitly integrated part is tested for diffusion only, and IRKC(full), where stability conditions are forced for both advection and diffusion, schemes.

Number of	ROS2	IRKC(fly)	IRKC(full)
F	424	429662	427911
F'	142	2005	2008
Linesearch	0	50	30
Newton iters	0	17425	17331
Rej. time steps	2	729	728
Acc. time steps	140	1276	1284
CPU Time	8000	20000	19500
Relative error on $t = 1.6$ s	$1.1 \cdot 10^{-3}$	$1.8 \cdot 10^{-3}$	
on $t = 3.2$ s	$2.5 \cdot 10^{-4}$	$8.3 \cdot 10^{-5}$	

method, with full Newton, is computationally the cheapest. However, the IMEX-RKC schemes performs much better than the other higher order time integration schemes. Its computationally cheaper time steps are in this case paying off compared to the more expensive time steps of BDF2 and ROS2. The integration statistics are summarized in Tables 5.5 and 5.6.

Table 5.5: Integration statistics over a small, purely transient, time frame for EB and BDF-2, with full Newton solver

Number of	EB	BDF-2
F	143	259
F'	66	148
Linesearch	8	0
Newton iters	66	148
Rej. time steps	1	7
Acc. time steps	31	52
CPU Time	1250	2750

Table 5.6: Integration statistics over a small, purely transient, time frame for ROS2, IRKC(fly), where stability for the explicitly integrated part is tested for diffusion only, and IRKC(full), where stability conditions are forced for both advection and diffusion, schemes.

Number of	ROS2	IRKC(fly)	IRKC(full)
F	350	21140	20500
F'	139	207	199
Linesearch	0	21	20
Newton iters	0	5319	5163
Rej. time steps	2	68	65
Acc. time steps	137	139	134
CPU Time	3000	2500	2450

For three-dimensional simulations the IMEX-RKC scheme is an attractive alternative to the unconditionally positive Euler Backward. Despite its conditional positivity, its advantage over the other ODE methods is its efficiency which is independent of the number of spatial dimensions. The dimensions of the linear systems appearing in the IMEX-RKC scheme do not change when going from two to three spatial dimensions. For all other ODE methods the dimension of the linear systems to be solved changes when going up in the number of spatial dimensions. In particular, the linear systems within fully implicit schemes are expensive to solve for three-dimensional problems. Iterative linear solvers are indispensable in that case.

The remaining chapters in this thesis will be devoted on the design of the Euler Backward solver. In particular, attention is paid to the robustness, and thus positivity, of the solver, and, of course, the reduction of computational costs.

CHAPTER 6

Solving the Nonlinear Equations: Inexact Newton Methods

The huge stiffness of the species equations (2.18) causes that (part of) the time integration should be done implicitly. For most ODE schemes, except for the Rosenbrock schemes, one or more systems of nonlinear algebraic equations have to be solved in each time step. In this chapter a system of nonlinear algebraic equations is denoted as

$$F(x) = 0. \tag{6.1}$$

In equation (6.1) F is assumed to be a continuously differentiable vector function $F : \mathbb{R}^n \to \mathbb{R}^n$, with $n \geq 1$. A classical algorithm to solve a system of nonlinear algebraic equations (6.1) is Newton's method, which is presented as Algorithm 1. Newton-based methods have been within the applied mathematics community the dominating approach to solve nonlinearly implicit PDEs. On the other hand, in the computational physics and computational fluid dynamics communities the emphasis is more on Picard-type linearizations, and splitting per equation or coordinate direction. This latter approach often allows a splitting error to remain in time, and little attention is paid to the nonlinear residual within a time step.

As remarked in Knoll & Keyes (2004), more recently computational scientists are taking a deeper look at the resulting errors in these splitting methods and resulting errors. As a result, the computational physics community is driven towards nonlinear multigrid methods, see, for instance, Wesseling (1992), and Newton-based methods, see, for instance, Kelley (1995) and Knoll & Keyes (2004).

Algorithm 1: Newton's method

Let x_0 be given.
for $k = 1, 2, \ldots$ until 'convergence' **do**
  Solve $F'(x_k)s_k = -F(x_k)$.
  Set $x_{k+1} = x_k + s_k$.
end for

The major strength of the classical Newton method is its local convergence property. If x_0 is sufficiently close to a solution x_*, then the Newton sequence $\{x_n\}$ generated by Algorithm 1 converges superlinearly to x_*. Further, under the assumptions that the Jacobian $F'(x_*)$ is nonsingular and F is Lipschitz continuous at x_* quadratic convergence is obtained. For a proof we refer to Ortega & Rheinboldt (2000).

In the classical Newton method there are mainly two difficulties. First, for large n the computation of the 'exact' Jacobian $F'(x_k)$ at iteration k can be expensive. Alternatives considered in literature are, for instance,

- computing the Jacobian $F'(x_k)$ numerically by means of finite differences, see , for instance, Kelley (2003),
- Broyden's method, which is a rank one update secant method, see, for instance, Ortega & Rheinboldt (2000), and,
- Jacobian-Free Newton-Krylov methods, in which the linear systems are solved using Krylov Subspace methods where the Jacobian matrix-vector product $F'(u)v$, with u and v arbitrary vectors, is approximated as
$$F'(u)v = \frac{F(u + \varepsilon v) - F(u)}{\varepsilon}, \tag{6.2}$$
with ε a small number. A nice survey is found in Knoll & Keyes (2004).

The second difficulty in the classical Newton method is to solve the so-called Newton equation
$$F'(x_k)s_k = -F(x_k), \tag{6.3}$$
in each nonlinear iteration. In practice, when n is usually large, solving (6.3) exactly can be expensive or even infeasable. Computing an exact solution of the Newton equation (6.3) may in particular not be justified when the k^{th} iterate x_k is far from a solution x_*. In that case it makes more sense to compute an approximation of the Newton update s_k.

The extension of the classical Newton methods by allowing computed approximations of the solution of the Newton equation (6.3) are called Inexact Newton methods. In Section 6.1 this class of Inexact Newton methods is discussed. Section 6.3 is devoted to global convergence properties of Newton's method.

Under the assumption that a time integration method is positivity conserving, it is generally not guaranteed that the nonlinear solutions are positive. In Section 6.4 a projected Newton method is introduced to overcome this difficulty. Its properties are discussed in Section 6.4.

6.1 Inexact Newton Methods

Instead of solving the Newton equation

$$F'(x_k)s_k = -F(x_k), \tag{6.4}$$

exactly, the Newton step s_k in Inexact Newton solvers is approximated by an iterative linear solver. In our case a preconditioned Krylov Subspace method is used. In Chapter 7 the linear solvers used are discussed. The approximated Newton step s_k has to satisfy the so-called Inexact Newton condition

$$||F(x_k) + F'(x_k)s_k|| \leq \eta_k ||F(x_k)||, \tag{6.5}$$

for a certain 'forcing term' $\eta_k \in [0, 1)$. In general form, the algorithm is presented as Algorithm 2. Note that the Inexact Newton condition (6.5) expresses

1. a certain reduction in the norm of $F(x_k) + F'(x_k)s_k$, which is the local linear model of F in a neighborhood of x_k, and,

2. a certain (relative) accuracy in solving the Newton equation $F'(x_k)s_k = F(x_k)$ by means of an iterative linear solver.

Algorithm 2: Inexact Newton

Let x_0 be given.
for $k = 1, 2, \ldots$ until 'convergence' **do**
  Find some $\eta_k \in [0, 1)$ and s_k that satisfy

$$||F(x_k) + F'(x_k)s_k|| \leq \eta_k ||F(x_k)||.$$

  Set $x_{k+1} = x_k + s_k$.
end for

Of course, the local convergence behavior of the inexact Newton method depends on the sequence of forcing terms η_k. The intuitive idea that smaller values of the forcing terms leads to fewer Newton iterations is illustrated in Dembo et al. (1982). Under the natural assumption that the sequence of forcing terms is uniformly less than one and that x_0 is sufficiently close to

x_*, Dembo et al. (1982) showed that a sequence of inexact Newton iterates $\{x_k\}$ converges linearly to x_*. Further, if

$$\lim_{k\to\infty} \eta_k = 0, \tag{6.6}$$

then $\{x_k\}$ converges superlinearly to x_*. In the case that F' is Lipschitz continuous at x_* and

$$\eta_k = O\|F(x_k)\|, \tag{6.7}$$

then the convergence is quadratically.

However, away from the solution, the function F and its local linear model may disagree considerably at a step that closely approximates the Newton step. When choosing η_k too small, this can lead to *oversolving* the Newton equation; meaning that imposing an accurate linear solution to an inaccurate Newton correction may result in a poor Newton update, and, therefore, little or no progress towards a solution. The latter has been experienced in, for instance, Shadid et al. (1997) and Tuminaro et al. (2002). Moreover, for Newton solvers with forced global convergence algorithms, like line-search (or backtracking), in which additional accuracy in solving the Newton equation requires additional expense, it may entail pointless costs. Then, a less accurate approximation of the Newton step is cheaper, and probably more effective.

6.2 Choosing the forcing term

In the literature several choices for the forcing term have been proposed. In this section we present the ones proposed by Eisenstat & Walker (1996). In their paper Eisenstat & Walker (1996) aimed to come up with forcing terms that achieve desirable fast convergence and tend to avoid oversolving. For a broader comparison other choices from literature are included. We start with the first forcing term.

6.2.1 Choice 1

The first choice, taken from Eisenstat & Walker (1996),is the following. Given the initial forcing term $\eta_0 \in [0, 1)$, then choose

$$\eta_k = \frac{\left|\|F(x_k)\| - \|F(x_{k-1}) - F'(x_{k-1})s_{k-1}\|\right|}{\|F(x_{k-1})\|}, \quad k = 1, 2, \ldots. \tag{6.8}$$

Observe that (6.8) directly reflects the agreement between F and its local linear model at the previous Newton step. If the initial iterate x_0 is sufficiently near a solution x_*, then the sequence $\{x_k\}$ produced by Algorithm 2 and the forcing term as in (6.8), converges super-linearly towards a solution. As in the classical case of the secant method, it follows that the order

of convergence equals $(1+\sqrt{5})/2$; see, for instance, Stoer & Bulirsch (1980), page 293. The irrational number $(1+\sqrt{5})/2$ is known as the golden ratio, see Hertz-Fischler (1998).

Usually the forcing term (6.8) avoids oversolving, but it might happen that it is chosen too small. As a safeguard we restrict η_k to be no less than a certain minimal value, which depends on η_{k-1} according to

$$\eta_{k-1}^{(1+\sqrt{5})/2}. \tag{6.9}$$

Note that this safeguard should only be activated as η_{k-1} is relative large. Therefore we first check whether $\eta_{k-1}^{(1+\sqrt{5})/2}$ is larger than a certain threshold, and if so, the safeguard becomes active. As was done in Eisenstat & Walker (1996), the threshold we use is 0.1. It appeared that this threshold value worked fine in our experiments, and therefore it was not necessary to change it. To summarize:

$$\text{Modify } \eta_k \leftarrow \max\{\eta_k, \gamma\eta_{k-1}^{(1+\sqrt{5})/2}\} \text{ whenever } \gamma\eta_{k-1}^{(1+\sqrt{5})/2} > 0.1.$$

6.2.2 Choice 2

Another way to base the forcing term on residual norms is

$$\eta_k = \gamma\frac{\|F(x_k)\|^2}{\|F(x_{k-1})\|^2}, \tag{6.10}$$

with $\gamma \in [0, 1)$ a parameter. Again, we have the safeguard:

$$\text{Modify } \eta_k \leftarrow \max\{\eta_k, \gamma\eta_{k-1}^2\} \text{ whenever } \gamma\eta_{k-1}^2 > 0.1.$$

Note that for the choice of (6.10) as forcing term, the order of convergence of Inexact Newton equals 2, see Eisenstat & Walker (1996). In Kelley (2003), (6.10) is chosen as forcing term. A brief discussion on the use of this forcing term can be found in Kelley (2003).

6.2.3 Choice 3

Following Dembo & Steihaug (1983) we put

$$\eta_k = \min\left(\frac{1}{k+2}, \|F(x_k)\|\right). \tag{6.11}$$

With forcing terms as in (6.11) the Inexact Newton method converges quadratically towards a solution x_*. Note that for the first few Newton iterations, thus for small k, relatively inaccurate approximations of Newton steps s_k are allowed. Although some information on F is incorporated, it does not reflect the agreement of F and its local linear model. Note as well that the forcing term (6.11) is scaling dependent.

6.2.4 Choice 4

Superlinear convergence of the Inexact Newton method is obtained if we take the forcing term

$$\eta_k = \frac{1}{2^{k+1}}. \tag{6.12}$$

Brown & Saad (1990) used this forcing term in their solver package called NKSOL for solving the classical driven cavity problem for incompressible fluid flow. For the forcing term (6.12) holds that for the first few Newton iterations inaccurate approximations of s_k are allowed. However, no information about F is incorporated.

6.2.5 Choice 5

Another possibility is to set the forcing term to a fixed value for all nonlinear iterations. For instance, $\eta_k = 10^{-1}$ gives moderately accurate approximations of the Newton step, whereas $\eta_k = 10^{-4}$ demands more accurate approximations of s_k. For this type of forcing terms we have linear convergence towards x_*.

In Section 6.6 these five forcing terms are compared in terms of robustness and efficiency. Further, an overall best, or more overall best forcing terms are appointed.

6.3 The Globalized Inexact Newton Algorithm

In general, the initial iterate x_0, which is mostly the best guess of the solution x_* available, is not in a neighborhood of x_*. In that case, the (Inexact) Newton method diverges. Thus, it is useful to augment the (Inexact) Newton method with a sufficient decrease condition on $||F||$ such that global convergence can be obtained.

In our work we use the Inexact Newton method globalized by backtracking, or linesearch, which can be found in Eisenstat & Walker (1994). The algorithm is written down as Algorithm 3.

The sufficient decrease condition in the Globalized Inexact Newton method is formulated as

$$||F(x_k + s_k)|| \leq (1 - t(1 - \eta_k))||F(x_k)||, \tag{6.13}$$

with $0 < t < 1$. The starting point of the derivation of inequality (6.13) is the Goldstein-Armijo α-condition

$$f(x_n + d_n) \leq f(x_n) + \alpha \nabla f(x_n)^T d_n, \tag{6.14}$$

where $0 < \alpha < 1$, see Dennis & Schnabel (1983). Condition (6.14) is a sufficient condition on f to let d_n be a sufficient decrease direction, see Dennis & Schnabel (1983).

Proposition 6.1. *Let $f = 1/2\|F\|_2^2$. If inequalities (6.5) and (6.14) hold, then inequality (6.13) also holds with $t = \alpha$.*

Proof. Substituting $f = 1/2\|F\|_2^2$ into the Goldstein-Armijo α-condition (6.14) gives

$$\|F(x_k + s_k)\|_2^2 \leq \|F(x_k)\|_2^2 + 2\alpha F(x_k)^T F'(x_k)s_k. \tag{6.15}$$

The last term on the right-hand side of inequality (6.15) can be bounded as

$$\begin{aligned} F(x_k)^T F'(x_k)s_k &= F(x_k)^T \left[F'(x_k)s_k + F(x_k) - F(x_k)\right] && (6.16)\\ &= -\|F(x_k)\|_2^2 + F(x_k)^T \left[F'(x_k)s_k + F(x_k)\right] && (6.17)\\ &\leq -\|F(x_k)\|_2^2 + \left[F'(x_k)s_k + F(x_k)\right]^T \left[F'(x_k)s_k + F(x_k)\right] && (6.18)\\ &= -\|F(x_k)\|_2^2 + \|F'(x_k)s_k + F(x_k)\|_2^2. && (6.19) \end{aligned}$$

Using the inexact Newton condition (6.5), inequality (6.19) yields

$$F(x_k)^T F'(x_k)s_k \leq -(1 - \eta_k)\|F(x_k)\|_2^2. \tag{6.20}$$

To summarize, inequality (6.15) is rewritten as

$$\|F(x_k + s_k)\|_2^2 \leq (1 - 2\alpha(1 - \eta_k))\|F(x_k)\|_2^2. \tag{6.21}$$

Note that the left-hand side of inequality (6.21) is always positive. Inequality (6.21) is only valid when the right-hand side of (6.21) is positive, which is true if and only if

$$2\alpha(1 - \eta_k) \leq 1. \tag{6.22}$$

For $|x| \leq 1$ holds the inequality

$$\sqrt{1 - x} \leq 1 - x/2, \tag{6.23}$$

such that inequality (6.21) reduces to

$$\|F(x_k + s_k)\| \leq (1 - \alpha(1 - \eta_k))\|F(x_k)\|. \tag{6.24}$$

Indeed, inequality (6.13) holds with $t = \alpha$. □

In implementing Algorithm 3 we choose each initial forcing term, where for the Choices 1 and 2 we select $\eta_0 = 1/2$, and then determine an initial approximated Newton step s_k by solving the Newton equation using an iterative linear solver. In the while-loop, each λ was chosen in the following way. In the case that after two reductions by halving the Newton step does not lead to sufficient decrease, then a quadratic polynomial model of

$$\phi(\lambda) = \|F(x_k + \lambda s_k)\|_2^2, \tag{6.25}$$

is build, which is based on the three most recent values of λ. The next λ is the minimizer of (6.25), subject to the safeguard that the reduction is at least one half and at most a tenth. Comprehensive descriptions of how to build this quadratic polynomial model can be found in Kelley (2003) and van Veldhuizen et al. (2007c).

Algorithm 3: Globalized Inexact Newton

Let x_0, $\eta_{\max} \in [0, 1)$, $t \in (0, 1)$ and $0 < \lambda_{\min} < \lambda_{\max} < 1$ be given.
for $k = 1, 2, \ldots$ until 'convergence' **do**
  Find some $\eta_k \in [0, \eta_{\max}]$ and s_k that satisfy

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\|.$$

  while $\|F(x_k + s_k)\| > (1 - t(1 - \eta_k))\|F(x_k)\|$ **do**
    Choose $\lambda \in [\lambda_{\min}, \lambda_{\max}]$
    Set $s_k \leftarrow \lambda s_k$ and $\eta_k \leftarrow 1 - \lambda(1 - \eta_k)$
  end while
  Set $x_{k+1} = x_k + s_k$.
end for

6.4 Globalized Projected Newton Methods

Preservation of non-negativity of species concentrations in the solution of the species equations (2.18) is crucial to avoid blow up of the solution. Suppose that the species equations (2.18) are discretized, in space and time, such that positivity is ensured. Thus, for example, spatial discretization by means of the hybrid Finite Volume scheme (3.5) - (3.6), and in time by Euler Backward. In Chapter 3, 4 and 5 we have seen that the exact solution of the resulting fully discrete system is positive. However, solving the resulting implicit relation by means of a Globalized (Inexact) Newton Method, see Sections 6.1 and 6.3, does not guarantee positivity of the solution vector of species concentrations. Moreover, numerical experiments revealed that certain preconditioned Krylov methods return repeatedly nonlinear solutions containing negative species concentrations. Thus, in practice, even for the unconditional positive Euler Backward method, (repetitions of) negative species concentrations can be observed. For this lacking property of the (Globalized) (Inexact) Newton method we present an adaptation to the algorithm such that it preserves positivity.

The idea is to generate sequences $\{x_n\}$ in the positive orthant which converge to a solution x_* of the nonlinear problem $F(x) = 0$, where it is assumed that such a positive solution exists. The fact that $\{x_n\}$ is in the positive orthant, gives that the solution x_* contains positive entries. These so-called Projected Newton methods originate from nonlinear optimization problems with constraints, and were first proposed by Bertsekas (1982). To the author's knowledge, these kind of ideas have not been applied into the field of PDEs.

Application of Projected Newton in the field of PDEs can be done as follows. Suppose we have computed a Newton direction s_k and that the new solution vector $x_k + s_k$ contains negative entries. In Figure 6.1 this situation is

illustrated for a two-dimensional case. Then, in order to maintain positivity of these entries we project the negative entries to zero and check whether this projected solution is still in the steepest descent direction. To be more specific, we test whether the projected solution suffices the augmented sufficient decrease condition, i.e.,

$$\|F(\mathcal{P}(x_k + s_k))\| > (1 - t(1 - \eta_k))\|F(x_k)\|, \tag{6.26}$$

where $\mathcal{P}$ is the projection on the positive orthant and α a typical small parameter. The i^{th} entry of $\mathcal{P}(x)$ is given as

$$\mathcal{P}_i(x) = \begin{cases} x_i & \text{if} \quad x_i \geq 0 \\ 0 & \text{if} \quad x_i < 0 \end{cases} . \tag{6.27}$$

When condition (6.26) is not satisfied, the search direction s_k and η_k will be adjusted by means of a linesearch procedure as described in Section 6.3. The resulting algorithm, called Globalized Inexact Projected Newton, is given as Algorithm 4.

Algorithm 4: Globalized Inexact Projected Newton

Let x_0, $\eta_{\max} \in [0, 1)$, $t \in (0, 1)$ and $0 < \lambda_{\min} < \lambda_{\max} < 1$ be given.
for $k = 1, 2, \ldots$ until 'convergence' **do**
  Find some $\eta_k \in [0, \eta_{\max}]$ and s_k that satisfy
    $\|F(x_k) + F'(x_k)s_k\| \leq \eta_k\|F(x_k)\|$.
  while $\|F(\mathcal{P}(x_k + s_k))\| > (1 - t(1 - \eta_k))\|F(x_k)\|$ **do**
    Choose $\lambda \in [\lambda_{\min}, \lambda_{\max}]$
    Set $s_k \leftarrow \lambda s_k$ and $\eta_k \leftarrow 1 - \lambda(1 - \eta_k)$
    If such λ cannot be found, terminate with failure.
  end while
  Set $x_{k+1} = \mathcal{P}(x_k + s_k)$.
end for

As in the case of linesearch, or backtracking methods, we cannot prove that inequality (6.26) can always be satisfied. Neither can we derive conditions for which it surely does not hold. However, we can plead on the fact that it is a useful extension.

The unconditional positivity of Euler Backward ensures that a non-negative solution exists. If we start with a positive initial guess in a neighborhood of the positive solution, then we may expect that the algorithm converges towards this solution. However, due to the use of approximate Jacobians and/or preconditioned Krylov solvers the solution is most likely approached from a non-positive direction. By projecting the negative entries to zero, it is still likely that we remain in a neighborhood of the solution.

It is a straightforward exercise to prove that when the augmented sufficient decrease condition (6.26) is satisfied and Algorithm 4 does not break

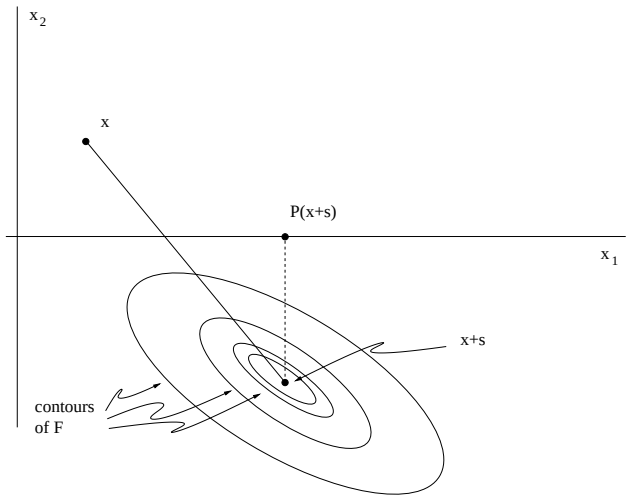


Figure 6.1: Illustration of the Projected Newton Method for a nonlinear problem of 2 variables, where $x = [x_1, x_2]^T$ and s the Newton search direction.

down, it converges to a solution. The proof is an analogue of the proof of Theorem 3.4 in Eisenstat & Walker (1994), except that the sufficient decrease condition has to be replaced by the augmented sufficient decrease condition (6.26).

6.5 Convergence Criteria

Convergence of the various Newton method's presented in this section is declared when

$$\|F(x_k)\| < \text{TOL}_{\text{rel}}\|F(x_0)\| + \text{TOL}_{\text{abs}}, \tag{6.28}$$

where TOL_{rel} and TOL_{abs} are, respectively, the relative termination tolerance and the absolute termination tolerance of the Newton process. Failure, or divergence, is declared when

- k reached the maximum number of Newton iterations $k_{\max}$,
- the iterative linear solver does not succeed in finding a suitable Newton step within the maximum number of allowed linear iterations, or, if applicable,
- the linesearch algorithm is not able to find a suitable Newton step after 10 iterations. (Taking more linesearch iterations into account would not make sense, because then the Newton update s_k would be too small to obtain convergence in the next iterations)

In the case that Newton's method diverges, then the common way to overcome divergence is to decrease the time-step size. In our code, as in many other codes, we halve the time step size and repeat Newton's process.

6.6 Numerical Experiments

In this section we report on numerical experiments with the forcing term choices outlined in Section 6.2. For Choice 1 and 2 the safeguards as presented in Section 6.2.1 and 6.2.2 are used, whereas for the other forcing term choices there are no safeguards. Numerical experiments are done for

1. the two-dimensional benchmark problem of Kleijn (2000) on spatial grids varying from 35×32 to 70×82 grid cells, and,
2. the similar chemistry problem as above on a three-dimensional computational grid consisting of $35 \times 32 \times 35$ grid cells.

Since the convergence of the Bi-CGSTAB algorithm depends heavily on the effectivity of the preconditioner, numerical experiments have been carried out with both effective and less effective preconditioners. Time integration is done by the Euler Backward scheme. In all simulations it is required that the species mass fractions remain positive.

The emphasis of the numerical experiments in this section is on the behavior of the various forcing terms. The numerical tests where the emphasis is on the Projected Newton method of Section 6.4 are presented in Chapter 8. If the computational costs for the Globalized Inexact Projected Newton method are favorable over the Globalized Inexact Newton method, then we use them, and vice versa.

In Table 6.1 the geometric averages[1] of the number of Newton iterations, function evaluations and Bi-CGSTAB iterations are listed for the forcing terms discussed in Section 6.2. These geometric averages are taken over simulations on various grid sizes and preconditioners. Conclusions drawn from Table 6.1 are discussed below. In Table 6.2 the results for the simulations with the most effective preconditioner only are summarized. These results are broken out in a separate table because in practice only the most effective preconditioner(s) are used.

The most expensive operations in these type of simulations are the construction of the Jacobian and to find the solution of the Newton equation. On the other hand, it is also important that the 'correct' solution of the nonlinear system of algebraic equations is found.

[1] The geometric mean of a data set $[a_1, a_2, \ldots, a_n]$ is given by

$$\left(\prod_{i=1}^{n} a_i\right)^{1/n} = \sqrt[n]{a_1 \cdot a_2 \cdot \cdots \cdot a_n}$$

Comparing the results presented in Tables 6.1 and 6.2 it is seen that with respect to the number of Bi-CGSTAB iterations the forcing term

$$\eta_k = \frac{\left| \|F(x_k)\| - \|F(x_{k-1}) - F'(x_{k-1})s_{k-1}\| \right|}{\|F(x_{k-1})\|}, \tag{6.29}$$

is clearly the best over the others. With respect to the number of Newton iterations it is observed that Choices 3,4 and 5 are much better than the other two. Certainly, it is expected that fixing the forcing term, as in Choice 5, is generally not an optimal strategy. In particular, the computational costs for three-dimensional simulations with forcing term 5 are high.

As remarked above, also the correctness of the nonlinear solutions is important. Various test with relatively weak preconditioners have revealed that most forcing terms cause convergence towards 'wrong' solutions. However, Choice 1 for the forcing term was the only one capable of finding the correct solution.

Choice 2 illustrates that more aggressive choices for the forcing term may decrease the number of Newton iterations. However, these 'aggressive' forcing terms can lead to oversolving, more linear iterations and less robustness. Less aggressive forcing terms, such as

$$\eta_k = \frac{\left| \|F(x_k)\| - \|F(x_{k-1}) - F'(x_{k-1})s_{k-1}\| \right|}{\|F(x_{k-1})\|}, \tag{6.30}$$

might need less linear iterations, improve robustness and lead to an increasing number of Newton iterations. However, experiments with weaker preconditioners have shown that less aggressive forcing terms return correct solutions, where aggressive forcing terms lack that property.

Finally, Choice 4 for the forcing term, e.g.,

$$\eta_k = \frac{1}{2^{k+1}}, \tag{6.31}$$

gives, unexpectedly, also very good results. It has a drawback that it is not based on the agreement of $F(x_k)$ and its local linear model. In the case that the number of Newton iterations increases to obtain convergence, the required accuracy of the solution of Newton's equation might become too high to obtain fast Bi-CGSTAB convergence. It has to be remarked that although good results are found for the succesful runs with forcing term (6.12), there were also fatal failures due to the above described drawback.

The forcing term of choice would be forcing term (6.8), because it returned always a correct solution and the least number of linear iterations is needed to obtain the time-accurate solution. Since the computational costs are mainly determined by the costs of computing, or approaching, the solution of the Newton equation in each Newton iteration, also forcing term

(6.10) is a good choice. Combined with the most powerful preconditioners it gave good results, whereas combining it with less effective preconditioners occasionally no solution is found at all. Thus, to summarize, the two 'best' forcing terms are Choice 1 (6.8) and Choice 2 (6.10). Although their numerical experiments dealt with other physical problems, and problems out of the PDE world, Eisenstat & Walker (1996) concluded as well that these forcing terms are overall the best.

Table 6.1: Summary of the number of BI-CGSTAB and Newton iterations and number of function evaluations over all simulations with various preconditioners.

Choice	η_k	F	Newton	Bi-CGSTAB
1	$\eta_k = \frac{\lvert\lvert\lvert\lvert F(x_k)\rvert\rvert - \lvert\lvert F(x_{k-1}) - F'(x_{k-1})s_{k-1}\rvert\rvert\rvert}{\lvert\lvert F(x_{k-1})\rvert\rvert}$	352.4	201.4	3120.3
2	$\eta_k = \gamma \lvert\lvert F(x_k)\rvert\rvert^2 / \lvert\lvert F(x_{k-1})\rvert\rvert^2$, $\gamma = 0.5$	320.8	176.8	3480.2
3	$\eta_k = \min(1/k+2, \lvert\lvert F(x_k)\rvert\rvert)$	303.4	161.5	6953.2
4	$\eta_k = 1/2^{k+1}$	264.3	147.1	3809.6
5	$\eta_k = 10^{-1}$	311.0	172.2	3721.9
5	$\eta_k = 10^{-4}$	270.5	127.0	6395.7

Table 6.2: Summary of the number of BI-CGSTAB and Newton iterations and number of function evaluations over the simulations with effective preconditioner only.

Choice	η_k	F	Newton	Bi-CGSTAB
1	$\eta_k = \frac{\lvert\lvert\lvert\lvert F(x_k)\rvert\rvert - \lvert\lvert F(x_{k-1}) - F'(x_{k-1})s_{k-1}\rvert\rvert\rvert}{\lvert\lvert F(x_{k-1})\rvert\rvert}$	339.0	193.0	1237.1
2	$\eta_k = \gamma \lvert\lvert F(x_k)\rvert\rvert^2 / \lvert\lvert F(x_{k-1})\rvert\rvert^2$, $\gamma = 0.5$	327.8	175.8	1428.6
3	$\eta_k = \min\left(\frac{1}{k+2}, \lvert\lvert F(x_k)\rvert\rvert\right)$	258.8	137.1	2276.4
4	$\eta_k = 1/2^{k+1}$	259.1	143.2	1532.0
5	$\eta_k = 10^{-1}$	333.5	176.5	1562.6
5	$\eta_k = 10^{-4}$	231.8	122.5	2880.0

CHAPTER 7

Preconditioned Krylov Methods

In Chapter 6 Inexact Newton methods, and extensions of such methods, have been discussed. Within these nonlinear solvers the Newton equation (6.3) is assumed to be solved inexaclty, i.e., the Newton step is approximated in some way. In Chapter 6 it has not been specified how such an approximation is obtained.

The Jacobian matrix in the Newton equation (6.3) is large and sparse. For such linear systems direct solution methods, such as the LU factorization, can be impractical, because the lower triangular matrix L and the upper triangular matrix U can be dense. Nowadays, with the computational power available for general two-dimensional problems the LU factorization of a sparse matrix is feasable. However, for three-dimensional problems, iterative solution methods are in general much more efficient.

For the problems considered in this study the number of unknowns depends on the number of spatial dimensions, the number of grid points in each spatial direction and the number of species in the gas mixture. In that case, considerable improvements are found in the two-dimensional case on the total workload to find the solution of the Newton equation when using iterative solution methods over direct solution methods. The computational effort to factorize the Jacobian matrix in the two-dimensional case is mainly due to the fill-in in the zeros between the most outer subdiagonal (and superdiagonal) and the main diagonal. The distance of these diagonals depends besides the number of mesh points also on the number of species.

For these type of computations the need for a computationally efficient linear solver is essential. Suitable candidates to solve such large and sparse linear systems are Krylov Subspace methods, see for instance Saad (2003). In Section 7.1 a comparison is made for the two major Krylov methods for general linear systems. For symmetric positive definite linear systems the

best iterative method is the conjugate gradient method, see for instance Saad (2003). Due to the partial derivatives of the reaction terms present in the Jacobian matrix one has to deal with unsymmetric linear systems.

In Section 7.2 the condition of the Newton equation and the consequences for the convergence behavior of the Krylov methods are studied. Section 7.3 is devoted to the different orderings of the unknowns and the consequence for the nonzero structures of the Jacobian matrix. To acclerate the convergence speed of Krylov Subspace methods effective preconditioners are crucial. In Section 7.4 various preconditioners are presented.

The construction of an effective preconditioner is a combination of art and science. In this case the science part represents two important observations. The ordering of the unknowns should be in such a way that solving equations involving the preconditioner can be done efficiently. Secondly, the computational algorithms and their implementation, should be optimal. This chapter is concluded by some numerical results.

7.1 Krylov Solver: Bi-CGSTAB versus GMRES

Generally speaking, for non-symmetric linear systems there are two choices for an iterative Krylov solver, i.e.,

- the class of GMRES-type methods and all its variations, see, for instance, Saad (2003), and,
- the class of Bi-CGSTAB methods and its variations, see, van der Vorst (1992).

For a description and discussion of these methods we refer to a standard text like that of Saad (2003). Most recently, Sonneveld & van Gijzen (2007) introduced the family of IDR(s) methods, which has like Bi-CGSTAB modest memory requirements. In their work, Sonneveld & van Gijzen (2007) observed that the IDR(s) method performs as well as or better than Bi-CGSTAB. However, in this study this family of methods has not been considered.

In Faber & Manteuffel (1984) it has been shown that it is impossible to obtain a Krylov method for general matrices which is optimal and has short recurrences. In this case, optimality is related to the minimization of the linear residual in a certain norm.

A GMRES-like iterative method has long recurrences and minimizes the linear residual in a certain norm. Per GMRES iteration one matrix-vector product has to be computed, but on the other hand a considerable number of vectors have to be in memory. To be more precise, if k iterations are needed to find an approximation of the solution, then k vectors have to be in memory. Further, when the number of iterations increases, also the work on vector updates grows. Usually, a restarted version of GMRES is used

to reduce the amount of vectors in memory. When restarting the GMRES algorithm after $\tilde{k}$ iterations, then at most $\tilde{k}$ vectors have to be in memory. A well known difficulty with restarted GMRES is that it can stagnate when the matrix is not positive definite. Further details are found in Saad (2003).

Bi-CGSTAB-type methods are not minimizing the residual, but have short recurrences. Each Bi-CGSTAB iteration needs two matrix-vector products, and needs seven vectors in memory.

For the applications considered in this study it is important to find the nonlinear solutions arriving from the implicit time integration efficiently. Finding those nonlinear solutions are done by means of an Inexact Newton method, in which the solution of the Newton equation is approximated up to a certain accuracy. The speed of convergence for both the GMRES and Bi-CGSTAB Krylov solvers is mainly determined by the strength of a preconditioner. Since neither of the two classes gives better convergence results for the type of problems considered the Bi-CGSTAB methods are selected. From the point of view of memory usage the Bi-CGSTAB method is favorable over the GMRES method as well, i.e., the number of vectors in memory are seven. Secondly, in this study the matrices are structured, and thus, the matrix-vector product is in that case relatively cheap. This also supports our choice of Bi-CGSTAB as the iterative linear solver in our codes.

7.2 Condition of the Newton Equation

The forward and backward reaction rate constants in the reaction terms of the species equations (2.18) differ orders of magnitude from each other, and from the advection and diffusion terms. Thus, the partial derivatives of the advection, diffusion and reaction terms differ orders of magnitude from each other as well. These partial derivatives are, multiplied by the time step size τ, the entries of the Jacobian matrix, such that the individual entries within the Jacobian matrix differ orders of magnitude from each other. Consequently, a large spread in the eigenvalue distribution of the Jacobian might occur.

Definition 7.1. *The condition number for matrix inversion with respect to a matrix norm $\|\cdot\|$ of a square matrix A is defined by*

$$\kappa(A) = \|A\| \cdot \|A^{-1}\| \tag{7.1}$$

if A is non-singular; and $\kappa(A) = +\infty$ if A is singular.

Based upon the reasoning above and the fact that the spectral radius $\rho(A) \leq \|A\|$, the condition number of the Jacobian matrix can be very large too. When integrating the advection, diffusion and reaction terms implicitly,

the Jacobian matrix J_F is of the following general form

$$J_F = c\mathbf{I} - \tau(\mathbf{A} + \mathbf{D} + \mathbf{R}), \tag{7.2}$$

where

- $c \in \mathbb{R}$ is a scalar,
- $\mathbf{A}$ is the discretized advection operator,
- $\mathbf{D}$ is the discretized diffusion operator,
- $\mathbf{R}$ is the discretized and linearized reaction operator, and,
- τ the time step size.

From relation (7.2) it follows that the magnitude of the entries in the Jacobian matrix also depends on the time step size τ. For small time step sizes the condition number is orders of magnitude smaller than for relatively larger time step sizes.

For the two-dimensional benchmark problem of Kleijn (2000), see Chapter 8 as well, the estimated condition number for each linear system to be solved within the time accurate simulations has been estimated. The LAPACK package is used to compute these estimations, see Andersen et al. (1995). The Euler Backward time integration method has been used, such that the time step size remains relatively large. Further, a projected Newton method has been used to maintain positive approximated solutions. If more than one Newton equation per time step has to be solved, then the average of these condition numbers is shown. As a function of real time in seconds, a typical order of magnitude of the condition number of the Jacobian is shown in Figure 7.1.

The simulation starts with a small time step size, which drops the condition number considerably. Between $t = 10^{-5}$ s and $t = 10^{-4}$ s the condition number increases as fast as the time step size increases. In the transition period between $t = 10^{-4}$ s and $t = 1$ s, the increasing condition number is a combination of increasing the time step size and the partial derivatives of the chemistry terms which are increasing in order of magnitude.

With an average inflow velocity of 0.1 m/s, and a distance of 0.1 m to be crossed by the gas mixture to enter the reaction zone, it takes about less than 1 s for the gas mixture to start reacting. In Figure 7.1 it can be seen that around $t = 1$ s the condition number is still growing up to $O(10^{11})$.

It is well known that the convergence speed of a Krylov method depends on the condition number. For the most prominent Krylov method, the Conjugate Gradient method, see for instance Saad (2003), it has been derived that

$$\|x - x_k\|_A \leq 2\left(\frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1}\right)^k \|x - x_0\|_A, \tag{7.3}$$

where $\|\cdot\|_A$ is the norm

$$\|y\|_A = \sqrt{y^T A y}, \tag{7.4}$$

for a symmetric positive definite matrix A. Inequality (7.3) implies that for matrices A with a large condition number, the convergence towards the solution is slow.

For the GMRES method similar results are derived. If A is a symmetric positive definite matrix, then

$$\|r_k\| \leq \left(\frac{\kappa^2(A) - 1}{\kappa^2(A)}\right)^{k/2} \|r_0\|, \tag{7.5}$$

for a sequence x_k generated by GMRES, and $r_k = b - Ax_k$. For positive definite matrices A it can be derived that

$$\|r_k\| \leq \left(1 - \frac{\lambda_{\min}(A^T + A)}{\lambda_{\max}(A^T + A}\right)^{k/2} \|r_0\|. \tag{7.6}$$

For general matrices A, which are assumed to be diagonalizable, i.e., $A = X\Lambda X^{-1}$, where Λ is the diagonal matrix of eigenvalues, it holds that

$$\|r_k\| \leq \kappa_2(X)\left(\min_{p\in\mathbb{P}_k} \max_{i=1,\ldots,m} |p(\lambda_i)|\right)\|r_0\|, \tag{7.7}$$

where $\mathbb{P}_k$ is the set of polynomials p of degree k with $p(0) = 1$, m is the dimension of A and λ_i the i-th eigenvalue of A. Roughly speaking, expressions (7.5) - (7.7) say that fast convergence is obtained when the eigenvalues of A are clustered away from the origin. However, due to the stiff chemistry terms we always have a few eigenvalues that are very large in absolute value, and thus slow GMRES convergence will be obtained.

For the class of Bi-CGSTAB methods similar relations will, most likely, hold, but no explicit expressions are available. To conclude, the linear systems in this study require effective preconditioners to decrease the condition number of the preconditioned system, and thus the number of linear iterations needed to converge towards the solution. This issue will be covered in Section 7.4.

7.3 Ordering of Unknowns

Essential for the performance of the direct linear solvers and iterative linear solver combined with an incomplete factorization type preconditioners is the ordering of unknowns. For the reacting flow problems studied here, the number of unknowns is equal to $N \cdot n$, where N is the number of species in the gas-mixture and n the total number of gridpoints resulting from either a two-dimensional or three-dimensional spatial discretization.

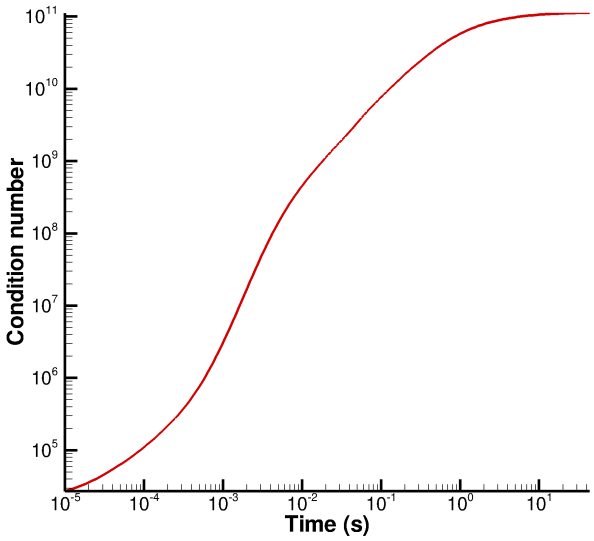


Figure 7.1: Condition-number of the Jacobian as function of time (in seconds)

Since the computational domain is usually k-dimensional rectangular parallelepiped, with $k = 2, 3$, and the computational grid is structured, the ordering of the n unknowns for a gas mixture of one species is straighforward. For a two-dimensional rectangularly shaped computational grid, the Jacobian matrix containing the partial derivatives of the discretized advection, diffusion and reaction operators has a nonzero pattern as presented in Figure 7.2. This ordering is called a *natural ordering*.

Following the description above, one gets for more than one species a repetition of the nonzero pattern illustrated in Figure 7.2 along the diagonal blocks. The partial derivatives of the reaction terms in equation (2.18) with respect to the remaining (N - 1) species appear as extra sub- or super-diagonals. For a two-dimensional computational mesh with n grid points the bandwidth of the Jacobian matrix is then $(N-1)n$. The nonzero pattern of the Jacobian matrix for the *natural ordering* is illustrated in Figure 7.3.

The bandwidth can be decreased considerably by ordering the unknown species mass fractions per grid point. For a two-dimensional computational grid with nr grid points in radial direction and nz grid points in axial direction, the bandwidth of the Jabobian matrix equals $nr \cdot N$. Remark that in this case we label the unknowns first in radial direction and thereafter in axial direction. The corresponding nonzero pattern of the Jacobian matrix is shown in Figure 7.4.

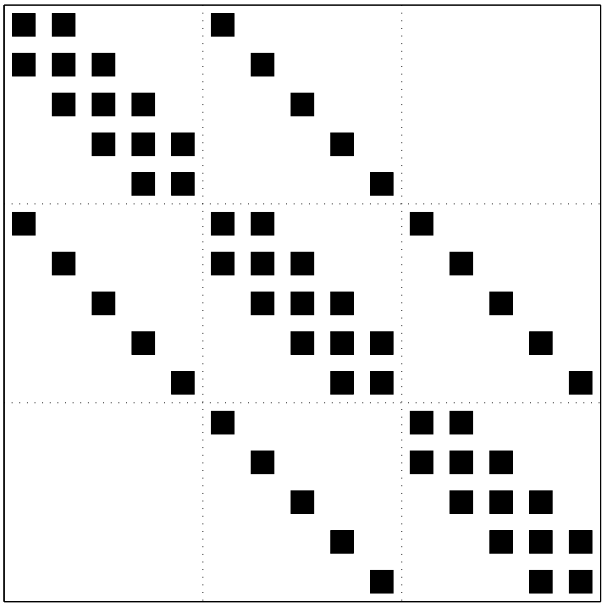


Figure 7.2: Nonzero pattern of the Jacobian matrix for a 5×3 grid

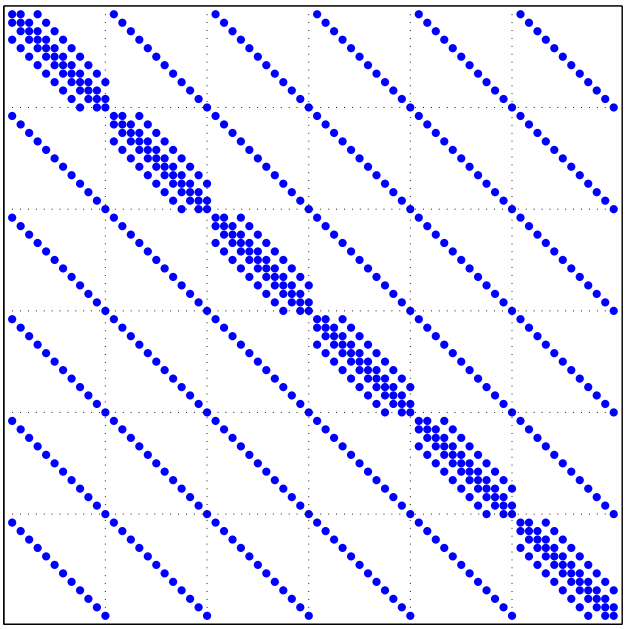


Figure 7.3: Nonzero pattern of the Jacobian-matrix for $s = 6$ and the unknowns ordered in a natural way.

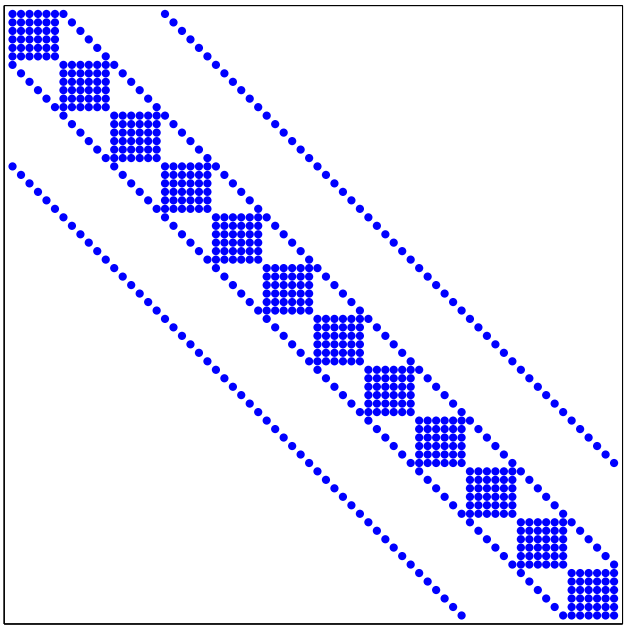


Figure 7.4: Nonzero pattern of the Jacobian-matrix for $s = 6$ for the per grid point ordering.

In van Veldhuizen et al. (2008b) an LU factorization of the Jacobian matrix is used, instead of an iterative linear solver, to find the solution of the Newton equation. Obviously, due to the amount of fill-in in the factorization of the sparse Jacobian matrix, the natural ordering is ruled out. However, the considerably much smaller bandwidth of the Jacobian matrix for the per grid point ordering reduces the amount of fill-in. Hence, the LU factorization is computationally feasable. Numerical results are found in Chapter 5 of this thesis and in van Veldhuizen et al. (2008b).

7.4 Preconditioners

As remarked in Section 7.1 the computational efficiency of the Krylov solver is for a great deal determined by the effectivity of the preconditioner. In this section we present two incomplete factorization based preconditioners, and two block diagonal based preconditioners.

7.4.1 Incomplete LU Factorization Preconditioners

For regularly structured computational grids, like present in this study, the Jacobian matrix is regularly structured as well. This property can be exploited to formulate incomplete LU factorization preconditioners in a simple way. In standard texts like for instance Saad (2003) and Barret et al. (1994), this has been illustrated for the inhomogenuous steady state advection-diffusion equation on a rectanglar domain. Spatial discretization is done by central Finite Volumes, where of course, it is assumed that this discretization is stable. The corresponding discretization matrix has a nonzero structure as in Figure 7.2.

These algorithms are easily extended for the species equations (2.18) with more than one species. The extra nonzero sub- and superdiagonals should be treated in the same way as the off-diagonals for the advection-diffusion case described above. For both ordering discussed in Section 7.3 with corresponding nonzero structures as in Figures 7.3 and 7.4, this extension is straightforward.

Basic iterative methods like Jacobi or Gauss-Seidel converge more quickly if the diagonal entry is relative large to the off-diagonals in its row or column. Techniques like block iterative methods can benefit if the entries in the diagonal blocks are large. For preconditioning techniques it is intuitively evident that large diagonals should be beneficial, see Duff & Koster (1999). Comparing both orderings, it is seen that for the per grid point ordering, the partial derivatives are clustered in the diagonal blocks, see Figure 7.4. Numerical experiments reveal that this ordering enhances the convergence speed of the iterative linear solver. The results are presented in Section 7.5.

Block Incomplete Factorization

For both orderings there is a natural block structure in the Jacobian matrix. For the natural ordering, see Figure 7.3, a block structure with blocks of dimension n, with n the number of grid points, over the species is present. Building an incomplete factorization on block level for this nonzero structure converts the diagonal blocks in the strictly lower triangular part of the nonzero structure into dense blocks. For fine two-dimensional and three-dimensional meshes, and a large number of reactive species in the gas mixture, it is impossible to store the strictly lower triangular part in computer memory.

Building an incomplete LU factorization on block level for the per grid point ordering appears to be efficient. In this section it will be illustrated for a reactangular computational grid on which the species equations (2.18), in cylindrical coordinates, are discretized by means of the hybrid Finite Volume scheme of Chapter 3.

Denote nr as the number of grid points in radial direction and nz the number of grid points in axial direction, such that the total number of grid points is $n = nr \cdot nz$. Ordering the unknown species mass fractions per grid point generates a Jacobian matrix with a nonzero structure consisting of blocks with a dimension equal to the number of species N. The blocks on the diagonal A_{ii}, $i = 1, \ldots, n$ are not sparse. The other nonzero blocks $A_{i-1,i}, A_{i,i-1}, A_{i-nr,i}$ and $A_{i,i-nr}$ are diagonal (sub)matrices, see Figure 7.4.

The Jacobian matrix can be split into three matrices, namely,

1. a matrix D, containing all blocks A_{ii} on the main diagonal,
2. the strictly upper part U, containing the blocks $A_{i-1,i}$ and $A_{i-nr,i}$, and,
3. the strictly lower part L, containing the blocks $A_{i,i-1}$ and $A_{i,i-nr}$.

The block incomplete LU factorization preconditioner is then written as

$$M = (D + L)D^{-1}(D + U), \tag{7.8}$$

where D is the block diagonal matrix containing the block pivots generated. Algorithm 5 described how this preconditioner is constructed. Since the upper and lower triangle parts of the matrix remain unchanged, only storage space for D is needed.

In the preconditioned Bi-CGSTAB algorithm the so-called preconditioned linear systems

$$Mx = b, \tag{7.9}$$

with M defined as in (7.8), have to be solved. In the computer code the following equivalent formulation has been implemented:

1. Solve z from $(D + L)z = b$, and,

Algorithm 5: Block ILU

Put $D_{ii} = A_{ii}$ for all $i = 1, \ldots, n$
for $i = 2, \ldots, n$ **do**
  if mod $(i, nr) \neq 0$ **then**
    $D_{i+1,i+1} = D_{i+1,i+1} - A_{i+1,i} D_{ii}^{-1} A_{i,i+1}$
  end if
  if $i + nr \leq N \cdot n$ **then**
    $D_{i+nr,i+nr} = D_{i+nr,i+nr} - A_{i+nr,i} D_{ii}^{-1} A_{i,i+nr}$
  end if
end for

2. Solve x from $(I + D^{-1}U)x = z$.

Solving $Mx = b$ using this formulation is outlined in Algorithm 6.

Algorithm 6: Preconditioner solve of a system $Mx = b$, with $M = (D + L)D^{-1}(D + U)$

for $i = 1, \ldots, n$ **do**
  Solve $D_{ii} z_i = b_i - \sum_{j<i} L_{ij} z_j$
end for
for $i = n, \ldots, 1$ **do**
  Solve $D_{ii} y = \sum_{j>i} U_{ij} x_j$
  Put $x_i = z_i - y$
end for

With respect to solving systems

$$D_{ii} y = \sum_{j>i} U_{ij} x_j, \tag{7.10}$$

and

$$D_{ii} z_i = b_i - \sum_{j<i} L_{ij} z_j, \tag{7.11}$$

as formulated in Algorithm 6, is done by building an LU factorization of D_{ii}. Since the dimension of D_{ii} equals the number of species, and is small with respect to the number of grid points, this is a cheap operation.

For the right multiplication of D_{ii}^{-1} and the diagonal matrix $A_{i,i+1}$, as found in Algorithm 5, we proceed as follows. The inverse of D_{ii} is computed exactly using the Gauss-Jordan decomposition, see for instance Strang (2003). The resulting inverse matrix is then multiplied by the diagonal matrix $A_{i,i+1}$. To solve the systems

$$D_{ii} y = \sum_{j>i} U_{ij} x_j, \tag{7.12}$$

and

$$D_{ii}z_i = b_i - \sum_{j<i} L_{ij}z_j, \tag{7.13}$$

as formulated in Algorithm 6, the Gauss-Jordan factorization of D_{ii} can be re-used.

Another approach to compute $D_{ii}^{-1}A_{i,i+1}$ is to compute the LU factorization of D_{ii} and subsequently solve N linear systems. In terms of floating point operations, or shorter flops, this approach costs $2/3N^3 + N \cdot N^2$ flops. The approach using the Gauss-Jordan decomposition needs N^3 flops to compute the exact inverse, and N^2 for the multiplication with the diagonal matrix. Based on the amount of flops we use the first approach, i.e., the Gaus-Jordan decomposition.

7.4.2 Block Diagonal Preconditioners

For the per grid point ordering of unknowns the nonzero pattern of the Jacobian matrix is as in Figure 7.4. As an approximation of the Jacobian matrix one could use the block diagonal matrix, which is easily obtained by omitting the off block diagonal elements. The resultant approximate Jacobian is easily invertable, because it consists of small, easily factorizable subsystems on the diagonal blocks.

Lumping

For the per grid point ordering another approximation of the Jacobian can be obtained, whose nonzero structure resembles the nonzero structure of the block diagonal preconditioner. This can be achieved by 'lumping' the Jacobian matrix. Note that it important to lump the same species. Thus, in the case of Figure 7.4 the four off-block diagonals are added to the main diagonal.

From a mathematical point of view this is a valid approximation of the Jacobian matrix as well. The off-block diagonal elements represent the contributions of the discretized advection-diffusion operator of the neighboring points of a certain grid point C in the computational grid, see Figure 3.1. Since these approximations are mostly second order accurate, see Chapter 3, such a contribution of a neighbor point of grid point C equal to the value of the solution in grid point C up to a first order truncation error. Thus, the contribution of this neighbor can be replaced by this first order approximation. Hence, a second order accurate approximation of the Jacobian matrix has been constructed.

This mass lumping approach can also be applied to the Jacobian matrix with the unknowns ordered in the natural ordering, see Figures 7.3 and 7.5. In that case, the diagonals marked by circles in Figure 7.5 should be added to the main diagonal. When constructing the resulting lumped matrix, wich

us a diagonal matrix (N-1) superdiagonals and (N-1) subdiagonals, the LU factors have the same nonzero pattern as the lumped matrix. However, the implementation for this ordering is more difficult than for the per grid point ordering.

7.4.3 Comparison of Costs: Flops

To indicate the amount of work for one of the above preconditioners, we present for each of them the number of floating point operations (flops) needed to build the preconditioner P, and the number of flops to solve $Px = b$. Note that per Newton iteration the preconditioner is built once, and the $Px = b$ is solved twice in each Bi-CGSTAB iteration. From Table 7.1 it can be concluded that the incomplete LU-factorization, the lumped Jacobian and the block diagonal are, in terms of flops, the cheapest to build, i.e., the number of flops scales linearly and n and cubically in N. The most expensive preconditioner to build is the blocked version of ILU. Thus, the extra fill-in in this preconditioner expresses itself in, of course, extra computational costs.

Table 7.1: Number of floating point operations to build the preconditioner P and to solve $Px = b$. The total number of grid points is denoted as n and N denotes the number of species.

	Building P	Solving $Px = b$
ILU(0)	$8nN^3$	$2n(N^2 + 4N)$
Lumped Jacobian	$2/3N^3n$	$2N^2n$
Block ILU	$2n(N^3 + 3N^2)$	$6N^2n$
Block diagonal	$2/3N^3n$	$2N^2n$

The extra fill-in for block ILU results also in extra computational costs for solving $Px = b$. The cheapest preconditioned systems to solve, in terms of flops, are those belonging to the lumped Jacobian and the block diagonal.

7.5 Numerical Results

In Sections 7.3 and 7.4 the influence of the ordering of unknowns came up for discussion. In Section 7.4.1, in particular, it is stated that linear systems with relatively large diagonal elements, compared the off-diagonal elements, converge quicker towards the solution. Further, it is stated that intuitively also holds for the preconditioners.

In Table 7.2 the results of simulations with the incomplete LU factorization, which is discussed in Section 7.4.1, as preconditioner. As test problem the benchmark problem of Kleijn (2000) is used, of which further details are

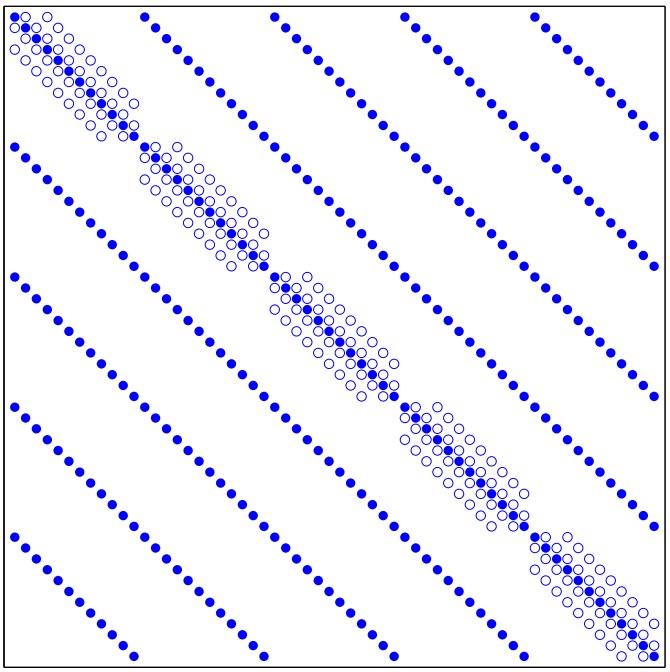


Figure 7.5: Nonzero pattern of the lumped approximations to the Jacobian matrix, where the unknowns are ordered according to the natural ordering. The super- and sub-diagonals marked by circles should be added to the main diagonal.

found in Chapter 8. As time integration Euler Backward is used and the nonlinear systems are solved by the Inexact Newton method discussed in Section 6.3.

It is quite clear that the various orderings have significant effect on the total computational effort needed to perform the simulation. Mainly, the computational costs are due to the linear solver. When less Bi-CGSTAB iterations are needed to find the solution, up to a certain accuracy, the total costs are expected to be lower. From Table 7.2 can be seen that the total number of linear iterations for the per grid point ordering is significantly less than for the natural ordering. Secondly, the approximated linear solutions obtained in the per grid point ordering are more accurate than those obtained with the natural ordering. This results in a lower number of Newton iterations.

	per grid point			natural		
Mesh size	35×32	35×47	70×82	35×32	35×47	70×82
F	203	407	675	205	476	811
F′	108	213	353	114	252	461
Newton	108	213	353	114	252	461
linesearch	9	61	124	8	76	138
lin iter	848	2111	7171	1131	2455	7942

Table 7.2: Integration statistics for GIN with ILU(0) as preconditioner for two orderings of the unknowns.

In Chapter 8 a comprehensive overview is given on the performance of the various preconditioners discussed in Section 7.4. Besides the performance of the preconditioners individually, the performance of these preconditions combined with the various Newton methods studied in Chapter 6 is discussed. To decrease the amount of total computational costs it is important to 'tune' the various parts of the solver.

CHAPTER 8

Numerical Results: Chemical Vapor Deposition

Numerical simulations presented in Chapter 5 and in this chapter are done for the Chemical Vapor Deposition process of silicon from silane. The gas phase and surface chemistry are modeled according to one of the two reaction models presented in this chapter. In Section 8.1.1 a chemistry model consisting of 7 species and 5 gas phase reactions without surface chemistry is presented, which can easily be used as a small test problem. Section 8.1.2 is devoted to the description of the classical model of the same process as published by Coltrin et al. (1989), which includes 17 gas species, 26 gas phase reactions and 14 surface reactions.

Further, two reactor configurations are considered. The first reactor configuration is the one used in the benchmark solution of Kleijn (2000). Since this benchmark problem is axisymmetric, the computational domain reduces to two spatial dimensions. Further details are discussed in Section 8.2.1. The second reactor configuration results in a three-dimensional computational domain. A detailed description can be found in Section 8.2.2.

As already mentioned in Section 2.6, silane and the formed reactive intermediates are highly diluted in the inert carrier gas helium. Since the velocity-, temperature-, density- and pressure fields are not influenced by the transient chemistry, it is justified to use the steady state flow field. For the two-dimensional simulations the steady state flow problem is solved using the computer code CVDMODEL of Kleijn, see for instance Kleijn (2000). The three-dimensional steady state flow is computed by the proprietary CFD software package CVD-X, which is developed at TNO Science and Industry, see TNO Science and Industry (2007).

8.1 Chemistry Models

In the first test case, published in van Veldhuizen et al. (2006c) and van Veldhuizen et al. (2006d), time accurate transient simulations are presented of the Chemical Vapor Deposition process of silicon from silane according to a reaction model with 7 species and 5 gas phase reactions. This reaction model does not account for surface chemistry. The chemistry model is discussed in Section 8.1.1. To facilitate easy reproduction, the diffusion coefficients and molecular weights for all species are presented as well. The computational domain is two-dimensional, because of axisymmetry (i.e., assuming that the tangential derivatives of all variables to be zero).

The second test case in this study, is the same Chemical Vapor Deposition process of silicon from silane, now modeled according to the classical 17 species and 26 gas phase and 14 surface reactions chemistry model as published by Coltrin et al. (1989). Kleijn (2000) published a two-dimensional steady state benchmark solution of this process for an axisymmetric stagnation flow Chemical Vapor Deposition reactor. Time accurate transient results for this benchmark problem are published in van Veldhuizen et al. (2007a) and van Veldhuizen et al. (2008b). The reaction model, diffusion coefficients and molecular weights are given in Section 8.1.2.

8.1.1 Chemistry model I: 7 species and 5 gas phase reactions

The 5 gas phase reactions are listed in Tables 8.1 and 8.2, in which all reactive gas phase species, except for the carrier gas helium He, can be found. Note that for this model only 6 nonlinearly and stiffly coupled species equations (2.18) have to be solved, because the mass fraction of He can be computed via the property that the mass fractions of all species in the gas mixture add up to one, see expression (2.4). The reaction terms in the species equations (2.18) are constructed as in expressions (2.20) and (2.21). The fit parameters A_k, β_k and E_k needed in the modified Arrhenius expression (2.21) are presented in Table 8.1. The backward rates are computed self-consistently from

$$k^g_{\text{backward}}(T) = \frac{k^g_{\text{forward}}(T)}{K^g(T)} \left(\frac{RT}{P^0}\right)^{\sum_{i=1}^{N} \nu_{ik}}, \tag{8.1}$$

with $K^g(T)$ the reaction equilibrium constants, see Section 2.4.4. To facilitate easy reproduction of the solutions presented in this thesis, the reaction equilibrium constants are fitted to a modified Arrhenius expression

$$K^g(T) = A_{k,\text{eq}} T^{\beta_{k,\text{eq}}} e^{\frac{-E_{k,\text{eq}}}{RT}}. \tag{8.2}$$

In expression (8.2) A_{eq}, β_{eq} and E_{eq} are fit parameters, which are given Table 8.2.

Table 8.1: Gas phase reaction mechanism and fit parameters of the forward reaction rate constant $k^g_{k,\text{forward}}$, see expression (2.21), for the 6 species and 5 reactions model described in Section 8.1.1. The parameter β_k is dimensionless, while E_k has unit kJ $\cdot$ mol^{-1} and the unit of A_k depends on the order of the reaction, but is expressed in units mole, m^3 and s.

Reaction	A_k	β_k	E_k
$SiH_4 \rightleftarrows SiH_2 + H_2$	1.09×10^{25}	-3.37	256
$Si_2H_6 \rightleftarrows SiH_4 + SiH_2$	3.24×10^{29}	-4.24	243
$Si_2H_6 \rightleftarrows H_2SiSiH_2 + H_2$	7.94×10^{15}	0	236
$SiH_2 + Si_2H_6 \rightleftarrows Si_3H_8$	1.81×10^{8}	0	0
$2SiH_2 \rightleftarrows H_2SiSiH_2$	1.81×10^{8}	0	0

Table 8.2: Gas phase reaction mechanism and fit parameters of the reaction equilibrium constants K^g, see expression (8.2), for the 6 species and 5 reactions model described in Section 8.1.1. The parameter β_{eq} is dimensionless, while E_{eq} has unit kJ $\cdot$ mol^{-1} and the units of A_{eq} depends on the order of the reaction, but is expressed in units mole, m^3 and s.

Reaction	$A_{k,eq}$	$\beta_{k,eq}$	$E_{k,eq}$
$SiH_4 \rightleftarrows SiH_2 + H_2$	6.85×10^{5}	0.48	235
$Si_2H_6 \rightleftarrows SiH_4 + SiH_2$	1.96×10^{12}	-1.68	229
$Si_2H_6 \rightleftarrows H_2SiSiH_2 + H_2$	3.70×10^{7}	0	187
$SiH_2 + Si_2H_6 \rightleftarrows Si_3H_8$	1.36×10^{-12}	1.64	-233
$2SiH_2 \rightleftarrows H_2SiSiH_2$	2.00×10^{-7}	0	-272

In order to be self contained, the expressions for the effective multicomponent diffusion coefficients $\mathbb{D}'_i$ are presented subsequently. For species i the effective multicomponent diffusion coefficient $\mathbb{D}'_i$ is fitted as a function of local temperature according to

$$\mathbb{D}'_i = \mathbb{D}'_{i,300} \left(\frac{T}{300} \right)^{1.7}, \tag{8.3}$$

with the diffusion coefficient at $T = 300$ K, $\mathbb{D}'_{i,300}$ as in Table 8.3. The molecular weights m_i of reactive species i in the gas mixture are listed in Table 8.3 as well.

Table 8.3: Fitting coefficients for the effective multicomponent diffusion coefficients and molecular weights of the various species in the gas mixture. The unit of the fitting constant $\mathbb{D}'_{i,300}$ is $m^2 \cdot s^{-1}$, and molecular weights are expressed in $kg \cdot mol^{-1}$.

Species	$\mathbb{D}'_{i,300}$	m
SiH_4	$4.77 \cdot 10^{-6}$	0.032118
SiH_2	$5.38 \cdot 10^{-6}$	0.030102
H_2SiSiH_2	$3.94 \cdot 10^{-6}$	0.060204
Si_2H_6	$3.72 \cdot 10^{-6}$	0.062219
Si_3H_8	$3.05 \cdot 10^{-6}$	0.092321
H_2	$8.02 \cdot 10^{-6}$	0.002016

8.1.2 Chemistry model II: 17 species and 26 gas phase reactions

In this reaction model after Coltrin et al. (1989) the decomposition of silane into silylene and hydrogen, initiates a chain of 25 homogeneous gas phase reactions leading to the formation and deformation of 14 silicon containing gas phase species. Again, the reaction terms in the species expressions (2.18) are constructed as in expressions (2.20) and (2.21). The backward rates are computed selfconsistently from expressions (8.1) and (8.2). The 26 reactions and the fit parameters A_k, β_k and E_k needed in the modified Arrhenius expression (2.21) for the forwad reaction rate constants are listed in Table 8.4. The fit parameters $A_{k,\mathrm{eq}}$, $\beta_{k,\mathrm{eq}}$ and $E_{k,\mathrm{eq}}$ needed in the modified Arrhenius expression (8.2) for the reaction equilibrium constants, are listed Table 8.5.

Each of the silicon containing species in the gas mixture may diffuse towards and react at the susceptor. In this model it is assumed that film growth is due to irreversible, unimolecular decompostion reactions of these species at the surface, leading to the deposition of solid silicon atoms and the desorption of gaseous hydrogen according to:

$$\mathrm{Si}_n\mathrm{H}_{2m} \xrightarrow{R^S_{\mathrm{Si}_n\mathrm{H}_{2m}}} n\ \mathrm{Si\ (s)} + m\ \mathrm{H_2\ (g)}, \tag{8.4}$$

$$\mathrm{Si}_n\mathrm{H}_{2m+1} \xrightarrow{R^S_{\mathrm{Si}_n\mathrm{H}_{2m+1}}} n\ \mathrm{Si\ (s)} + m\ \mathrm{H_2\ (g)} + \mathrm{H\ (g)}, \tag{8.5}$$

where $n = 1, 2$, or 3, and $m = 0, 1, 2, 3$, or 4. The molar reaction rate R^S_i for the decomposition of gas species i is given as

$$R^S_i = \frac{\gamma_i}{1 - \gamma_i/2} \frac{P f_i}{(2\pi m_i R T_s)^{1/2}}, \tag{8.6}$$

Table 8.4: Fit parameters of the forward reaction rates (2.21) for the benchmark problem. The parameter β_k is dimensionless, while E_k has unit kJ $\cdot$ mol^{-1} and A_k depends on the order of the reaction, but is expressed in units mole, m^3 and s.

Reaction	A_k	β_k	E_k
$SiH_4 \rightleftharpoons SiH_2 + H_2$	1.09×10^{25}	−3.37	256
$SiH_4 \rightleftharpoons SiH_3 + H$	3.69×10^{15}	0.0	390
$Si_2H_6 \rightleftharpoons SiH_4 + SiH_2$	3.24×10^{29}	−4.24	243
$SiH_4 + H \rightleftharpoons SiH_3 + H_2$	1.46×10^{7}	0.0	10
$SiH_4 + SiH_3 \rightleftharpoons Si_2H_5 + H_2$	1.77×10^{6}	0.0	18
$SiH_4 + SiH \rightleftharpoons Si_2H_3 + H_2$	1.45×10^{6}	0.0	8
$SiH_4 + SiH \rightleftharpoons Si_2H_5$	1.43×10^{7}	0.0	8
$SiH_2 \rightleftharpoons Si + H_2$	1.06×10^{14}	−0.88	189
$SiH_2 + H \rightleftharpoons SiH + H_2$	1.39×10^{7}	0.0	8
$SiH_2 + H \rightleftharpoons SiH_3$	3.81×10^{7}	0.0	8
$SiH_2 + SiH_3 \rightleftharpoons Si_2H_5$	6.58×10^{6}	0.0	8
$SiH_2 + Si_2 \rightleftharpoons Si_3 + H_2$	3.55×10^{5}	0.0	8
$SiH_2 + Si_3 \rightleftharpoons Si_2H_2 + Si_2$	1.43×10^{5}	0.0	68
$H_2SiSiH_2 \rightleftharpoons Si_2H_2 + H_2$	3.16×10^{14}	0.0	222
$Si_2H_6 \rightleftharpoons H_3SiSiH + H_2$	7.94×10^{15}	0.0	236
$H_2 + SiH \rightleftharpoons SiH_3$	3.45×10^{7}	0.0	8
$H_2 + Si_2 \rightleftharpoons Si_2H_2$	1.54×10^{7}	0.0	8
$H_2 + Si_2 \rightleftharpoons SiH + SiH$	1.54×10^{7}	0.0	168
$H_2 + Si_3 \rightleftharpoons Si + Si_2H_2$	9.79×10^{6}	0.0	198
$Si_2H_5 \rightleftharpoons Si_2H_3 + H_2$	3.16×10^{14}	0.0	222
$Si_2H_2 + H \rightleftharpoons Si_2H_3$	8.63×10^{8}	0.0	8
$H + Si_2 \rightleftharpoons SiH + Si$	5.15×10^{7}	0.0	22
$SiH_4 + H_3SiSiH \rightleftharpoons Si_3H_8$	6.02×10^{7}	0.0	0
$SiH_2 + Si_2H_6 \rightleftharpoons Si_3H_8$	1.81×10^{8}	0.0	0
$SiH_3 + Si_2H_5 \rightleftharpoons Si_3H_8$	3.31×10^{7}	0.0	0
$H_3SiSiH \rightleftharpoons H_2SiSiH_2$	1.15×10^{20}	−3.06	28

Table 8.5: Fit parameters of the gas phase equilibria constants (8.2) for the benchmark problem. The parameter β_{eq} is dimensionless, while E_{eq} has unit kJ $\cdot$ mol^{-1} the unit of A_{eq} depends on the order of the reaction, but is expressed in units mole, m^3 and s.

Reaction	$A_{k,eq}$	$\beta_{k,eq}$	$E_{k,eq}$
$SiH_4 \rightleftharpoons SiH_2 + H_2$	6.85×10^{5}	0.48	235
$SiH_4 \rightleftharpoons SiH_3 + H$	1.45×10^{4}	0.90	382
$Si_2H_6 \rightleftharpoons SiH_4 + SiH_2$	1.96×10^{12}	−1.68	229
$SiH_4 + H \rightleftharpoons SiH_3 + H_2$	1.75×10^{3}	−0.55	−50
$SiH_4 + SiH_3 \rightleftharpoons Si_2H_5 + H_2$	1.12×10^{-6}	2.09	−6
$SiH_4 + SiH \rightleftharpoons Si_2H_3 + H_2$	1.82×10^{-4}	1.65	21
$SiH_4 + SiH \rightleftharpoons Si_2H_5$	1.49×10^{-10}	1.56	−190
$SiH_2 \rightleftharpoons Si + H_2$	1.23×10^{2}	0.97	180
$SiH_2 + H \rightleftharpoons SiH + H_2$	2.05×10^{1}	−0.51	−101
$SiH_2 + H \rightleftharpoons SiH_3$	2.56×10^{-3}	−1.03	−285
$SiH_2 + SiH_3 \rightleftharpoons Si_2H_5$	1.75×10^{-12}	1.60	−241
$SiH_2 + Si_2 \rightleftharpoons Si_3 + H_2$	5.95×10^{-6}	1.15	−225
$SiH_2 + Si_3 \rightleftharpoons Si_2H_2 + Si_2$	2.67×10^{0}	−0.18	59
$H_2SiSiH_2 \rightleftharpoons Si_2H_2 + H_2$	1.67×10^{6}	−0.37	112
$Si_2H_6 \rightleftharpoons H_3SiSiH + H_2$	1.17×10^{9}	−0.36	235
$H_2 + SiH \rightleftharpoons SiH_3$	1.42×10^{-4}	−0.52	−183
$H_2 + Si_2 \rightleftharpoons Si_2H_2$	7.47×10^{-6}	−0.37	−216
$H_2 + Si_2 \rightleftharpoons SiH + SiH$	1.65×10^{3}	−0.91	180
$H_2 + Si_3 \rightleftharpoons Si + Si_2H_2$	1.55×10^{2}	−0.55	189
$Si_2H_5 \rightleftharpoons Si_2H_3 + H_2$	1.14×10^{6}	0.08	210
$Si_2H_2 + H \rightleftharpoons Si_2H_3$	3.43×10^{-4}	−0.31	−149
$H + Si_2 \rightleftharpoons SiH + Si$	1.19×10^{3}	−0.88	29
$SiH_4 + H_3SiSiH \rightleftharpoons Si_3H_8$	7.97×10^{-16}	2.48	−233
$SiH_2 + Si_2H_6 \rightleftharpoons Si_3H_8$	1.36×10^{-12}	1.64	−233
$SiH_3 + Si_2H_5 \rightleftharpoons Si_3H_8$	1.06×10^{-14}	1.85	−318
$H_3SiSiH \rightleftharpoons H_2SiSiH_2$	9.58×10^{-3}	0.50	−50

where T_s denotes the temperature of the wafer surface and f_i is the species mole fraction computed from relation (2.5).

The sticking coefficient γ_i of species i, $0 \leq \gamma_i \leq 1$, is equal to one for all silicon containing species, except for

- $\gamma_{Si_3H_8} = 0$,
- $\gamma_{Si_2H_6} = 0.537\exp(\frac{-9400}{T_s})$, and,
- $\gamma_{SiH_4} = {}^1/_{10}\gamma_{Si_2H_6}$.

In Kleijn (2000) it is remarked that it is not clear which values of the reactive sticking coefficients are used for Si_3H_8 and Si_2H_5 in Coltrin et al. (1989). In the text of their work, Coltrin et al. (1989) mention values equal to one for the sticking coefficients of Si_3H_8 and Si_2H_5, whereas the presented results seem to indicate that actually values equal to zero are used. In this thesis, we use the values used in Kleijn (2000), which are given above.

For this second chemistry model both concentration diffusion and thermal diffusion are considered. The effective multi-component diffusion coefficients $\mathbb{D}'_i$ in expression (2.13) for the ordinary diffusion flux are fitted as function of the temperature according to

$$\mathbb{D}'_i = \mathbb{D}'_{i,300}\left(\frac{T}{300}\right)^{\beta_{D,i}}. \tag{8.7}$$

The fitting constants $\mathbb{D}'_{i,300}$ and $\beta_{D,i}$ are listed in Table 8.6. For a dilute gas mixture as in the present study, with all species except helium present in trace amounts only, the multi-component thermal diffusion coefficient $\mathbb{D}^T_i$ in the thermal diffusion flux for species i, see expression (2.16), is fitted as a function of temperature, species concentration and density as

$$\mathbb{D}^T_i = \rho\omega_i\alpha_{TD,i}\mathbb{D}'_i, \tag{8.8}$$

see also Kleijn (1995). The fitting constants $\alpha_{TD,i}$ are listed in Table 8.6. The molecular weights of the reactive gaseous species in the gas mixture are listed in Table 8.6 as well.

8.2 Reactor Geometry and Configuration

This section is devoted to the description of the two reactor geometries and configurations mentioned in the introduction of this chapter. For both configurations the gas mixture at the reactor inlet consists of 0.1 mole% silane diluted in the inert carrier gas helium. Further, for both configurations the temperature of the gas mixture at the inflow is 300 K, and the susceptor is heated up to 1000 K. Two-dimensional computations have also been performed for other wafer temperatures.

Table 8.6: Fitted properties of the various species in the gas mixture according to expression (8.7) and (8.8), and the molecular weights of the reactive species. The unit of the fitting constant $\mathbb{D}'_{i,300}$ is $m^2 \cdot s^{-1}$, whereas the fitting constants β_D and α_{TD} are dimensionless. The unit of molecular weight is $kg \cdot mol^{-1}$.

Species	$\mathbb{D}'_{i,300}$	β_D	α_{TD}	m_i
H	2.66×10^{-4}	1.67	−0.25	0.001008
H_2	1.58×10^{-4}	1.65	−0.16	0.002016
Si	6.29×10^{-5}	1.75	0.57	0.028086
SiH	7.20×10^{-5}	1.66	0.73	0.029094
SiH_2	6.78×10^{-5}	1.67	0.80	0.030102
SiH_3	6.30×10^{-5}	1.67	0.85	0.031110
SiH_4	5.86×10^{-5}	1.67	0.91	0.032118
Si_2	5.34×10^{-5}	1.75	0.74	0.056172
Si_2H_2	5.03×10^{-5}	1.67	1.13	0.058188
Si_2H_3	4.88×10^{-5}	1.67	1.17	0.059196
H_2SiSiH_2	4.74×10^{-5}	1.67	1.20	0.060204
H_3SiSiH	4.74×10^{-5}	1.67	1.20	0.060204
Si_2H_5	4.59×10^{-5}	1.67	1.24	0.061212
Si_2H_6	4.47×10^{-5}	1.67	1.24	0.062219
Si_3	4.82×10^{-5}	1.75	0.84	0.084258
Si_3H_8	3.62×10^{-5}	1.67	1.61	0.092321

More details on the reactor configuration for the two-dimensional axisymmetric simulations are given in Section 8.2.1. The three-dimensional reactor is discussed in Section 8.2.2.

8.2.1 Two-Dimensional Reactor

The first reactor configuration is illustrated in Figure 8.1. Because of axisymmetry, the computational domain consists of one half of the (r-z) plane. The boundary conditions along the computational domain are summarized in Figure 8.1.

The pressure in the reactor is equal to the atmospheric pressure. From the top a gas mixture, consisting of 0.1 mole% silane diluted in helium, enters the reactor with a uniform temperature T_{in} = 300 K and uniform velocity v_{in} = 0.1 m/s. At a distance of 0.1 m below the inlet a susceptor with a diameter of 0.3 m and a surface temperature of T_s = 1000 K is placed. In the hot region above the susceptor the reactive gas silane decomposes into silylene and hydrogen. On this susceptor surface reactions take place leading to the deposition of solid silicon. The outer walls have a temperature T_{wall} = 300 K. Like the benchmark problem in Kleijn (2000), we study the case where the susceptor is *not* rotating. Then, strong radial variations in the species concentrations, the velocity profile and the temperature profile are observed. For a susceptor rotating a suitable speed the flow field in the reactor is virtually one-dimensional, see, for instance, Kleijn (2000). The rotating susceptor case is *not* studied in this thesis.

Since the gas phase reactants are highly diluted, we use the steady state velocity-, temperature-, density- and pressure fields, see Section 2.6. The steady state flow problem is solved using the computer code CVDMODEL, which has been tested in detail over the last decades, see, for instance, Kleijn et al. (1989), Kuijlaars et al. (1995) and Kleijn (2000). For the present reactor chamber, the streamlines and temperature field with the wafer temperature equal to 1000 K, which are computed via the code CVDMODEL, are shown in Figure 8.2.

8.2.2 Three-Dimensional Reactor

The reactor configuration that results in a three-dimensional computational domain is illustrated in Figure 8.3. The reactor chamber is a cuboid, with a length and a width of 0.35 m, and an height of 0.1 m. Exactly in the center of the top plane is a cuboidic inlet-pipe placed, with a length and a width of 0.10 m, and an height of 0.05 m. The square wafer, with edges of 0.30 m, is placed exactly in the center of the bottom plane of the cuboid. An outlet-pipe is attached to the bottom plane in the remaining 0.05 m of space between the wafer and the side wall.

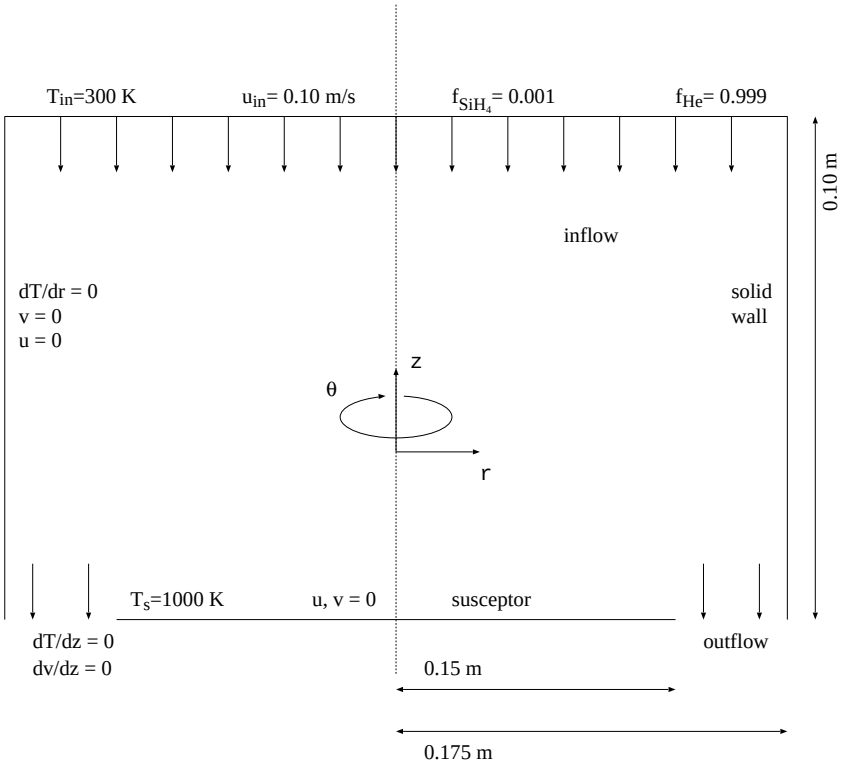


Figure 8.1: Reactor geometry and boundary conditions.

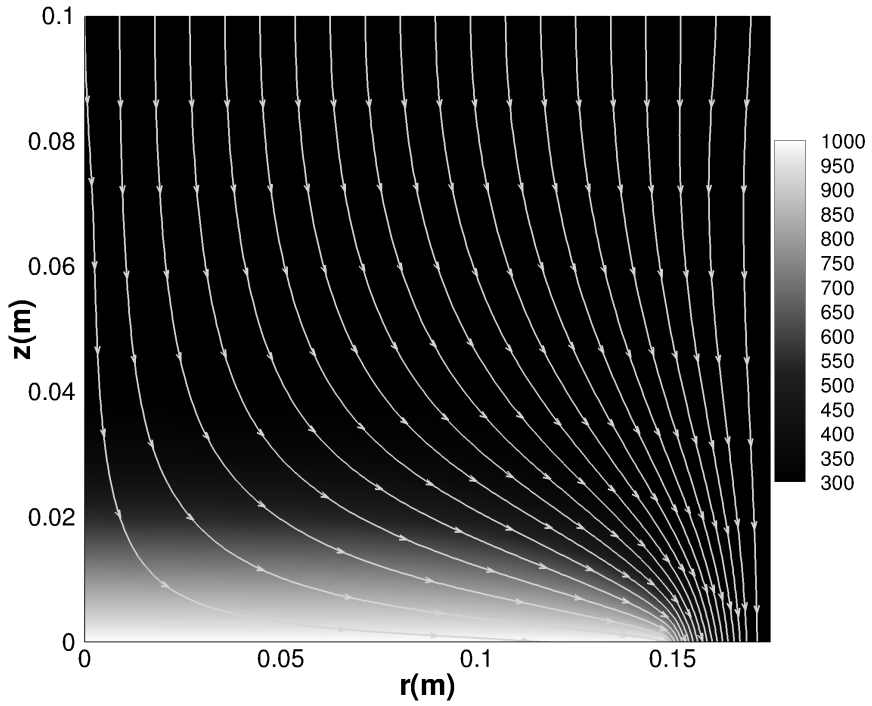
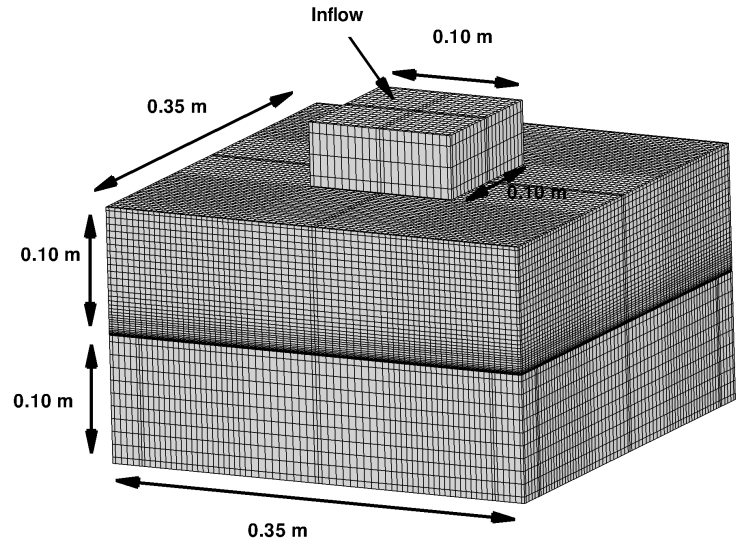
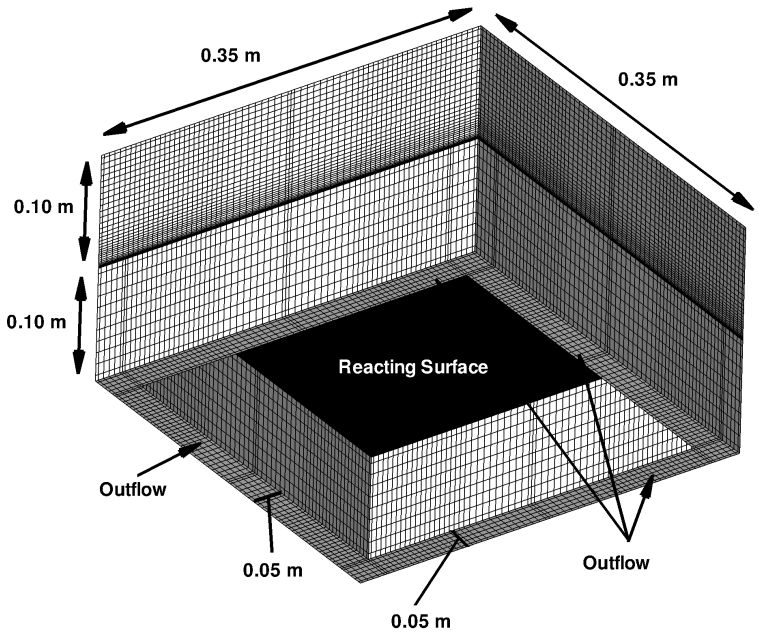


Figure 8.2: Streamlines and temperature field in Kelvin for the right half part of the reactor illustrated in Figure 8.1. The wafer temperature is equal to $T_s = 1000$ K.



(a) Side view



(b) Bottom view

Figure 8.3: Side and bottom view of the reactor geometry which leads to three-dimensional computational domain. The typical measures, which are given in Section 8.2.2, are illustrated as well.

The computational domain is reduced to one quarter of the actual reactor by imposing symmetry conditions on the two symmetry planes. The boundary conditions for the species transport equations (2.18) and the steady state flow problem are summarized in Figure 8.4.

The pressure in the reactor is equal to the atmospheric pressure. Further, for the steady state flow problem of the inert carrier gas helium the following boundary conditions are imposed:

- at the inlet we take a uniform inflow velocity profile with $v_{\text{in}} = 0.1$ m/s,
- the temperature of the gas mixture at the inlet is uniformly distributed with $T_{\text{in}} = 300$ K,
- the non-reacting solid walls are adiabatic resulting in zero normal temperature gradients, and,
- at all solid walls, both non-reacting and reacting, no-slip conditions are imposed.

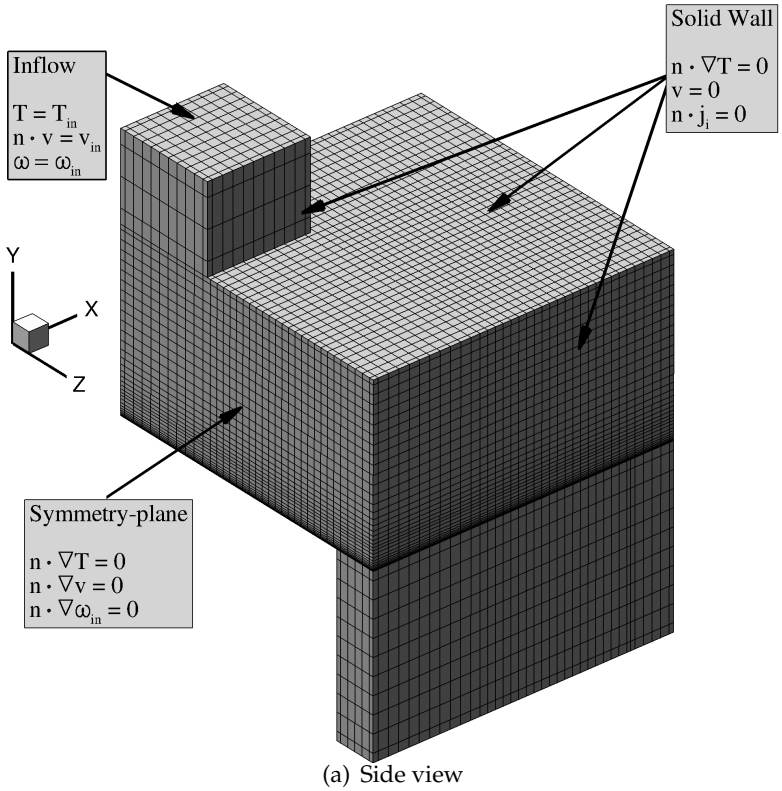
The steady state flow problem is solved using the proprietary CFD package CVD-X, which is developed at TNO Science and Industry, see TNO Science and Industry (2007).

Since the area of interest for the species equations (2.18) is the reactor chamber, the transient simulations are done for that part of the reactor only. Thus, the computational domain for the species equations (2.18) reduces, after imposing the symmetry conditions, into the configuration illustrated in Figure 8.4 without inflow- and outflow-pipe. The streamlines and temperature distribution for the reactor, without inflow- and outflow pipes, are shown in Figure 8.5.

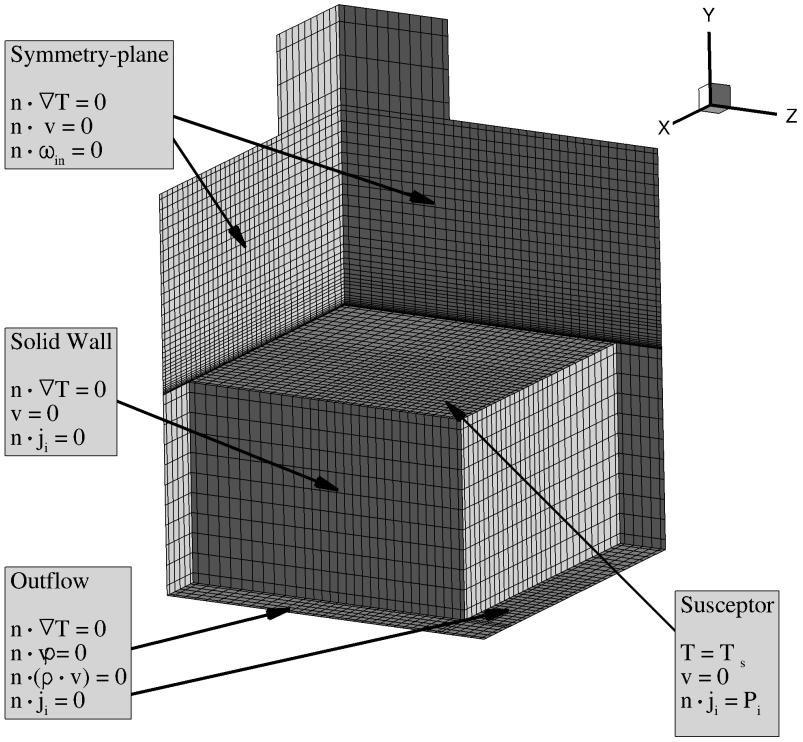
8.3 Validation of Two-Dimensional Steady State Solutions

Correctness of our steady state solution, obtained after long time integration, is validated against the steady state solution obtained with the software of Kleijn (2000). All simulations presented in this section are test cases where the wafer is *not* rotating.

The results presented in this section are obtained via Euler Backward time integration, in which the system of nonlinear algebraic equations is solved with the globalized Inexact Newton method discussed in Section 6.3. The Newton equation is solved by the preconditioned Bi-CGSTAB algorithm by van der Vorst (1992). As preconditioner, the block version of the incomplete LU factorization is used. The steady state solutions are obtained after long time transient simulations by the solution strategy as described above.



(a) Side view



(b) Symmetry-plane view

Figure 8.4: Three-dimensional reactor geometry and corresponding boundary conditions. Recall that j_i denotes the total diffusive flux of species i and P_i the net mass production rate of gaseous species i at the wafer. The computational grid has 35 grid cells in the x and z direction, and 32 in the y direction. Note that the grid is finer above the heated susceptor.

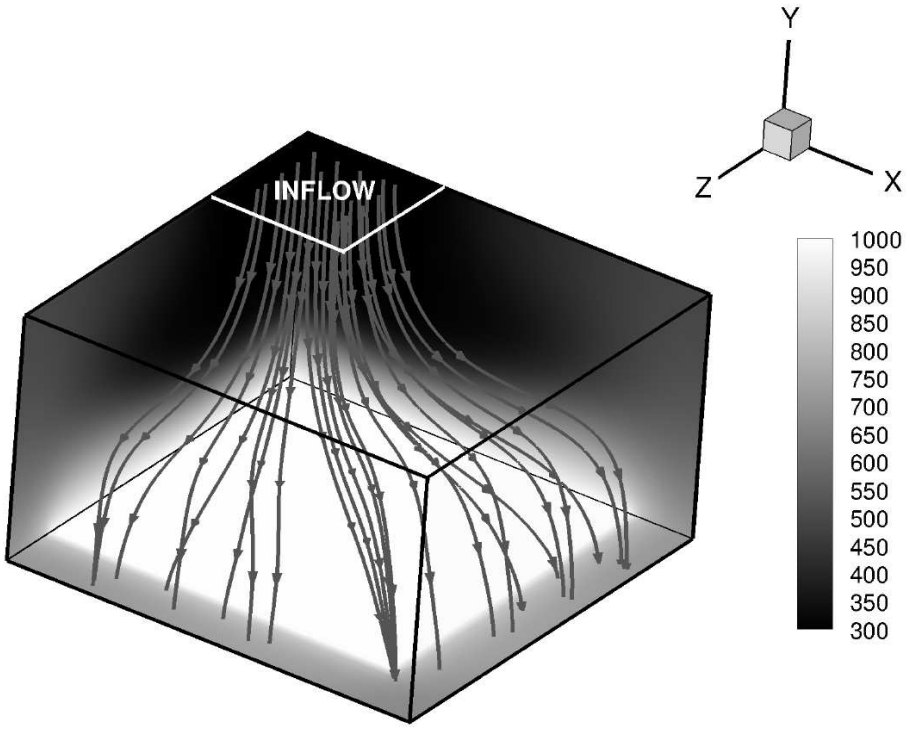


Figure 8.5: Streamlines and temperature distribution in Kelvin for the reactor chamber of Figure 8.4, without inflow- and outflow pipes. The flow field has been computed by CVD-X, see TNO Science and Industry (2007).

In van Veldhuizen et al. (2006b) solutions with gas phase Chemistry Model I are validated against solutions from Kleijn's code CVDMODEL. The agreement between the solutions was found to be excellent. Here, we restrict ourselves to the validation of the much more demanding Chemistry Model II.

8.3.1 Steady State Solutions for Chemistry Model II

In Figure 8.6 steady state mass fraction profiles are presented for some selected species, as well as the ones obtained by Kleijn (2000), for a wafer temperature equal to 1000 K. In this case, the total steady state deposition rate of silicon at the symmetry axis as found by Kleijn (2000) is 1.92 $^{nm}/_{s}$, whereas a deposition rate of 1.93 $^{nm}/_{s}$ has been found with the present numerical method as described above. Both values compare excellently to those obtained with the well-known 1-dimensional CVD simulation code SPIN within the Chemkin family Coltrin et al. (1993).

Figure 8.7 shows radial profiles of the total steady state deposition rates for wafer temperatures varied from 900 K up to 1100 K. Again, the agreement is for all wafer temperatures excellent. For all studied temperatures, the steady state growth rates obtained with the present transient solution method were found to differ less than 5% from those obtained with Kleijn's steady state code.

From Figure 8.7 various conclusions are drawn. First of all, it is concluded that the silicon deposition rate depends strongly on the wafer temperature, i.e., higher wafer temperatures lead to higher silicon deposition rates. For low wafer temperatures the deposition rate is mainly reaction limited, meaning that the deposition follows an Arrhenius dependence on temperature. However, for increasing wafer temperatures the chemical reactions become faster, causing mass transport to become rate limiting. In other words, the gases mainly responsible for silicon deposition cannot be transported to the wafer as fast as they are consumed chemically. As a result, the deposition rate does not increase any further for higher temperatures.

Secondly, the total deposition rates illustrated in Figure 8.7 are strongly non-uniform in radial direction. For wafer temperatures larger than or equal to 1000 K the total deposition rate increases towards the edge of the reacting surface, whereas for wafer temperatures below 1000 K the total deposition rate decreases towards the edge of the susceptor. These effects are, again, ascribed to the transition from kinetic to transport limitation, due to the decreasing thermal and concentration boundary layer thickness towards the susceptor edge. On the one hand, a decreasing thermal boundary layer thickness leads to reduced residence times of the gas species in the boundary layer, and thus reduced gas phase decomposition. On the other hand, a reduced concentration boundary layer thickness increases the mass transport for transport limited deposition growth. This explains

the increasing deposition rate towards the edge of the wafer for increasing wafer temperatures.

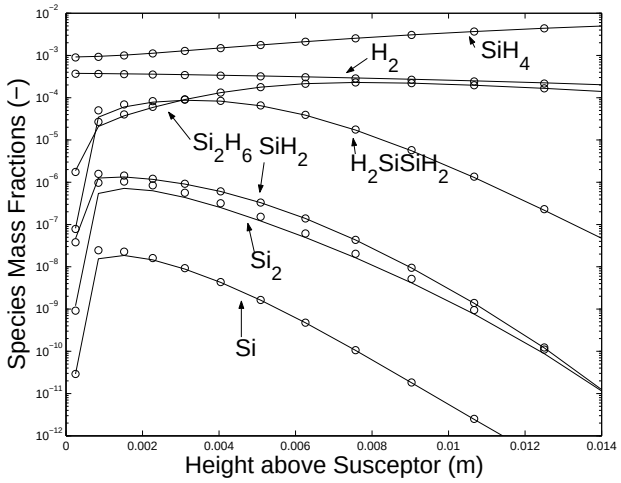


Figure 8.6: Axial steady state concentration profiles along the symmetry axis for some selected species. Solid lines are solutions from Kleijn (2000), circles are long time steady state results obtained with the present transient time integration methods.

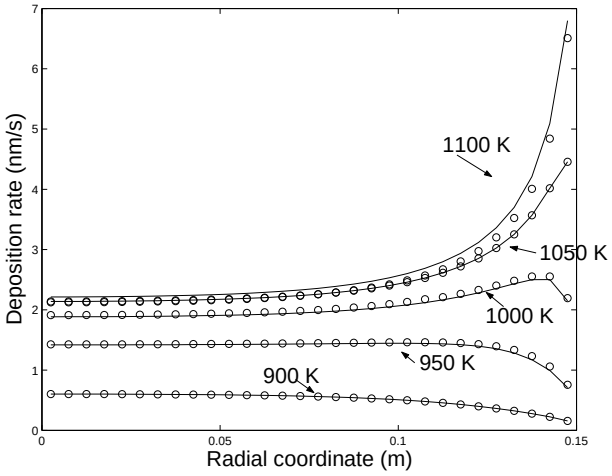


Figure 8.7: Radial profiles of the total steady state deposition rate for wafer temperatures varied from 900 K up to 1100 K. Solid lines are Kleijn's steady state results, circles are long time steady state results obtained with the present transient time integration method.

8.4 Transient Two-Dimensional Solutions for Chemistry Model II

In Figure 8.8 transient deposition rates are presented for some selected species, as well as the transient total deposition rate. It can be seen that the time dependent behavior of these deposition rates is monotonically increasing and stabilizes when the solution is in steady state. Also shown are the steady state deposition rates obtained with the software of Kleijn Kleijn (2000), which are in very good agreement with our current results. In Figure 8.9 we present transient total deposition rates for simulations with wafer temperatures varying from 900 K up to 1100 K. The time dependent behavior of all deposition rates is monotonically increasing until the species concentrations are in steady state.

In Figure 8.10 the transient behavior of the gas phase chemistry can be seen quite clearly. At time $t = 0.5$ s we see that reactive silane, entering the reactor from the top has not yet reached the reactive susceptor surface. At an inlet velocity of 0.1 $^{m}/_{s}$ and a distance between the inlet and the susceptor of 0.1 m this actually takes approximately 1 s. This is confirmed by Figure 8.8 and 8.9, in which it can be seen that deposition does not start until $t \sim 1$ s. A couple of seconds later, at time $t = 5$ s, when the CVD process is almost in steady state, we see that along the reacting surface almost all silane molecules either have been decomposed into volatile reaction products, or have been adsorped to the susceptor surface to form a solid silicon film, see Figure 8.10.

8.4.1 Further Discussion on the Deposition Rates for Chemistry Model II

Clearly, the purpose of (transient and steady state) simulations is to understand the relative effects of fluid transport and chemistry within these processes. For example, it can be determined which gas-species are most responsible for bringing silicon to the surface for deposition. As mentioned before, this depends on the wafer temperature, but can also depend on the carrier gas, see Coltrin et al. (1989). However, the latter is not considered in this study.

Figure 8.11 shows radial profiles of deposition rates due to various chemical species in the reaction mechanism for various wafer temperatures. It illustrates that the reactive intermediate species are mainly responsible for the deposition.

In Figure 8.12 the concentrations of SiH_2 for wafer temperatures $T_s = 900$ K and $T_s = 1100$ K are shown. We see indeed that for $T_s = 1100$ K the concentrations of SiH_2 are much higher along the reacting surface than for $T_s = 900$ K. Note that in Figure 8.12 the legends of both concentration fields differ two orders of magnitude. For the species concentrations of H_2SiSiH_2,

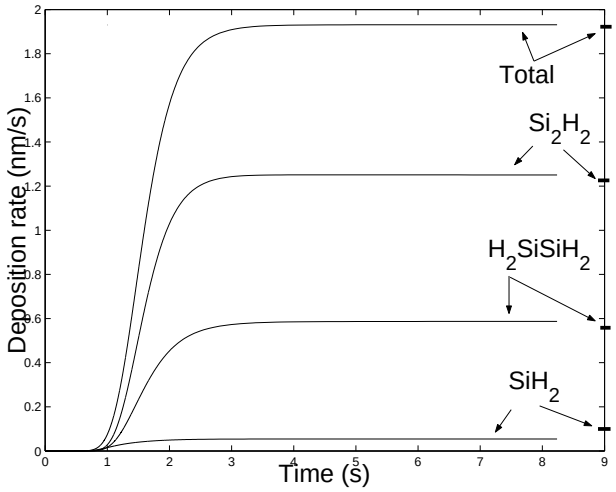


Figure 8.8: Transient deposition rates due to some selected species on the symmetry axis for simulations with a non-rotating wafer at 1000 K. On the right vertical axis: steady state deposition rates obtained with Kleijn's steady state code Kleijn (2000).

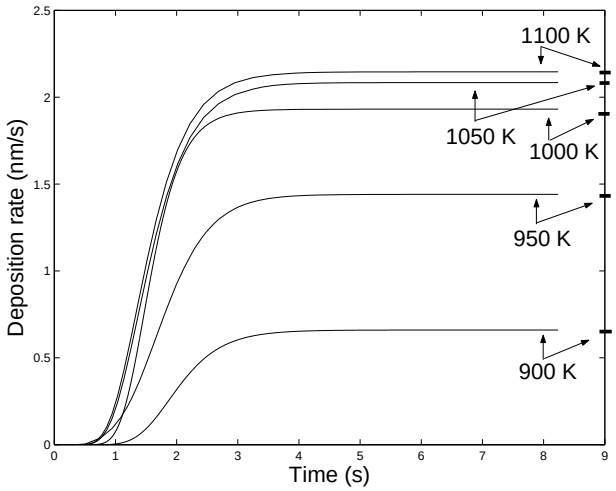
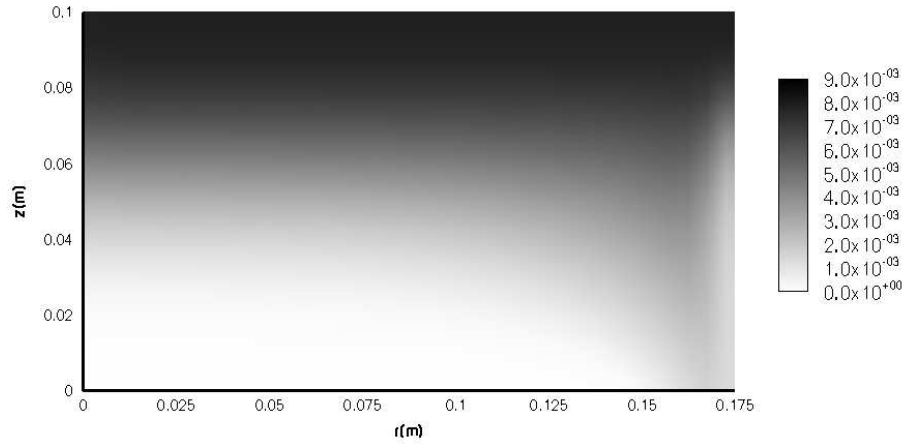
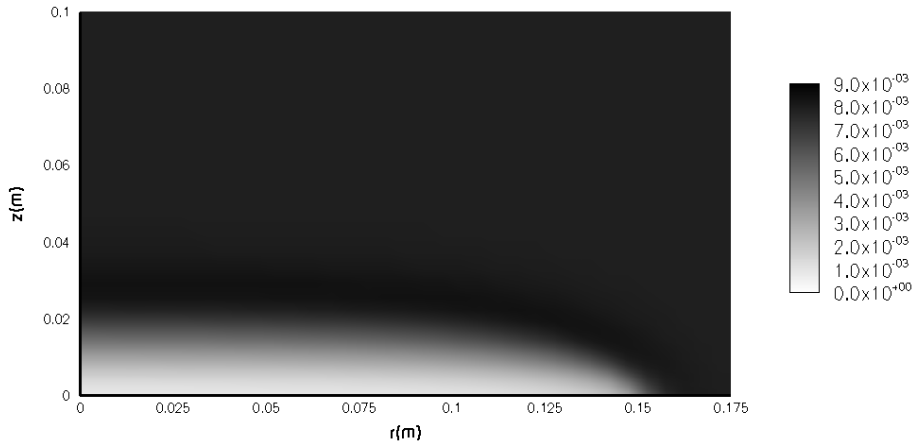


Figure 8.9: Transient total deposition rates on the symmetry axis for wafer temperatures varying from 900 K up to 1100 K. On the right vertical axis: steady state total deposition rates obtained with Kleijn's steady state code Kleijn (2000).

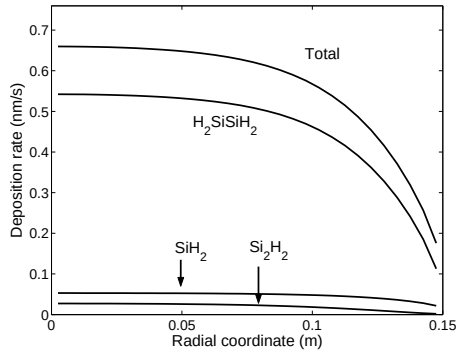


(a) $t = 0.5$ s

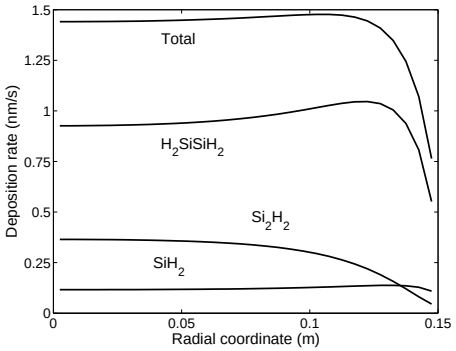


(b) $t = 5$ s

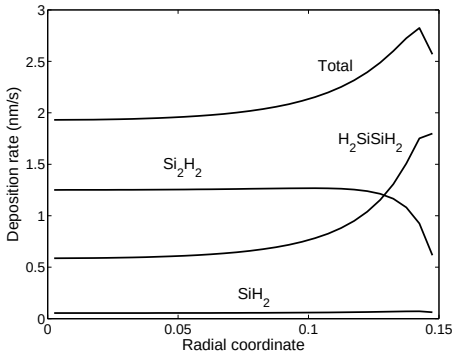
Figure 8.10: Mass fraction profiles of silane on time $t = 0.5$ s (a) and $t = 5$ s (b).



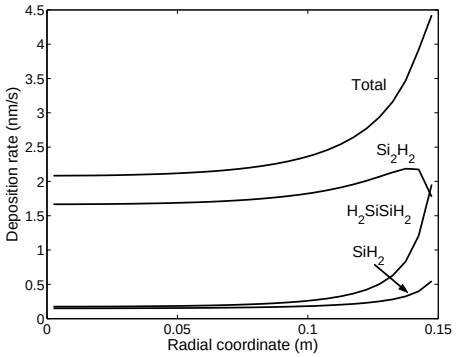
(a) 900 K



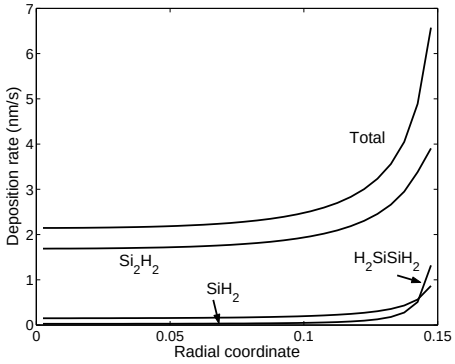
(b) 950 K



(c) 1000 K

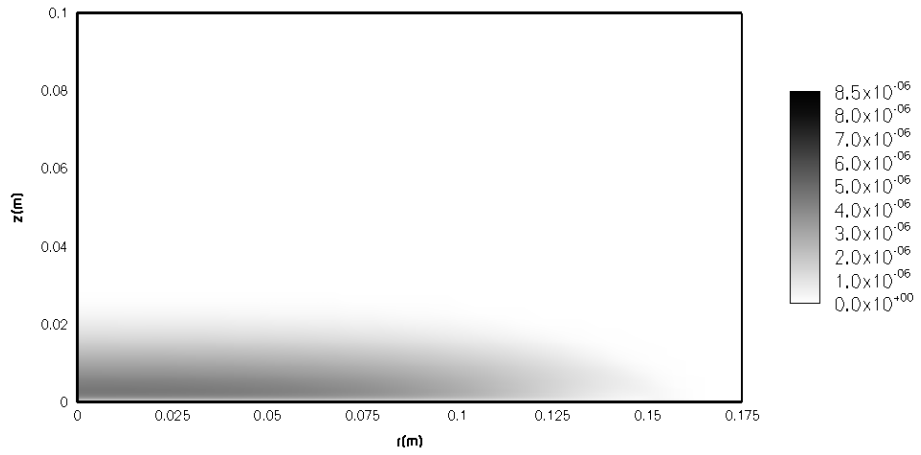


(d) 1050 K

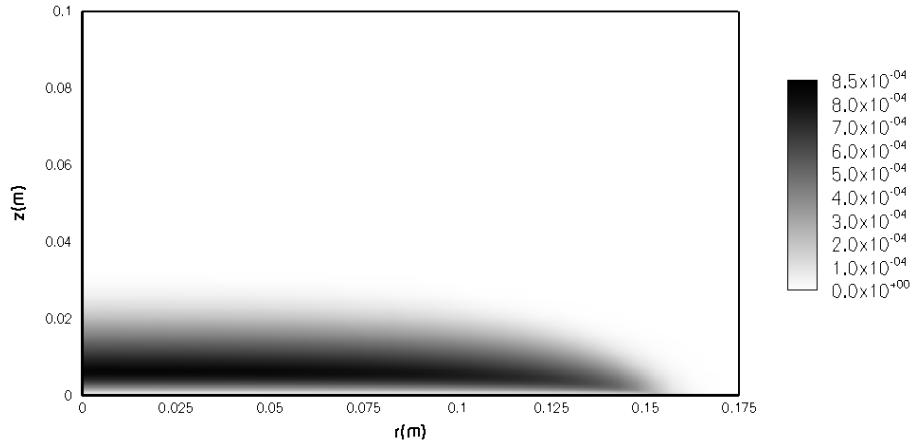


(e) 1100 K

Figure 8.11: Radial deposition profiles for wafer temperatures from 900 K up to 1100 K.

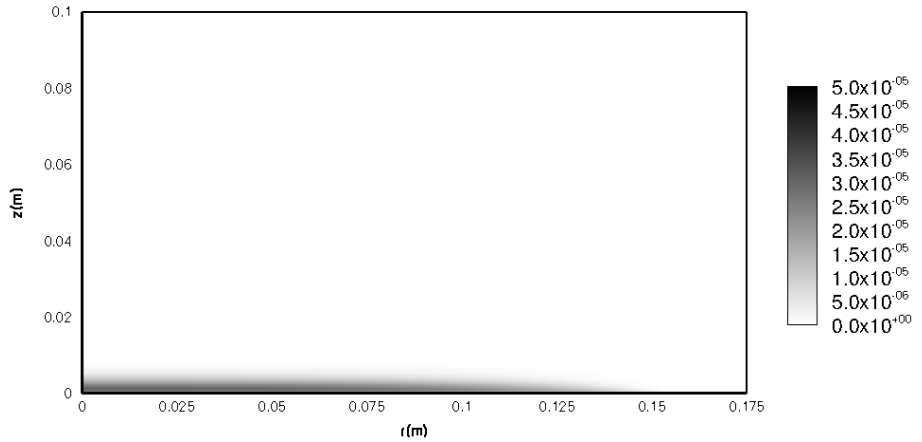


(a) T_s = 900 K

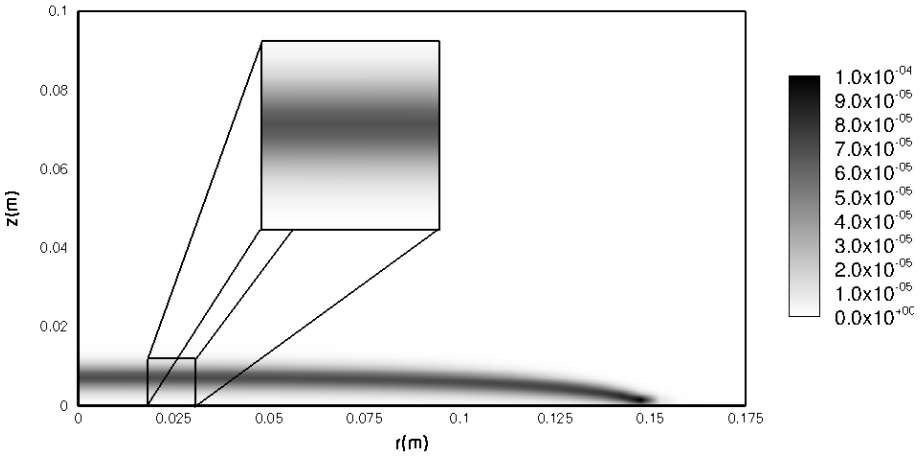


(b) T_s = 1100 K

Figure 8.12: Mass fraction profiles of Si_2H_2 for wafer temperature T_s = 900 K (a) and T_s = 1100 K (b). Note that the legends differ two orders of magnitude.



(a) T_s = 900 K



(b) T_s = 1100 K

Figure 8.13: Mass fraction profiles of H_2SiSiH_2 for wafer temperature T_s = 900 K (a) and T_s = 1100 K (b). Note that the legends are not identical.

which are shown in Figure 8.13, we see that the concentration of H_2SiSiH_2 for T_s = 1100 K is nearly zero along the wafer. This results in a relative small contribution of H_2SiSiH_2 to the deposition rate.

This section is concluded by discussing the influence of thermal diffusion on the (transient and steady state) deposition rate. In Section 2.4.2 it is mentioned that for reactors in which large temperature gradients are present, the thermal diffusion effect is important. In Figure 8.14 deposition rates for simulation with and without thermal diffusion are shown. The wafer temperature was set to 1000 K. Comparing transient computations with and without thermal diffusion gives an average difference of 20 – 25 % in deposition rate. For computations as these it is thus important to include thermal diffusion.

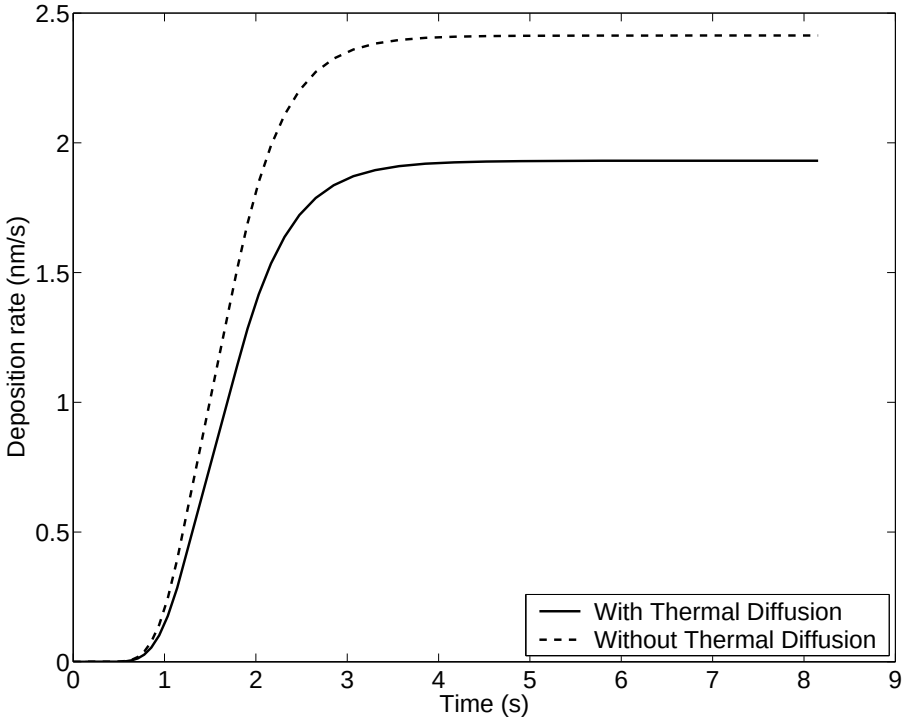


Figure 8.14: Total time accurate deposition rates on the symmetry axis as a result of computations with and without thermal diffusion. The wafer temperature is set to 1000 K.

8.5 Three-Dimensional Simulations

For the three dimensional transient simulations again only results are presented for the chemistry model with 17 (gas phase) species, 26 gas phase

reactions and 14 surface reactions. To the author's knowledge, similar results on this problem, or a problem of similar complexity have not been published earlier. First, a validation of the steady state solution is done.

8.5.1 Validation of Steady State Solution

To validate the steady state solution obtained via the simulations on the three-dimensional computational domain, we compare the species mass fractions along the intersection of the two symmetry planes. It is expected that the solution along this intersection line agrees well with the two-dimensional results at the $r = 0$ symmetry axis. In Figure 8.15 the steady state mass fraction profiles are presented for some selected species, as well as the ones obtained by Kleijn (2000), see also Figure 8.6. From Figure 8.15 can be concluded that these mass fraction profiles agree quite well.

The total steady state deposition rate of silicon along this symmetry line is 1.85 $^{nm}/_{s}$. Again, this value compares excellently to those found for the two-dimensional axisymmetric case in Section 8.3.

In Figure 8.16 the total deposition rate, the deposition rate due to most important growth species along the diagonal from the center of the wafer to the corner point of the wafer, as well as the radial deposition profiles belonging to two-dimensional axisymmetric simulations are shown. Comparing the two-dimensional and three-dimensional deposition profiels in the neighborhood of the symmetry axis it is concluded that all deposition rates agree very well. Towards the boundary of the wafer the flow fields of the two-dimensional and three-dimensional simulations differ too much to expect any agreement on the deposition rates at all.

Figure 8.17 shows the total steady state silicon deposition rate on the wafer. This figure clearly illustrates the strongly non-uniform, three-dimensional behavior of the reactor under the operation conditions described in Section 8.2.2.

8.5.2 Time Accurate Transient Results

In Figure 8.18 the total transient deposition rate of solid silicon on two locations on the wafer is displayed for $T_s = 1000$ K. Again, the deposition rates are monotically increasing in time. From this illustration it can be seen that at the corner point of the wafer it takes much longer for the deposition rate to reach its steady state value.

8.6 Discussion on the Integration Statistics

For all simulations in this section the simulations are being run from inflow conditions until the steady state solution is reached. We allow the maximum

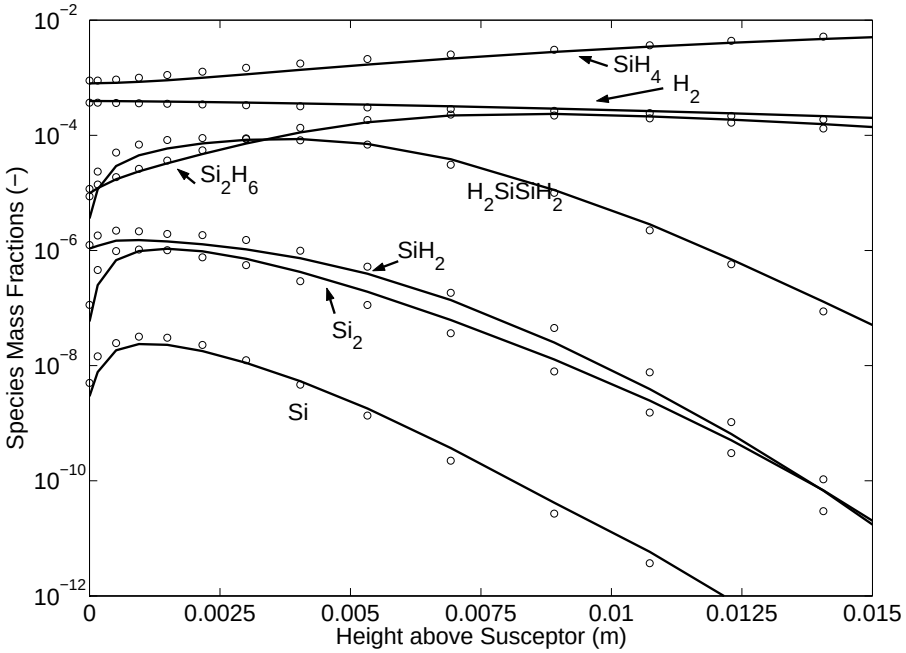


Figure 8.15: Axial steady state concentration profiles along the intersection of the two symmetry planes. Solid lines are the profiles belonging to the three-dimensional simulations, circles are profiles along the symmetry axis belonging to the two-dimensional axisymmetric case.

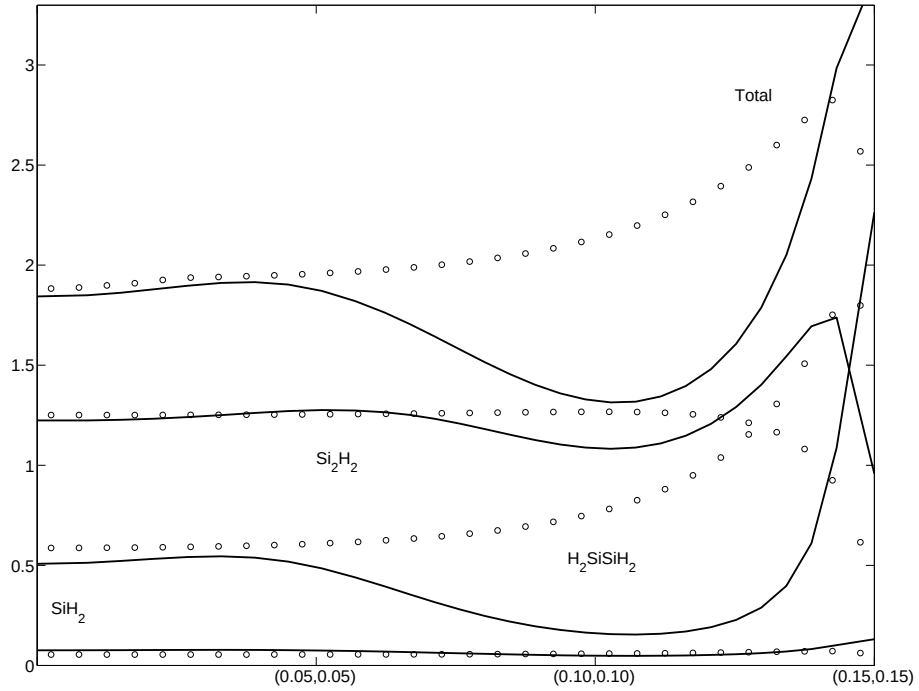


Figure 8.16: Solid lines are steady state deposition rates along the intersection of the origin and the cornerpoint of the wafer (i.e. $(x, y, z) = (0.15, 0, 0.15)$). The circles are radial steady state profiles belonging to the two-dimensional axisymmetric case.

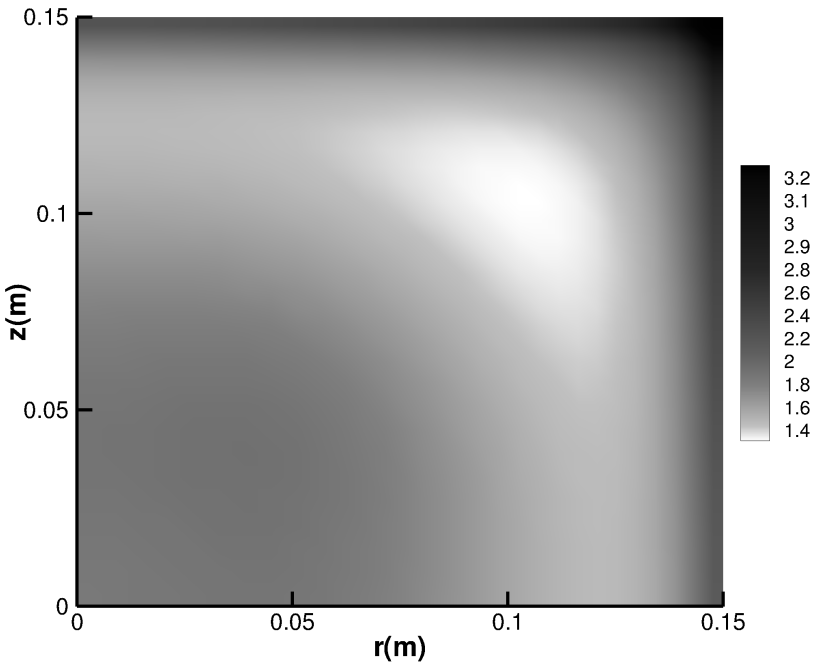


Figure 8.17: Steady state total deposition rate above the wafer.

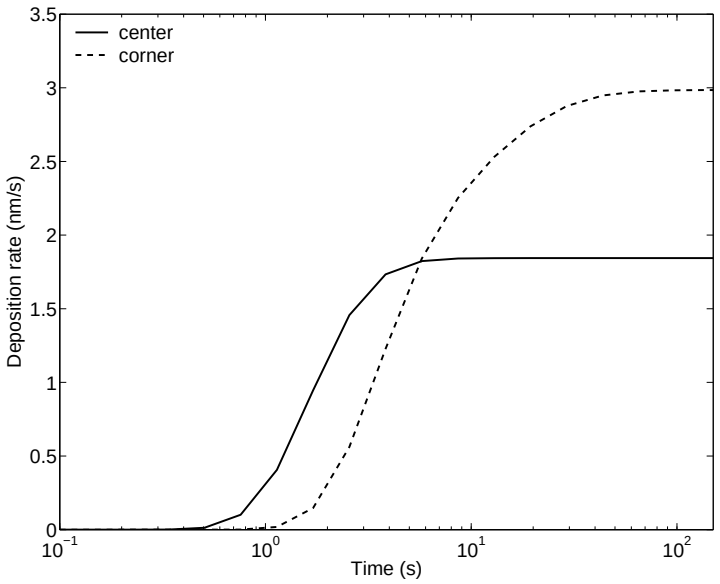


Figure 8.18: Transient total deposition rate in the center axis and in the corner (that is $(x, y, z) = (0.15, 0, 0.15)$) of the susceptor for a wafer temperature equal to $T_s = 1000$ K.

number of time steps to be 1000. With respect to the allowed number of Newton iterations per time step we remark the following. The strongly nonlinear reaction terms sometimes cause difficulties in finding the correct search direction. More specific, in the time frame right before steady state is reached, we experienced that to find the correct search direction might take a few extra Newton iterations. Therefore, the maximum number of Newton iterations is set to 50, whereas in most nonlinear solvers the maximum number of Newton iterations is set to 20 or 25.

Further, it has to be mentioned that in this section time accurate transient results are shown for different wafer temperatures varying from 900 up to 1100 K. Because of the large activation energies of some of the reactions (see Tables 8.1, 8.2, 8.4 and 8.5), such temperature differences lead to large qualitative and quantitative differences in the solutions. The behavior and the integration statistics of the computational method is, however, not influenced by the wafer temperature. Therefore, we will restrict ourselves to present the integration statistics for one wafer temperature per computational grid.

First, integration statistics are presented for the two-dimensional simulations. For the 6 species and 5 gas phase reactions chemistry model we present the most relevant integration statistics only. Integration statistics for the benchmark problem with 17 (gas phase) species, 26 gas phase reactions

and 14 surface reactions are discussed in detail. In particular, the behavior of the various preconditioners presented in Chapter 7 is highlighted.

Integration statistics for the three-dimensional simulations with the 17 (gas phase) species, 26 gas phase reactions and 14 surface reactions chemistry model are presented in Section 8.6.2.

8.6.1 Integration Statistics for Two-Dimensional Simulations

Chemistry Model I

We only present the integration statistics for the optimal combination of Euler Backward time integration, Inexact Newton method and preconditioner. Combining the Globalized Inexact Newton method with the Block ILU preconditioner is the most efficient choice with respect to total computational costs. The forcing term in the Inexact Newton method used is the one of Section 6.2.2, i.e.,

$$\eta_k = \gamma \frac{\|F(x_k)\|^2}{\|F(x_{k-1})\|^2}, \quad \text{with} \quad \gamma = 0.5. \tag{8.9}$$

The integration statistics for this configuration of the solver for three different computational grids are listed in Table 8.7.

The three computational grid are equidistant in radial direction and the grid spacing in axial direction is gradually decreasing towards the wafer surface. The coarsest grid consists of $nr = 35$ grid points in radial direction and $nz = 32$ grid points in axial direction. The remaining two grids consist of $nr = 35$ by $nz = 47$ grid cells, and $nr = 70$ by $nz = 82$ grid cells.

The effect of different grid sizes reflects in the number of linear iterations and CPU time. Looking at the CPU times in Table 8.7 it can be seen that the CPU times increases pretty much proportional with the number of grid points. Due to the high quality of the block ILU preconditioner, no rejected time steps are observed in these simulations.

For the other forcing terms and preconditioners similar integration statistics are found, see van Veldhuizen et al. (2008a). The number of rejected time steps due to negative species concentrations is restricted to one or two for the other preconditioners combined with the various forcing terms. Thus, it can be concluded that for this test problem with a small chemistry model the Projected Newton method will not improve the computational efficiency. On the other hand, the robustness will slightly increase, since no rejected time steps are found for all preconditioners.

Chemistry Model II

Due to its stronger nonlinearity and a larger stiffness, the number of Newton iterations increases for the 17 species and 26 gas phase reactions chemistry

Table 8.7: Number of operations for the 7 species and 5 reactions problem on three computational grids. The wafer temperature is for each computational grid different.

Grid size	35×32	35×47	70×82
Wafer temperature	1000 K	950 K	900 K
Newton iters	80	89	91
Rej. time steps	0	0	0
Acc. time steps	36	36	36
Line search	9	6	7
Lin iters	430	1,137	1,003
CPU time (sec)	140	230	690

model, see Tables 8.8 and 8.9. In these tables relevant statistics are listed for the forcing terms presented in Sections 6.2.1 and 6.2.2 and the preconditioners of Chapter 7. The simulations have been performed on the same three grids as discussed in the previous section. Further, in Figure 8.19 the CPU times are shown. With respect to CPU time it can be concluded that the incomplete factorization preconditioners perform significantly better than the block diagonal preconditioners. For the finer grid, the solver equipped with these preconditioners even do not return a time dependent solution over the time frame from inflow to steady state. The block incomplete factorization preconditioner is favorable over the ILU(0) preconditioner, when looking to CPU times. Further, note that for the most preconditioners using projected Newton instead of globalized Inexact Newton leads to slight improvements in terms of computational efficiency. However, combining the projected Newton method with the block diagonal preconditioners and forcing term (6.10) gives a considerable improvement of the computational effort needed.

From Tables 8.8 and 8.9 it can clearly be seen that for larger meshes the number of Bi-CGSTAB and Newton iterations increases considerably. With respect to the approximate linear relation between the number of grid points and the total CPU time, the following can be remarked. For the two coarser grids, i.e., the 35×32 and 35×47, the CPU times increase pretty much proportional with the number of grid points. However, the CPU times for the finer 70×82 grid do not scale linearly with the CPU times of the coarser grids. The finer grid is much finer in the thermal and concentration boundary layer than the other two grids, such that the system of species equations becomes much stiffer. This is being reflected in Jacobian matrices in the Newton iteration having much higher condition numbers. Even effective preconditioners like Block ILU are not able to drop the condition number sufficiently in order to obtain very fast Bi-CGSTAB convergence

as was the case on the coarser grids. Hence, for the simulations on this fine grid the number of linear iterations increases faster than for the coarser grid, which reflects itself in the CPU times.

Looking at the results of the ILU(0)- and Block ILU preconditioner, it can be concluded that both behave well with respect to positivity, and thus the differences between the projected and regular Newton method are minimal. As remarked above, the Block ILU preconditioner is overall computationally cheaper than ILU(0). This can be explained by the fact that a considerably smaller amount of linear iterations is needed, see Tables 8.8 and 8.9. Apparently, the extra fill-in generated by the Block ILU preconditioner, which is a combination of large and small entries, gives a much better approximation of the Jacobian matrix than the ILU(0) preconditioner.

Both block diagonal precondioners are performing bad with respect to positivity, in particular combined with forcing term (6.10). In this case the projected Newton method brings relief. The computational costs descrease by a factor 10, but are still higher than for the incomplete factorization type preconditioners. Mainly, this is due to the total number of linear iterations, which is between a factor of $5-10$ higher. Probably, the fact that the inverse of the Jacobian is approximated by inverting only the 'large' terms, is not close enough.

For the author it is not clear why the incomplete factorization preconditioners perform better with respect to positivity than the others. In Appendix A a related question is discussed. As far as known to the author, it is even not clear whether linear systems $Ax = b$, with A symmetric positive definite and satisfying the M-matrix property, and b component-wise positive, whose solution is approximated (up to a certain accuracy level) via the Conjugate Gradient method, is positive. Moreover, the conditions a preconditioner needs to fulfill in order to maintain this positivity property are unknown. We think, that answering these questions might explain the behavior observed in our experiments.

8.6.2 Integration Statistics for Three-Dimensional Simulations

As remarked at the beginning of Section 8.6, for the three-dimensional simulations we only report numerical results for the 17 species, 26 gas phase reactions and 14 surface reations model after Coltrin et al. (1989). Numerical experiments reveal that simulations running from inflow conditions to steady state give multiple time step rejections due to negativity for all preconditioners presented in Chapter 7. For all simulations on the three-dimensional meshes no solutions are found without the application of the Globalized Inexact *Projected* Newton method, see Section 6.4.

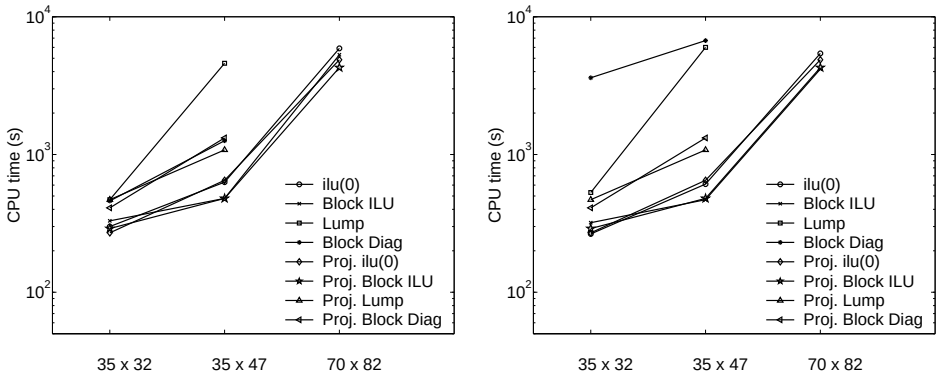
Numerical experiments have been carried out on two computational grids. The first one consists of $35 \times 32 \times 35$ grid cells, whereas the second one has $70 \times 70 \times 70$ grid cells. The integration statistics for the simulations

Table 8.8: Number of Bi-CGSTAB and Newton iterations for forcing terms (6.8) and (6.10) and various preconditioners on three computational grids for the Globalized Inexact Newton method. Choice 1 corresponds to forcing term (6.8) and Choice 2 corresponds to forcing term (6.10). If a steady state has not been reached then we write nf in the corresponding entry. Further, the number of rejected time steps due to negative species are specified.

	35×32			35×47			70×82		
	# lin. it.	# Newt.	# Neg.	# lin. it.	# Newt.	# Neg.	# lin. it.	# Newt.	# Neg.
Choice 1									
ILU(0)	848	108	1	2,073	205	0	7,522	353	0
Block ILU	624	111	2	772	153	0	2,069	325	0
Lump	4,987	152	0	72,091	1,241	177	nf	nf	nf
Block diag	4,219	149	1	22,498	285	2	nf	nf	nf
Choice 2									
ILU(0)	1129	101	1	2,541	194	1	11,100	395	3
Block ILU	838	104	2	859	148	0	2,144	299	0
Lump	7,927	149	2	106,833	1,391	122	nf	nf	nf
Block diag	13,371	1,379	403	28,140	2,054	583	nf	nf	nf

Table 8.9: Number of Bi-CGSTAB and Newton iterations for various forcing terms and preconditioners on three computational grids for the Globalized Inexact *Projected* Newton method. Choice 1 corresponds to forcing term (6.8) and Choice 2 corresponds to forcing term (6.10). If a steady state has not been reached then we write nf in the corresponding entry.

	35×32		35×47		70×82	
	# lin it.	# Newt.	# lin. it.	# Newt.	# lin. it.	# Newt.
Choice 1						
ILU(0)	825	101	2,086	211	7,930	385
Block ILU	556	97	772	153	1,921	308
Lump	4,654	149	10,655	250	nf	nf
Block diag	4,313	133	25,605	290	nf	nf
Choice 2						
ILU(0)	1,009	94	2,505	202	8,895	351
Block ILU	718	93	859	148	2,290	306
Lump	5,819	127	23,598	196	nf	nf
Block diag	6,275	125	29,253	223	nf	nf



(a) Forcing term 1 (b) Forcing term 2

Figure 8.19: CPU times for various grids and forcing terms.

from inflow conditions until steady state with the Euler Backward solver combined with the Globalized Inexact Projected Newton method for the coarsest grid are listed in Table 8.10. The statistics for the simulations on the finest grid with the same solver configuration are given in Table 8.11. The number of rejected time steps in both tables corresponds to the number of time that the Globalized Inexact *Projected* Newton method diverged; when using this method time step rejections due to negative species is impossible.

Two components of the Euler Backeard solver are expensive and might cause an increase in computational costs. The first on is the Krylov solver. In Table 8.10 we see that the total computational costs are mainly determined by the number of linear iterations, i.e., for the block diagonal preconditioners the number of Newton iterations is low and linear iterations is high.

On the other hand, per Newton iteration the Jacobian is evaluated analytically. For the numerical experiments in the present paper the partial derivatives of the chemistry term can be calculated at low cost, such that the exact Jacobian matrix is relatively cheaply assembled, but still a relative expensive component in the solver. For the experiments on the coarse grid we see an increasing number of Newton iterations for the 'weaker' block diagonal preconditioners compared to the incomplete factorization preconditioners. Again, this will increase the computational costs for the block diagonal preconditioners. For the numerical experiments on the fine grid we observe larger number of Newton iterations for the incomplete factorization preconditioners. Apparently, for these effective preconditioners in some time steps the Newton step was oversolved. For the ILU(0) preconditioner this phenomena leads to somewhat longer simulations times.

When going to finer grids in these three-dimensional simulations the preconditioned Krylov solvers can be accounted for the rise in computational costs. Generally, the relation between the number of grid points and

total cpu time will not longer scale linearly, as was the case in the two-dimensional axisymmetric case. As can be seen from Tables 8.10 and 8.11, eight times more grid points leads to approximately 35 times higher CPU times.

To summarize, from Tables 8.10 and 8.11 the following conclusions can be drawn. In term of computational times the Block ILU preconditioner combined with the Globalized Inexact Projected Newton method is the best method to compute a fully time-accurate transient solution of laminar reacting gas flow problems. Secondly, the reduction of computational costs is most effectively done by reducing the computational costs of the linear solver, for instance by effective preconditioning.

Table 8.10: Number of operations for the 17 species and 26 reactions problem on the three-dimensional computational grid consisting of $35 \times 32 \times 35$ grid cells. The wafer temperature has been set to 1000 K.

Preconditioner	ILU(0)	block ILU	Lumped Jac	block diag
Newton	239	156	332	327
Rej. time steps	3	0	0	0
Acc. time steps	44	43	43	43
Line search	51	20	31	29
Lin iters	3,196	2,481	17,472	18,392
CPU (s)	20,100	17,500	28,000	29,000

Table 8.11: Number of operations for the 17 species and 26 reactions problem on the three-dimensional computational grid consisting of $70 \times 70 \times 70$ grid cells. The wafer temperature has been set to 1000 K.

Preconditioner	ILU(0)	block ILU	Lumped Jac	block diag
Newton iters	539	436	366	367
Rej. time steps	11	11	9	9
Acc. time steps	55	53	52	52
Line search	223	142	114	107
Lin. iters	7,830	5,525	47,105	48,810
CPU (hrs)	200	167	203	260

8.6.3 Integration Statistics for IMEX-RKC methods for Three-Dimensional Simulations

As remarked earlier in Section 5.6, the IMEX-RKC method described in Section 5.5, is an attractive alternative to the unconditional Euler Backward time integration method. However, the IMEX-RKC method is conditional positive, but its computational efficiency is independent of the number of spatial dimensions. The dimesion of the linear systems appearing in the solver do not change when going from two to three spatial dimensions.

Numerical simulations on the three dimensional meshes reveal that, again, the positivity requirement on the species concentrations is crucial for the computational costs. Repeatedly rejected time steps due to negativity are observed. Moreover, because the physical time towards steady state is larger than in the two-dimensional axisymmetric case, the computational costs for simulations from $t = 0$ until steady state are higher than for the projected Newton Euler Backward solver.

8.7 Comparing Projected Newton Methods with Clipping

As mentioned in Chapter 6, there are two approaches to avoid negative species concentrations, i.e., clipping on the time level, and Euler Backward time integration combined with the Projected Newton method. In this section we compare the numerical results of the two-dimensional benchmark problem with the 16 species and 26 reactions chemistry model (see also Section 8.4), obtained via these positivity conserving strategies.

Fortunately, for this benchmark problem the steady state solutions found with both the projected Newton and clipping methods are identical. However, looking at the time dependent solutions differences are found. To show the differences between the clipping on time level and the projected Newton method we compute the integral

$$\int_0^t Q_{\text{dep,Si}} dt, \tag{8.10}$$

where $Q_{\text{dep,Si}}$ is the molar deposition rate of silicon atoms at the reacting surface. Thus, integral (8.10) is the total number of moles of silicon atoms deposited in a certain time frame.

The integral (8.10) has been computed for the time frame between $t = 0$ s and $t = 2$ s, and for the time frame between $t = 0$ s and steady state . These computations have been done on the 35×32 and 35×47 meshes. On both spatial meshes we computed a time accurate solution, in which only spatial errors are present. Remark that due to the stringent accurary

requirements in the time accurate solution the time step size is very small, such that postivity is no issue.

The total number of moles of deposited silicon for both methods are presented for the purely transient time frame between inflow conditions and 2 s in Table 8.12, and from inflow conditions until steady state in Table 8.13. In both tables the number of moles of deposited silicon found in the time accurate solutions are listed as well. Comparing the number of mole of deposited silicon computed with both strategies, with a time accurate solution shows which strategy gives most accurate results. The spatial error is for all computations identical, such that the differences can only be assigned to errors due to time discretization and the solution techniques to solve the nonlinear system(s). Obviously, it is expected that the projected Newton method, which is mass conserving, is more accurate than clipping, in which mass is added when putting negative species concentrations to zero.

As can be found in Table 8.12, the number of moles of deposited silicon in the transient time frame between inflow and 2 s found by the clipping method differs about 8 % with the result found in the time accurate solution. For our projected Newton method the differences are in the order of 2 %.

Table 8.12: Number of moles of deposited silicon in the time frame from inflow conditions to 2 s for Projected Newton methods, clipping and a time accurate solution.

	Proj. Newt.	Clipping	Time accurate
35×32	$1.30 \cdot 10^{-5}$	$1.38 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$
35×47	$1.28 \cdot 10^{-5}$	$1.35 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$

Table 8.13: Number of moles of deposited silicon in the time frame from inflow conditions until steady state for Projected Newton methods and clipping. The difference in percents is listed as well.

	Proj. Newt.	Clipping	Time accurate
35×32	$1.65 \cdot 10^{-4}$	$1.68 \cdot 10^{-4}$	$1.64 \cdot 10^{-4}$
35×47	$1.62 \cdot 10^{-4}$	$1.64 \cdot 10^{-4}$	$1.60 \cdot 10^{-4}$

When measuring until steady state the differences between both approaches are smaller. For the clipping strategy a difference with the time accurate solution is about 2 – 2.5 % for both meshes. The projected Newton method gives a difference of 1 %, see Table 8.13. The smaller differences are explained by (i) both clipping and projected Newton eventually return the same steady state solution, and, (ii) the molar deposition rate just before

steady state is much larger than the deposition rate in the purely transient time frame.

Moreover, when looking at mass balances at time $t = 2$ s for the atoms $e =$ Si, H and He the differences between the projected Newton method and clipping are even more elaborated. At time $t = 2$ s we compute for atom e

$$\frac{\int_0^2 Q_{\text{in},e} - Q_{\text{dep},e} - Q_{\text{out},e}\, dt - \int_{\text{reactor}} c_e(2,r,z)\, dS}{\int_0^2 Q_{\text{in},e}\, dt}, \tag{8.11}$$

where $c_e(2,r,z)$ is the molar concentration of atom e in the reactor at time $t = 2$ s and spatial coordinate (r,z). For simulations with the projected Newton method on the 35×32 and 35×47 grids the absolute value of expression (8.11), which should be zero, is of order $O(10^{-8})$ for all atoms e. However, when using the clipping strategy, in which mass is added when negative species concentrations are set to zero, following values for expression (8.11) are found:

- expression (8.11) for the silicon atom is $-2.3 \cdot 10^{-2}$ for simulations on the 35×32 grid, and $-1.02 \cdot 10^{-2}$ for simulations on the 35×47 grid,
- for the H atoms expression (8.11) equals $-3.1 \cdot 10^{-2}$ mol on the 35×32 grid, and $-2.4 \cdot 10^{-2}$ for simulations on the 35×47 grid, and,
- for the helium atoms He expression (8.11) is of the order $O(10^{-10})$ for both grids.

Thus, on the 35×32 grid 2% is added to the total moles of silicon atoms that entered the reactor, and 3.1% is has been added to the total moles of H atoms that entered the reactor. On the 35×47 grid this is 1% and 2.4% for the silicon and hydrogen atoms respectively. This results clearly show that the projected Newton method preserves mass, whereas clipping fails. We believe that larger differences are found when performing numerical experiments for inherently transient chemically reacting flow problems.

8.8 Conclusions

From the numerical results on Chemical Vapor Deposition presented in this chapter several conclusions can be drawn. The numerical methods tested all use Euler Backward time integration and Globalized Inexact Newton methods to solve the nonlinear system in each time step. Also tests are performed where the Globalized Inexact Projected Newton method is used. Further, the performance of various preconditioners is compared.

For two-dimensional simulations of an axisymmetric reactor it is concluded that application of projected Newton methods instead of the 'regular' type of Newton methods gives occasionally an improvement in computational efficiency. These slight improvements are only observed when

the weaker block-diagonal type preconditioners are used. However, for three-dimensional experiments, the Projected Newton methods are indispensable. Further, we have shown through numerical experiments that traditional clipping methods, which are not mass conserving, give less accurate time dependent solutions than the projected Newton methods, which conserve mass.

On the other hand, the total computational costs are also determined by the efficiency of the linear solver. In this chapter preconditioned Krylov solvers are used. Various preconditioners are presented and compared. Choosing the best preconditioner, in this case block ILU, combined with the project Newton methods enables us to compute time dependent solutions, from inflow until steady state, on a $70 \times 70 \times 70$ grid with 17 reactive species.

CHAPTER 9

Numerical Modeling of Solid Oxide Fuel Cells

The emphasis of the numerical methods and simulation results in this thesis has been mainly on Chemical Vapor Deposition (CVD). As mentioned in Chapter 1 all numerical methods should also be applicable to the numerical modeling of other chemically reacting (laminar) flow problems. Another typical example of such an application are Solid Oxide Fuel Cells (SOFC). SOFC's are electrochemical conversion devices that produce electricity directly from oxidizing a fuel and have a wide variety of applications from use as auxiliary power units in vehicles to stationary power generation with outputs from 100 W to 2 MW, see for instance Singhal & Kendall (2003). Typically, they operate at temperatures between 600 °C and 1000 °C. The models and results discussed in this chapter are on Segmented-in-Series architectures of SOFCs, see for instance Gardner et al. (2000) and Nakamura et al. (2005). In Section 9.1 a brief introduction is given on SIS-SOFC modeling.

The mathematical and numerical difficulties in the numerical modeling of SIS-SOFCs and CVD are to a large extent identical. For both stationary and instationary computations the inherent stiffness of the reaction terms in the transport equations for the reactive species cause the simulations to be computationally expensive.

With respect to instationary simulations and positivity of the species mass fractions, it is remarked that the absence of advection terms in the SOFC model equations is favorable. As mentioned in Section 5.3.1, the second order Rosenbrock schemes are positive (in the linear sense) for all time step sizes for diffusion - reaction equations. From that perspective, the set of suitable positive time integration methods is larger for this particular

application. However, in this chapter we only present steady state solutions, which are computed by the (adapted) Euler Backward solver, combined with Inexact Newton methods and preconditioners as presented in Chapters 5, 6 and 7.

This chapter is organized as follows. In Section 9.2 the mathematical model of Solid Oxide Fuel Cells (SOFC) is shortly discussed, with special attention for the electrochemistry, species transport in porous media and elementary catalytic chemistry in so-called segmented-in-series SOFCs.

Section 9.3 is devoted to the numerical methods used in the simulations. In Section 9.4 some steady state numerical results are presented, which have already been published in Kee et al. (2008). Further, in Section 9.5 we enumerate a collection of mathematical challenges with respect to the numerical modeling of SIS-SOFCs.

9.1 Introduction

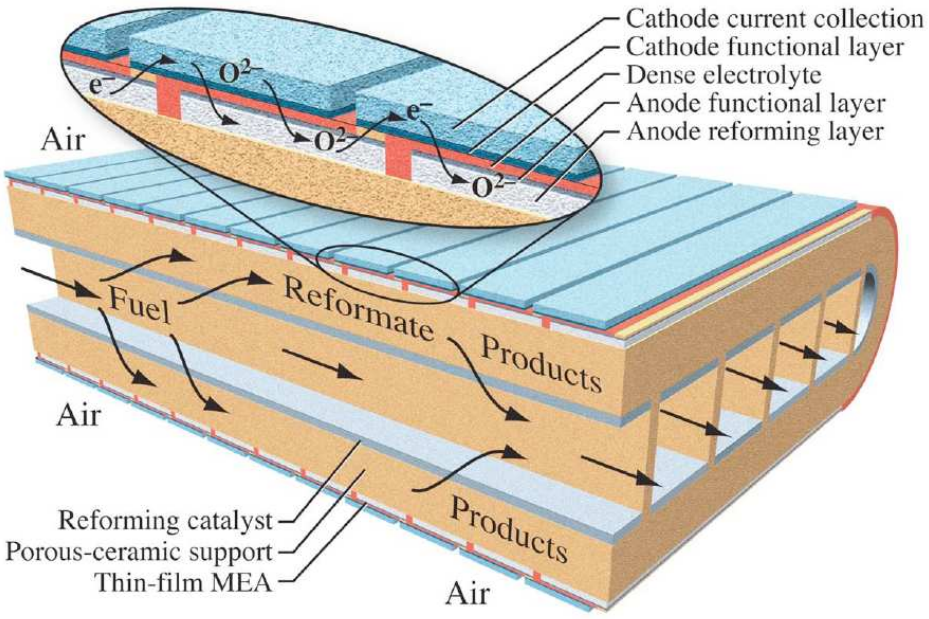


Figure 9.1: Segmented-in-series SOFC module after Kee et al. (2008)

In Figure 9.1 a section of a segmented-in-series (SIS) SOFC module is illustrated. In this module, the planar SIS cells are arrayed on the outside of a porous-ceramic support structure, where fuel is flowing inside the support and air flows on the outside. Each cell is composed of a membrane-electrode assembly (MEA) that consists of a cermet anode (e.g., Ni-YSZ),

dense electrolyte (e.g., YSZ), and composite cathode (e.g., LSM-YSZ). Each layer is usually on the order of a few tens of microns thick. The inset of Figure 9.1 shows that each cell is connected electrically in series with its neighboring cells. The width of each cell is typically in the order of tens of millimeters. As illustrated here, there is a porous catalyst layer applied to the interior of the support structure, which can assist reforming hydrocarbon fuel streams.

In each unit cell, electrons are introduced into the cathode via electric conduction. Electrochemical reduction of oxygen transfers the electron to an oxygen ion O^{2-} in the ion-conducting phase. With the dense electrolyte being purely an ion conductor, only oxygen ions can be transferred from the cathode into the dense electrolyte. On the anode side fuel is electrochemically oxidized by the oxygen ions to deliver electrons in the electronic-conducting phase. In Figure 9.2 this is illustrated in the balloon. For detailed descriptions we refer to Zhu et al. (2005) and Zhu & Kee (2008) and the references therein.

Rolls-Royce is developing such planar architectures, see Gardner et al. (2000) and Agnew et al. (2007), and call their particular design an Integrated Planar Solid Oxide Fuel Cell (IP-SOFC). Potential benefits of using SIS architectures over others are, for instance, the short current paths and the series connection that builds up voltage on the module, such that internal resistance is lowered. Furthermore, the cells can be fabricated at relative low cost technologies, see Kee et al. (2008). The model presented in this chapter of the thesis is a quantitative tool to assist evaluating design alternatives.

Over the last years significant advances have been made in developing numerical models for IP-SOFC systems. For instance, Haberman and Young developed a three-dimensional CFD model, which incorporates porous media flow, reforming chemistry and electrochemistry, to investigate the effects of fuel and air flow as well as heat and mass transport on the system level, see Haberman & Young (2004), Haberman & Young (2006) and Haberman & Young (2008). Costamagna et al. (2004) developed an electrochemical model, in which fluid and mass transport are coupled with chemical and electrochemical processes, to represent an IP-SOFC system.

The model presented in this chapter and published in Kee et al. (2008) is a two-dimensional numerical model for an SIS unit cell. Compared to prior literature, this model makes significant advances in the fundamental representation of chemistry and electrochemistry. Electric potentials for both ion- and electron-conducting phases are modeled throughout the entire cell. Hence, both ionic and electron fluxes are predicted throughout the system. Electrochemical charge-transfer chemistry depends on the local temperature, gas-phase composition, and electric-potential differences between phases. The spatial extent of the charge-transfer region depends on electrode structure, including primary particle sizes, phase densities, porosity, tortuosity, etc. Porous media gas-phase transport is represented

with a Dusty-Gas model. The model also represents elementary catalytic chemistry (typically tens of elementary reactions) within the anode, which is important to represent internal reforming when using hydrocarbon fuels.

9.2 Mathematical Description of SOFC

In Figure 9.2 the two-dimensional representation of an MEA unit cell of the SIS-SOFC module illustrated in Figure 9.1 is given. For the present study holds that the modeling of the distributed charge-transfer chemistry and catalytic reforming only consider the unit cell. Thus, the modeling of the fuel flow and the porous media transport in the support layer are not considered. Modeling these components and the system as a whole is certainly important. However, the modeling of the other components is relatively straightforward compared to the modeling of the chemistry in the MEA.

The balloon in Figure 9.2 shows ion and electron transport at the microscopic particle scale. Considering the dimensions of one MEA unit cell it is impractical to model at the particle scale. Instead, the problem is posed as continuum partial differential equations that describe the electric potentials for the electrode and electrolyte phases as well as Faradaic charge transfer between phases. Percolation theory is used to make the connection between particle and continuum representations, see Zhu & Kee (2008). The porous-media flow of gases in the pore spaces is modeled with a Dusty-Gas model. Catalytic reforming and partial oxidation within the anode is based upon an elementary reaction mechanism. Because the anode and cathode thicknesses of a SIS-SOFC cell are on the order of 50 μm, the electrochemical charge-transfer processes are likely distributed throughout most of the porous electrode structure, see Zhu & Kee (2008). Therefore, SIS models must accommodate charge-transfer electrochemistry throughout the MEA. The derivation of the distributed charge-transfer model used in this chapter is published in Zhu & Kee (2008). In Sections 9.2.1, 9.2.2 and 9.2.3 a summary of this model is presented.

9.2.1 Porous Media Transport and Chemistry

For chemically reacting gas flows in porous media the transport equation for gas-phase species i, $i = 1, \ldots, N$, is given by

$$\frac{\partial(\phi_g \rho \omega_i)}{\partial t} + \nabla \cdot \mathbf{j}_i = m_i \dot{s}_i, \tag{9.1}$$

where ϕ_g is the porosity and $\dot{s}_i$ is the molar production rate per unit volume of the gas phase species via thermal and electrochemical reactions. Accordingly, $\dot{s}_i$ is function of temperature, gas concentrations, surface species

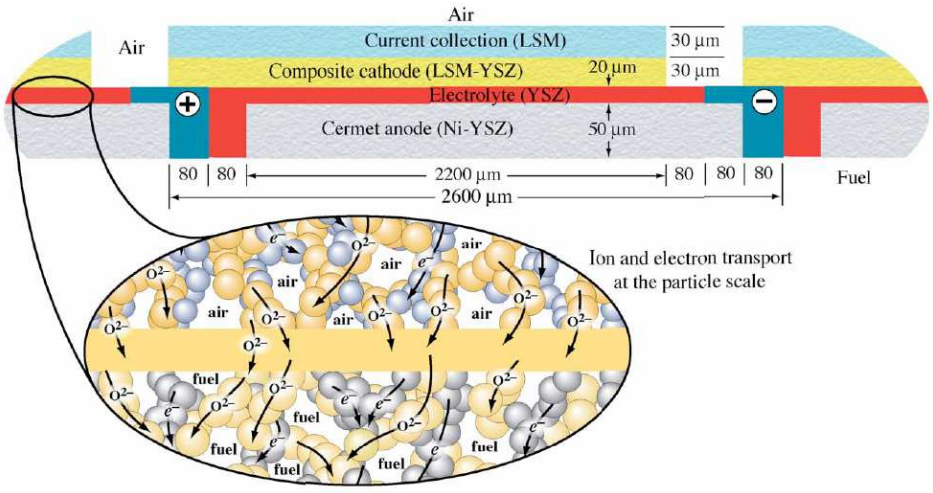


Figure 9.2: Unit cell and its physical dimensions used in the present study after Kee et al. (2008)

coverages and the electric potential among the electrode and electrolyte phases. The overall mass continuity equation is

$$\frac{\partial(\phi_g \rho)}{\partial t} + \sum_{i=1}^{N} \nabla \cdot \mathbf{j}_i = \sum_{i=1}^{N} m_i \dot{s}_i. \tag{9.2}$$

Remark that in this case the diffusive fluxes and molar production rate do not add up to zero. In the diffusive fluxes is also accounted for transport due to pressure gradients such that they do not add up to zero. Due to the dependence of the molar production rates on local electric potential differences, the molar production rate do not add up to zero.

Recall that instead of solving equation (9.1) for all species in the gas mixture, the mass fraction of species N is computed via the property that all mass fraction add up to one, i.e.,

$$\omega_N = 1 - \sum_{i=1}^{N-1} \omega_i. \tag{9.3}$$

The species mass fluxes $\mathbf{j}_i$ are evaluated by the the Dusty-Gas model, see Mason & Malinauskas (1983). Mathematically the Dusty-Gas model is denoted as the implicit relationship

$$\sum_{i \neq k} \frac{c_i \mathbf{J}_k - c_k \mathbf{J}_i}{c_{\text{tot}} D^e_{ki}} + \frac{\mathbf{J}_k}{D^e_{k,\text{Kn}}} = -\nabla c_k - \frac{c_k}{D^e_{k,\text{Kn}}} \frac{B_g}{\mu} \nabla P, \tag{9.4}$$

where c_i is the molar concentration of species i, $c_{\text{tot}} = {}^{P}/_{RT}$ the total molar concentration, B_g the permeability, $\mathbf{J}_i$ is the molar diffusion flux of species i and μ the mixtue viscosity. Molar diffusion and mass diffusion are related through

$$\mathbf{J}_i = \frac{\mathbf{j}_i}{m_i}. \tag{9.5}$$

Further, in equation (9.4), D^e_{ki} is the effective diffusion coefficient and $D^e_{k,\text{Kn}}$ is the effective Knudsen diffusion coefficient. Knudsen diffusion is a means of diffusion caused by gas-wall collisions between the gas molecules and the walls of the pores in the porous medium. Clearly, the Knudsen diffusion coefficients depends upon the porous media structure, which is characterized by the porosity, the average pore radius r_p and tortuosity τ_g. The effective binary and Knudsen diffusion coefficients are evaluated as

$$D^e_{ki} = \frac{\phi_g}{\tau_g} D_{ki}, \tag{9.6}$$

and

$$D^e_{i,\text{Kn}} = \frac{2}{3} \frac{r_p \phi_g}{\tau_g} \sqrt{\frac{8RT}{\pi m_i}}. \tag{9.7}$$

Remark that Knudsen diffusion becomes more important for Knudsen numbers Kn, defined as

$$\text{Kn} = \frac{\xi}{r_p}, \tag{9.8}$$

with ξ the mean free path length, larger than 0.01. Typically, the mean free path lengths for the species in the SOFC models considered in the present are of the order of the pore diameter. Thus, Kn > 0.01. The binary diffusion coefficients D_{ki} in expression (9.6) is determined from kinetic theory, see for instance Kee et al. (2003). Finally, the permeability can be evaluated from the Kozeny-Carman relationship as

$$B_g = \frac{\phi_g^3 d_p^2}{72\tau_g(1-\phi_g)^2}, \tag{9.9}$$

where d_p is the particle diameter. Further details of the Dusty Gas Model can be found in Zhu et al. (2005).

With respect to the molar production rate of the gas-phase species $\dot{s}_i$ the following is remarked. Typically, the pore spaces are sufficiently small such that the most likely collisions are between gas molecules and surfaces of the particles in the porous media. Consequently, the gas-phase homogeneous kinetics is usually negligible, such that $\dot{s}_i$ is a function of local temperature, gas composition, surface coverage and electric potential differences between electrode and electrolyte phases.

The boundary conditions needed to solve equations (9.1) and (9.2) are:

- at the interface with the air and fuel compartments the gas-phase composition is assumed to be equal to that within either the bulk fuel or air flow, and,
- at the interfaces with the dense electrolyte and interconnectors the gas-phase species fluxes vanish.

The results presented in this chapter use a reaction mechanism that is developed for Ni-YSZ composites, see Hecht et al. (2005). This mechanism, which does not specifically account for coke-formation reactions, considers 42 reactions among 6 gas-phase species and 12 additional surface-adsorbed species. It is presented in Table 9.1.

9.2.2 Charge Conservation

In the MEA structure there are two participating electric phases, i.e., the electron conducting phase and the ion-conducting phase. The electrochemical charge-transfer reactions depend on the electric-potential difference between these participating phases. The mathematical model of the charge transport involves three electric potentials:

- the electric potential for the electron-conducting phase in the anode Φ_a, and,
- the electric potential for the electron-conducting phase in the cathode Φ_c, and,
- the electric potential for the ion-conducting phase Φ_e,

which satisfy the conservation equations

$$\frac{\partial q_e}{\partial t} = \nabla \cdot \sigma_e^e \nabla \Phi_e - \begin{cases} \dot{s}_{a,e} & \text{within the anode} \\ 0 & \text{within the electrolyte} \\ \dot{s}_{c,e} & \text{within the cathode} \end{cases}, \tag{9.10}$$

$$\frac{\partial q_a}{\partial t} = \nabla \cdot \sigma_a^e \nabla \Phi_a + \dot{s}_{a,e} \quad \text{within the anode, and,} \tag{9.11}$$

$$\frac{\partial q_c}{\partial t} = \nabla \cdot \sigma_c^e \nabla \Phi_c + \dot{s}_{c,e} \quad \text{within the cathode.} \tag{9.12}$$

In equations (9.10), (9.11) and (9.12) q_m ($m = a, c, e$) is the charge in the particular phase, $\dot{s}_{m,e}$ ($m = a, c, e$) is the charge-transfer rate between the phases, σ_a^e is the effective conductivity of the electron-conducting phase in the anode, σ_c^e is the effective conductivity of the electron-conducting phase in the cathode and σ_e^e is the effective conductivity of the ion-conducting phase in the electrolyte.

Table 9.1: Heterogeneous reaction mechanism for CH_4 reforming on Ni-based catalysts. This mechanism is taken from Zhu et al. (2005).

$H_2 + (Ni) + (Ni) \rightarrow H(Ni) + H(Ni)$	$H(Ni) + H(Ni) \rightarrow (Ni) + (Ni) + H_2$
$O_2 + (Ni) + (Ni) \rightarrow O(Ni) + O(Ni)$	$O(Ni) + O(Ni) \rightarrow (Ni) + (Ni) + O_2$
$CH_4 + Ni \rightarrow CH_4(Ni)$	$CH_4(Ni) \rightarrow (Ni) + CH_4$
$H_2O + (Ni) \rightarrow H_2O(Ni)$	$H_2O(Ni) \rightarrow (Ni) + H_2O$
$CO_2 + (Ni) \rightarrow CO_2(Ni)$	$CO_2(Ni) \rightarrow (Ni) + CO_2$
$CO + (Ni) \rightarrow CO(Ni)$	$CO(Ni) \rightarrow (Ni) + CO$
$O(Ni) + H(Ni) \rightarrow OH(Ni) + (Ni)$	$OH(Ni) + (Ni) \rightarrow O(Ni) + H(Ni)$
$OH(Ni) + H(Ni) \rightarrow H_2O(Ni) + (Ni)$	$H_2O(Ni) + (Ni) \rightarrow OH(Ni) + H(Ni)$
$OH(Ni) + OH(Ni) \rightarrow O(Ni) + H_2O(Ni)$	$O(Ni) + H_2O(Ni) \rightarrow OH(Ni) + OH(Ni)$
$O(Ni) + C(Ni) \rightarrow CO(Ni) + (Ni)$	$CO(Ni) + (Ni) \rightarrow O(Ni) + C(Ni)$
$O(Ni) + CO(Ni) \rightarrow CO_2(Ni) + (Ni)$	$CO_2(Ni) + (Ni) \rightarrow O(Ni) + CO(Ni)$
$HCO(Ni) + (Ni) \rightarrow CO(Ni) + H(Ni)$	$CO(Ni) + H(Ni) \rightarrow HCO(Ni) + (Ni)$
$HCO(Ni) + (Ni) \rightarrow O(Ni) + CH(Ni)$	$O(Ni) + CH(Ni) \rightarrow HCO(Ni) + (Ni)$
$CH_4(Ni) + (Ni) \rightarrow CH_3(Ni) + H(Ni)$	$CH_3(Ni) + H(Ni) \rightarrow CH_4(Ni) + (Ni)$
$CH_3(Ni) + (Ni) \rightarrow CH_2(Ni) + H(Ni)$	$CH_2(Ni) + H(Ni) \rightarrow CH_3(Ni) + (Ni)$
$CH_2(Ni) + (Ni) \rightarrow CH(Ni) + H(Ni)$	$CH(Ni) + H(Ni) \rightarrow CH_2(Ni) + (Ni)$
$CH(Ni) + (Ni) \rightarrow C(Ni) + H(Ni)$	$C(Ni) + H(Ni) \rightarrow CH(Ni) + (Ni)$
$O(Ni) + CH_4(Ni) \rightarrow CH_3(Ni) + OH(Ni)$	$CH_3(Ni) + OH(Ni) \rightarrow O(Ni) + CH_4(Ni)$
$O(Ni) + CH_3(Ni) \rightarrow CH_2(Ni) + OH(Ni)$	$CH_2(Ni) + OH(Ni) \rightarrow O(Ni) + CH_3(Ni)$
$O(Ni) + CH_2(Ni) \rightarrow CH(Ni) + OH(Ni)$	$CH(Ni) + OH(Ni) \rightarrow O(Ni) + CH_2(Ni)$
$O(Ni) + CH(Ni) \rightarrow C(Ni) + OH(Ni)$	$C(Ni) + OH(Ni) \rightarrow O(Ni) + CH(Ni)$

Equations (9.1) and (9.2) are nonlinearly coupled through the reaction terms to equations (9.10), (9.11) and (9.12). As mentioned earlier in Section 9.2.1, the molar production rate for each species in the k-th reaction can be represented as

$$\dot{s}_{i,k} = (\nu''_{ik} - \nu'_{ik}) q_k, \tag{9.13}$$

where q_k is the rate-of-progress variable for the k-th reaction. If the charge-transfer rate is represented in a Butler-Volmer equation, see Section 9.2.3,

the rate-of-progress q_k of the electrochemical step k can be calculated as

$$q_k = \frac{i_{e,k}}{n_{e,k}F}. \tag{9.14}$$

In expression (9.14) $n_{e,k}$ represents the number of electrons transferred and $i_{e,k}$ the current density per unit length resulting from the k-th electrochemical reaction. The total charge-transfer rate per unit volume from all the electrochemical reactions can be expressed as

$$\dot{s}_{m,e} = \sum_{i=1}^{I} \lambda_{\mathrm{TPB}}^{V} i_{e,i}, \tag{9.15}$$

with $\lambda_{\mathrm{TPB}}^{V}$ the triple-phase-boundary length per unit volume. For detailed information we refer to Zhu & Kee (2008).

The boundary conditions needed to solve the system of equations (9.1), (9.2), (9.10), (9.11) and (9.12) are summarized in Figure 9.3.

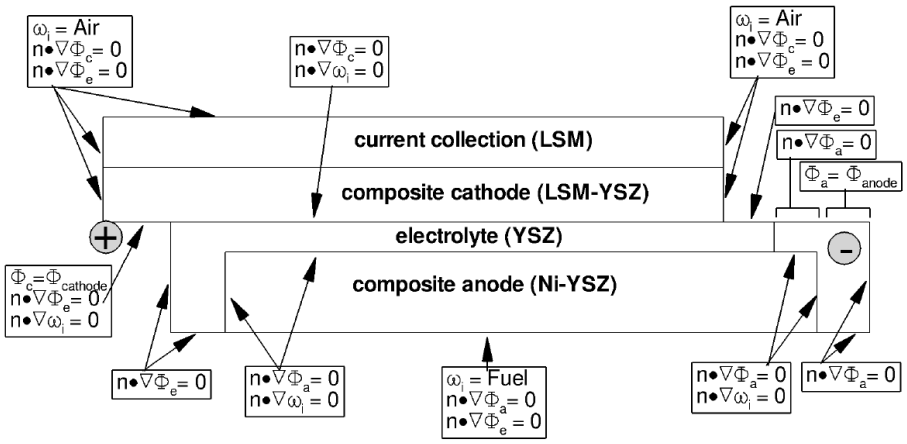


Figure 9.3: Boundary conditions

9.2.3 Charge Transfer Processes

Although the fuel stream on the anode may consist of hydrogen, hydrocarbons and carbon monoxide, in this study it is assumed that hydrogen is the only electrochemically active fuel species, see Zhu et al. (2005). The electrochemical H_2 oxidation within the anode and O_2 reduction within the cathode is written globally as

$$\mathrm{H_2}(g) + \mathrm{O}^{2-}(el) \rightleftharpoons \mathrm{H_2O}(g) + 2e^{-}(a), \text{ and} \tag{9.16}$$

$$1/2O_2(g) + 2e^-(c) \rightleftharpoons O^{2-}(el), \tag{9.17}$$

where $O^{2-}(el)$ are oxygen ion within the bulk electrolyte, $e^-(a)$ are the electrons within the anode and $e^-(c)$ are the electrons within the cathode. The local charge transfer rates for electrochemical H_2 oxidation within the anode and O_2 reduction within the cathode can be represented in Butler-Volmer form as

$$i_{e,\mathrm{BV}} = i_0 \left\{ \exp\left(\frac{\alpha_a F \eta_{\mathrm{act}}}{RT}\right) - \exp\left(-\frac{\alpha_c F \eta_{\mathrm{act}}}{RT}\right) \right\}, \tag{9.18}$$

where α_a and α_c are the anodic and cathodic symmetric factors, respectively, and η_{act} the activation overpotentials.

The local electric-potential difference within the anode E_a and the local electric-potential difference within the cathode E_c are defined as

$$E_a = \Phi_a - \Phi_{a,e}, \quad E_c = \Phi_c - \Phi_{c,e}. \tag{9.19}$$

The activation overpotential for the anode η_{act} is then defined as

$$\eta_{\mathrm{act},a} = E_a - E_a^{\mathrm{eq}}, \tag{9.20}$$

where E_a^{eq} is the local equilibrium electric-potential difference in the anode. Analogously, the activation overpotential for the cathode η_{act} is defined as

$$\eta_{\mathrm{act},c} = E_c - E_c^{\mathrm{eq}}, \tag{9.21}$$

with E_c^{eq} is the local equilibrium electric-potential difference in the cathode. Under the assumption that throughout the MEA structure the bulk concentration O^{2-} is spatially uniform, the local equilibrium electric-potential differences in the anode and cathode, respectively, can be evaluated as

$$E_a^{\mathrm{eq}} = \frac{\mu^\circ_{H_2O} - \mu^\circ_{H_2}}{2F} + \frac{RT}{2F} \ln\left(\frac{P_{H_2O,a}}{P_{H_2,a}}\right), \text{ and,} \tag{9.22}$$

$$E_c^{\mathrm{eq}} = \frac{\mu^\circ_{O_2}}{4F} + \frac{RT}{2F} \ln(P_{O_2,c}), \tag{9.23}$$

where μ_i° is the standard-state chemical potential of species i, and P_i is the partial pressure of species i measured in atmospheres.

The exchange current densities i_0 in the Butler-Volmer equation (9.18) represent the temperature and species dependencies for the charge-transfer reactions. Zhu et al. (2005) derived the following expression for the exchange current density of H_2 oxidation:

$$i_{0,H_2} = i^*_{H_2} \frac{\left(P_{H_2}/P^*_{H_2}\right)^{(\alpha_a - 1)/2} \left(P_{H_2O}\right)^{\alpha_a/2}}{1 + \left(P_{H_2}/P^*_{H_2}\right)^{1/2}}. \tag{9.24}$$

The parameter $P^*_{H_2}$ depends upon hydrogen adsorption and desorption rates. The temperature dependence can be expressed as

$$i^*_{H_2} = i^*_{\text{ref},H_2} \exp\left(-\frac{E_{a,H_2}}{R}\left[\frac{1}{T} - \frac{1}{T_{\text{ref}}}\right]\right), \tag{9.25}$$

where E_{a,H_2} is an activation energy, and the parameter i^*_{ref,H_2} is assigned empirically to fit measured polarization data at the reference temperature T_{ref}. Similarly, the exchange current density for oxygen reduction at the cathode is

$$i_{0,O_2} = i^*_{O_2} \frac{\left(P_{O_2}/P^*_{O_2}\right)^{\alpha_a/2}}{1 + \left(P_{O_2}/P^*_{O_2}\right)^{1/2}}, \tag{9.26}$$

where

$$i^*_{O_2} = i^*_{\text{ref},O_2} \exp\left(-\frac{E_{a,O_2}}{R}\left[\frac{1}{T} - \frac{1}{T_{\text{ref}}}\right]\right). \tag{9.27}$$

In expression (9.26) the parameter $P^*_{O_2}$ depends upon the adsorption and desorption rates. In expression (9.27) E_{a,O_2} is an activation energy and i^*_{ref,O_2} is assigned empirically. Again, the partial pressures in expressions (9.24) and (9.26) are measured in atmospheres.

9.3 Numerical Methods

Two numerical approaches are used to compute the numerical solution of the coupled system of the nonlinear partial differential equations (9.1), (9.2), (9.10), (9.11) and (9.12) formulated in the previous section. Since the interest is only in the steady state solution, Kee et al. (2008) computed this steady state solution by means of solving subsequently and iteratively the model for electric-potential distributions, the species transport, and the chemical and electrochemical reactions. Since the reaction terms do not involve spatial operators, the chemistry and electrochemistry are solved per grid point throughout the electrode structures by a time-relaxation algorithm.

Another approach, used in this thesis, is to solve the system of partial differential equations and algebraic constraints using the Euler Backward solver as described in Chapter 5, 6 and 7. Remark that some adaptations have to be made for the incorperation of the PDEs describing the electric-potential distributions (see equations (9.10), (9.11) and (9.12)) and the linear constraints to compute the diffusive fluxes (see equation (9.4)). The latter are simply added in the right-hand side of the semi-discrete system,i.e.,

$$\begin{bmatrix} w'(t) \\ 0 \end{bmatrix} = \begin{bmatrix} F_1(t, w(t)) \\ F_2(t, w(t)) \end{bmatrix}, \tag{9.28}$$

where $F_1(t, w)$ represents the spatially discretized electric-potential and species equations in the cathode and anode, and $F_2(t, w)$ represents the spatially discretized linear constraints for the diffusion fluxes in the cathode and anode.

The calculation of thermodynamic and transport properties for each species, and the evaluation of the chemical reaction rates in the detailed reaction mechanisms, has been done with the C++ Cantera interface, see Goodwin (2008). However, the solution procedures and remaining other parts of the code are written in Fortran 90. An interface has been created such that the Cantera C++ routines can be called from the Fortran subroutines.

The resulting nonlinear algebraic equations are solved with the Globalized Inexact Newton method described in Chapter 6. It appears that for these simulations Projected Newton methods are not necessary. In the case one wants to apply the Projected Newton method, the projection only has to be applied to the species concentrations, since electric potentials can be negative.

The resulting linear systems are solved with a preconditioned Bi-CGSTAB method, where a block ILU preconditioner has been used. This preconditioner will be discussed in more detail. The block-nonzero structure of the fully coupled system is illustrated in Figure 9.4. The diagonal block EP corresponds to the nonzero structure of the discretized Laplace operator, whereas the nonzero blocks SC, SA, DC and DA have the nonzero structure of the Jacobian-matrix as in the species equations appearing in Chemical Vapor Deposition simulations, see Chapter 7.

The approach to compute a block incomplete factorization of the matrix with a nonzero structure as in Figure 9.4 is as follows. Here, we recycle as much as possible of the block ILU preconditioner described in Chapter 7. The block incomplete factorization as implemented in our code uses the blocks as illustrated in Figure 9.4. To obtain this block incomplete factorization the diagonal blocks EP, SC, SA, DC and DA need to be 'inverted'. Instead of computing the 'exact' inverse (as was done in Chapter 7) we use the block ILU factorization of Section 7.4.1 as an approximation of the inverse. For the block EP corresponding to the electric-potential equations a regular incomplete factorization is used. The multiplications of the 'approximated' inverse diagonal elements by a matrix D is computationally cheap to compute, because D is a diagonal matrix. Exactly the same situation is observed in the construction of the block ILU factorization preconditioner described in Chapter 7.

Using this solver configuration computing times to calculate the steady state solution of these problems take about $4\frac{1}{2}$ - $5\frac{1}{2}$ hours for moderately fine meshes. These meshes contain of about 3000 - 5000 finite volume cells and produce accurate results. However, computing times might be reduced by incorporating information that is known beforehand. It is expected, and

this can also be seen in the numerical results in the next section, that the solution in the direction normal to the cathode-electrode interface behaves virtually one-dimensionally along a substantial part of this interface. In our current solver this information is not used in the algorithm to solve the nonlinear and linear systems.

A first approach to use this information is the following. Over most of the computational domain where the solution behaves virtually one-dimensional, a preconditioner could be build based upon a line solver (in the direction normal to the cathode-electrode interface). Using the correct ordering of unknowns, this implies that that along this section the preconditioner is locally exact.

Another approach to decrease the total computational costs would be on the level of Newton iteration. It might be very well possible to improve the initial guess by using the information that a very large part of the solution behaves as the solution of a one-dimensional 'shadow' problem. The resulting number of Newton iterations decreases when the initial guess is 'closer' to the solution. In that case the total computational costs will decrease considerably.

9.4 Numerical Results

The results reported in this section are based on an LSM-YSZ|YSZ|Ni-YSZ SIS unit cell. The physical dimensions are shown in Figure 9.2 and model parameters are shown in Table 9.2. The porous cathode consists of two layers: 30 μm LSM-YSZ functional layer near the dense electrolyte layer to promote charge-transfer chemistry and a 30 μm LSM layer to increase the lateral electric conductivity. The dense YSZ electrolyte is 20 μm thick and the Ni-YSZ anode is 50 μm thick. Overall the unit cell is 2600 μm wide.

As an example, consider the fuel mixture at the interface between the porous support and the anode to consist of 66.1 mole% H_2, 21.8 mole% CO, 11.6 mole% CH_4, 0.3 mole% H_2O, and 0.2 mole% CO_2. This mixture, chosen somewhat arbitrarily, is the equilibrium product of an initial mixture of 60 mole% CH_4 and 40 mole% H_2O at 800 °C and atmospheric pressure. The cathode side of the unit cell is exposed to air. The cell is operating at 800 °C, atmospheric pressure and the difference between the cathode interconnect potential (left side of unit cell) and the anode interconnect (right side of the unit cell) is 0.5 V.

Figure 9.5 is a composite image that illustrates many aspects of the solution. The color contours represent electric potentials of the electrolyte phase Φ_e. The white lines superimposed on the color contours are "path lines" for electron flux. These lines originate on nine equal intervals across the interconnect face. These path lines follow the electron paths from the interconnect face, but they do not represent the magnitudes of the electron

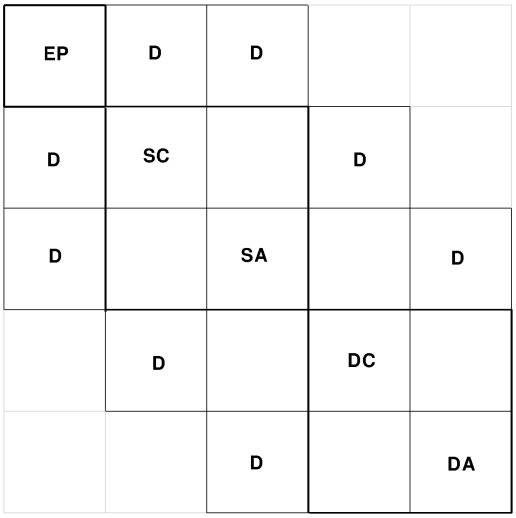


Figure 9.4: Nonzero structure of the Jacobian matrix. The block EP corresponds to the partial derivatives of the electric-potential equations, the block SC corresponds to the partial derivatives of the species equations in the cathode, the block SA corresponds to the partial derivatives of the species equations in the anode, the block DC corresponds to the partial derivatives of the algebraic constraints for the diffusion fluxes in the cathode and the block DA corresponds to the partial derivatives of the algebraic constraints for the diffusion fluxes in the anode. The blocks D are diagonal blocks representing partial derivatives of the coupling between various unknowns.

Table 9.2: Parameters for modeling the MEA unit cell

Parameter	Value	Unit
Anode		
Thickness H_a	50	μm
Porosity ϕ_g	0.35	
Ni volume fraction ϕ_{Ni}	0.23	
YSZ volume fraction ϕ_{YSZ}	0.42	
Tortuosity τ_g	4.50	
Ni particle radius r_{Ni}	0.50	μm
YSZ particle radius r_{YSZ}	0.50	μm
Effective Ni electric conductivity at 800 °C σ_a^e	1134.42	$\mathrm{S \cdot cm^{-1}}$
Effective YSZ electric conductivity at 800 °C σ_e^e	0.01156	$\mathrm{S \cdot cm^{-1}}$
Exchange current factor $i^*_{\mathrm{ref,H_2}}$	$4.80 \cdot 10^3$	$\mathrm{A \cdot cm^{-3}}$
Activation energy $E_{\mathrm{a,H_2}}$	120.0	$\mathrm{kJ \cdot mol^{-1}}$
Reference temperature T_{ref}	800.0	°C
Anodic symmetric factor	1.5	
Cathodic symmetric factor	0.5	
Cathode		
Thickness H_c	60	μm
Porosity ϕ_g	0.35	
LSM volume fraction ϕ_{LSM}	0.31	
YSZ volume fraction ϕ_{YSZ}	0.34	
Tortuosity τ_g	4.00	
LSM particle radius r_{LSM}	0.625	μm
YSZ particle radius r_{YSZ}	0.625	μm
Effective LSM electric conductivity at 800 °C σ_c^e	46.03	$\mathrm{S \cdot cm^{-1}}$
Effective YSZ electric conductivity at 800 °C σ_e^e	$7.47 \cdot 10^{-3}$	$\mathrm{S \cdot cm^{-1}}$
Exchange current factor $i^*_{\mathrm{ref,O_2}}$	130.0	$\mathrm{A \cdot cm^{-3}}$
Activation energy $E_{\mathrm{a,H_2}}$	120.0	$\mathrm{kJ \cdot mol^{-1}}$
Reference temperature T_{ref}	800.0	°C
Anodic symmetric factor	0.75	
Cathodic symmetric factor	0.25	
Electrolyte		
Thickness	20	μm
Effective YSZ electric conductivity at 800 °C σ_e^e	0.04226	$\mathrm{S \cdot cm^{-1}}$

flux.

There are several interesting observations that can be gathered from the electron path lines. Assuming that the dense electrolyte is a pure ion conductor, the electron path lines cannot penetrate into the dense electrolyte layer. Rather, charge crosses the dense electrolyte in the form of O^{2-}-flux. As discussed subsequently, in the cathode charge is transferred from electrons in the electrode phase (LSM) to the oxygen ion in the electrolyte phase (YSZ). Then, within the anode charge is transferred from the electrolyte phase (YSZ) to the electrode phase (Ni). The concentration of electron path lines in the LSM layer is clear evidence of its lower electrical resistance.

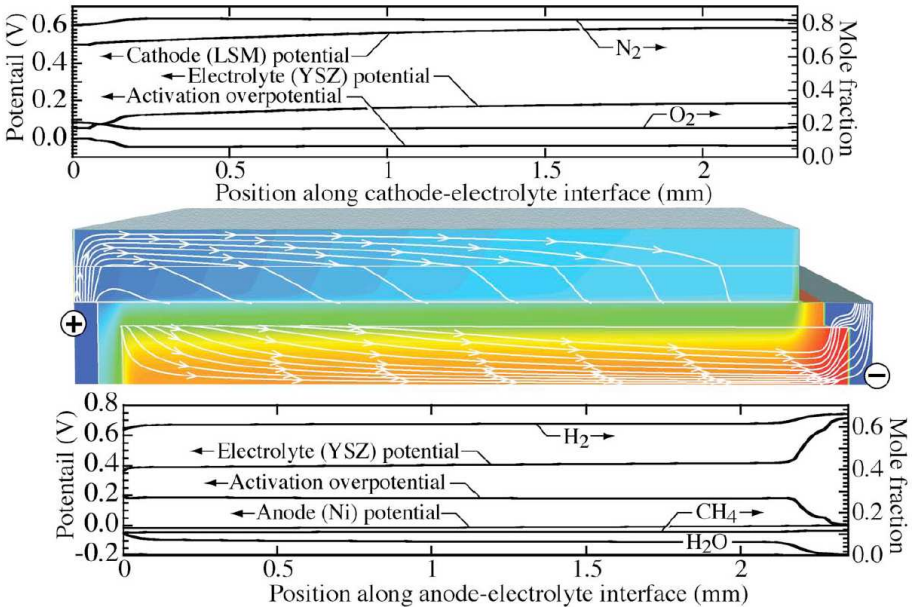


Figure 9.5: Steady state solution for a unit cell and nominal operating conditions

The interconnect structures in the nominal cell are only 80 μm wide. Thus, all the electrical current produced by the roughly 2200 μm width of the active cell must be channeled into the relatively small interconnect. This current concentration can potentially cause local heating, or other possibly deleterious effects on the cell materials. One expects that careful design of the interconnect regions is important to cell reliability and lifetime.

The upper graph in Figure 9.5 shows profiles of several variables along the interface between the cathode and the dense electrolyte. The cathode layer is LSM-YSZ, and the model assumes that the LSM particles (cathode phase) are pure electron conductors and the YSZ particles (electrolyte phase) are pure oxygen-ion conductors. The electric potential of the cathode phase is always greater than the electrolyte phase. This relationship is the result of

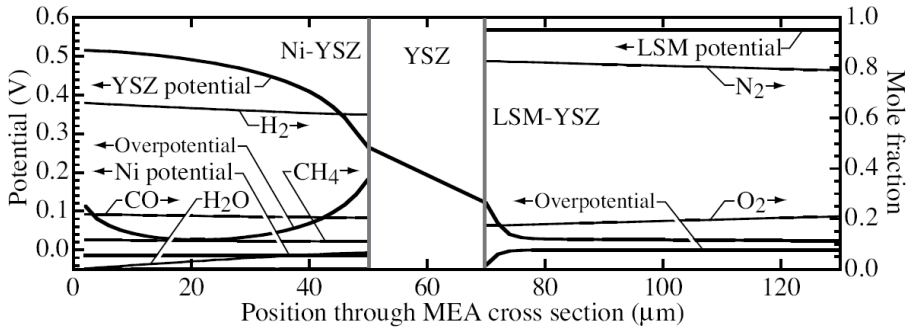


Figure 9.6: Steady state solution profiles across the MEA unit cell midway between the anode and cathode interconnectors

local electrical double layers between the electrode and electrolyte particles. The electric potentials of both the cathode phase Φ_c and electrolyte phase Φ_e increase from left to right. Within the composite cathode structure, both negatively charged ions and electrons are being transported from left to right. Such charged-species transport is driven in the direction of the negative electrochemical-potential gradient. This means that negatively charged species generally must be transported up the electric-potential gradient (i.e., from regions of relatively more negative electric potential toward regions of relatively more positive electric potential). The activation overpotential in the cathode structure is negative. Based on the Butler-Volmer equation, a negative overpotential means that the net electrochemical charge-transfer is "cathodic" (meaning that electrons are consumed).

Figure 9.5 shows mole-fraction profiles for N_2 and O_2. As expected, because of oxygen consumption the N_2 mole fraction increases slightly while the O_2 mole fraction decreases slightly.

The lower graph in Figure 9.5 shows solution profiles within the cermet anode structure along the interface between the dense electrolyte and the anode. Similar to the cathode structure, the electric-potential profiles increase lightly from left to right. Again, this is because the negatively charged ions and electrons are transported up the electric potential gradients. However, because the Ni electrical conductivity is so high, the electric-potential gradient within the anode phase is small. The lateral electric-potential gradient within the electrolyte phase (YSZ) is also small. This is because oxygen-ion flux is dominantly in the direction normal to the dense electrolyte. Upon transferring charge from the oxygen ion in the YSZ phase to a electron in the Ni phase, the electron flux toward the anode interconnector provides a low-resistance path.

Gas-phase species profiles are also shown on Figure 9.5. The gradients are generally small. The H_2 mole fraction is roughly 61% at the dense

electrolyte interface, while the unreacted level at the interface with the porous support is roughly 66 %. The decrease near the dense electrolyte is expected as a result of consumption via electrochemical charge-transfer reactions. The H_2 electrochemical consumption rate decreases slightly from left to right, causing a slight increase in H_2 mole fraction profile. The consumption rate is highest near the cathode interconnect as a result of slightly higher activation overpotentials in upper left-hand corner of the anode structure.

The H_2O mole fraction is roughly 7 % at the dense electrolyte interface, while at the porous-support interface it is only 0.3 %. Because H_2O is the product of electrochemical charge-transfer reactions, the increased levels are expected. The H_2 O decreases slightly from left to right as a result of decreased charge transfer rates toward the anode interconnect, and some consumption by steam-reforming of CH_4. The CH_4 levels are around 11 %, which is slightly lower than the 11.6 % in the feed. The CH_4 is reformed with electrochemically produced steam to produce some H_2 and CO.

Over much of the MEA unit-cell many of the solution components are essentially one-dimensional. For example, species profiles depend primarily upon the direction normal to the dense electrolyte. The electric potentials have greater lateral variations than the species profiles, but still vary primarily in the direction normal to the dense electrolyte. In the vicinity of the interconnectors there is significant two-dimensional behavior as the current is channeled into the connections between unit cells.

Figure 9.6 shows solution profiles through the MEA and normal to the dense electrolyte at a lateral position in the middle of the unit cell (specifically, 1200 μm from the left edge of the anode structure). In this figure the anode is on the left and cathode is on the right. The anode phase (Ni) electric potential is essentially uniform at Φ_a = - 0.014 V and the cathode phase (LSM) electric potential is essentially uniform at Φ_a = 0.5 V. The anode interconnector is at Φ_a = 0 V and the cathode interconnector is at Φ_a = 0.5 V. As discussed in connection with Figure 9.5, the deviations from the interconnector values are due to the potential gradients needed to support the lateral current flux.

The electrolyte phase (YSZ) electric potential varies considerably through the thickness of the MEA. This is because of the relatively low ion conductivity and the charge-transfer process between the electrode and electrolyte phases. On the anode side, the electrode (Ni) electric potential is always lower than the electrolyte (YSZ) potential, and the overpotential is always positive. This relationship is established because of the electrical double-layer between the electrode and electrolyte phases. The electric-potential difference $\Delta\Phi = \Phi_{Ni} - \Phi_{YSZ}$ is always negative within the anode structure, but becomes less negative near the dense electrolyte interface. Consequently, the anodic charge-transfer rate is higher near the dense electrolyte interface. As the magnitude of $\Delta\Phi$ decreases (i.e., the YSZ particles become

less positive) the rate of the charge-transfer reaction (9.16) increases. This can be understood qualitatively in the sense that as the strength of the double layer decreases, it becomes easier to transfer negative charge into the negative electrode.

Within the dense electrolyte the electrolyte-phase electric potential is linear. The oxygen-ion flux is proportional to the electric-potential gradient and the ion conductivity. Assuming that YSZ is a pure ionic conductor, there is no need to consider an electrode-phase electric potential within the dense electrolyte.

Within the LSM-YSZ cathode structure, the electric potential of the electrode (LSM) phase is positive relative to the electric potential of the electrolyte (YSZ) phase. As in the anode structure, there is significant spatial variation in the YSZ electric-potential profile. The electric-potential difference $\Delta\Phi = \Phi_{LSM} - \Phi_{YSZ}$ is always positive within the cathode structure, but becomes less positive near the dense electrolyte interface. Consequently, the cathodic charge-transfer rate is higher near the dense electrolyte interface. As the magnitude of $\Delta\Phi$ decreases (i.e., the YSZ particles become more positive) the rate of the charge-transfer reaction (9.17) increases. This can be understood qualitatively in the sense that as the strength of the double layer decreases, it becomes easier to transfer negative charge from the electrode into the relatively negative electrolyte. Within the cathode structure, the activation overpotential is always negative. In the context of the Butler-Volmer equation, a negative overpotential drives the charge-transfer reaction in the cathodic direction (i.e., consuming electrons).

The gas-phase mole-fraction profiles shown in Figure 9.6 have expected behaviors. In the anode pore spaces, the H_2 fuel decreases toward the dense electrolyte as a result of electrochemical consumption. The electrochemical reaction product H_2O decreases toward the interface with the support structure. CO and CH_4 levels are nearly uniform, but both decreases slightly toward the dense electrolyte interface. On the cathode side, the O_2 mole reaction decreases toward the dense-electrolyte interface where it is consumed by electrochemical reduction of O_2 to form O^{2-}. Consequently, the N_2 mole fraction must increase near dense electrolyte interface.

9.5 Summary, Conclusions and Future Challenges

The model presented in this chapter describes the transport, thermal catalytic chemistry and electrochemistry within a unit cell of an SIS-SOFC. The electrochemical charge-transfer occurs throughout the electrode structures, whereby the local charge-transfer rate depends on the local gas mixture composition and electric-potential differences between electrode and electrolyte phases. The reforming chemistry in the anode is modeled by a elementary reaction mechanism. Charge-transfer chemistry was represented

in Butler-Volmer form.

Important characteristics of the performance of an SIS-SOFC has been shown by means of an illustrative example problem. Although the computational model is two-dimensional, the computed solutions behave one-dimensionally over the largest part of the computational domain. In the vicinity of the interconnectors, however, there is a strong two-dimensional behavior as the electronic current is channeled into the interconnection between cells. For the design of SIS-SOFC stacks, these regions of high current densities are critical.

The numerical software used is briefly described in Section 9.3. As remarked earlier, it is believed that the present solver, and in particular the linear solvers, can be further optimized. Specifically, the behavior of a substantial part of the solution is virtually one-dimensional. Probably, great computational advantages can be achieved when this type of information is included in the numerical solvers. Two potential strategies are formulated in Section 9.3.

The computing times for the simulation results mentioned in Section 9.3 are hard to compare with those for the numerical experiments performed in Kee et al. (2008). The implementation of our current solver is not optimal. In particular, the communication between the C++ routines and the Fortran 90 routines is sometimes time consuming. However, it is believed that the fully coupled approach, presented in this thesis, gives faster convergence towards the solution than the approach followed in Kee et al. (2008). Where we measured computing times of several hours, the computations of Kee et al. (2008) took a couple of hours more. No significant differences in the solutions computed with both solvers are found.

The solution strategy used in Kee et al. (2008) has been proven to be very robust; the solver has been used in numerous other SOFC computations found in for instance Zhu et al. (2005) and Zhu & Kee (2008). The solver presented in this chapter has only been applied to the example considered in Section 9.4. It should be remarked that the convergence speed of the present solver depends on the initial value. Changing the initial solution gives either slower, or faster convergence towards the steady state. From that point of view the present solver is not as robust as the solver of Kee et al. (2008).

The final remark concerns the need for a general stationary solver for this type of problems. Since the physical models are constantly under development, it would be useful when a general solver exists for this type of problems. The two important properties that this solver needs to have are:

- it should be capable to deal with extremely stiff systems of nonlinear equations,
- the iterative linear solver should converge reasonably fast towards a

linear solution,which means that effective preconditioning is needed.

Generally speaking, the construction of effective preconditioners is crucial for this type of computations.

CHAPTER 10

Conclusions

The aim of this study was to develop robust and efficient numerical methods for the instationary simulation of laminar reacting gas flows. Typically, these flows are found in production processes such as Chemical Vapor Deposition, or in laminar combustion. Another example of non-turbulent reacting gas flows is a Solide Oxide Fuel Cell, in which the reactants are flowing and the reactions are taking place in a porous medium. For the design of time accurate simulation software for these type of problems, it is important to understand the numerical difficulties one might encounter.

Usually, finding the solution of the flow problem is a rather trivial task compared to the solution of the system of the advection-diffusion-reaction equations for a large number of reactants and intermediate species. These equations are stiffly coupled through the reaction terms, which typically include dozens of finite rate elementary reaction steps with largely varying rate constants. The solution of such systems of stiff equations is difficult, for both stationary and instationary simulations.

The approach followed in this study consists of two parts. The first step is to study discretization techniques in space and time. In Section 10.1 concluding remarks on discretization techniques are discussed. In Section 10.2 we review the proposed solution techniques and we present the concluding remarks concerning this topic. Subsequently, in Section 10.3 the evolution of the computing times of a representative test problem is discussed. Finally, in Section 10.4 recommendations for future research are formulated.

10.1 Concluding Remarks on Discretization Techniques

For the species concentrations it is important that their non-negativity is conserved. Since the applications studied in this thesis are characterized by relatively low Reynolds number flows, and consequently low cell-Péclet numbers, positivity of the species concentration equations is easily conserved in spatial discretization. For such low cell-Péclet numbers, even the second order accurate central differencing scheme is positivity preserving.

The story is completely different for time integration of the semi-discrete ODEs, which are obtained after spatial discretization. Lots of research has been done by the ODE community, which resulted in a huge amount of literature on stable integration of stiff ODEs. However, the search for a stiffly stable time integration method that also preserves positivity of the solution can be restricted to first order accurate time integration methods. Higher order time integration methods put a severe criterium on the time step size to ensure non-negativity of the species concentrations. For reacting flow problems this criterium is much more restrictive towards the time step size than stability.

The Euler Backward time integration is the only known method which is proven to be unconditionally positive. This, and the above motivates the design of a nonstationary solver using the Euler Backward time integration method. Solving the species transport equations fully coupled and fully implicitly involves the solution of huge systems of nonlinear algebraic equations. In the next section conclusions on the solution techniques proposed in this thesis are formulated.

10.2 Concluding Remarks on Solution Techniques

Within the applied mathematics communities the traditional approach to solve systems of nonlinear algebraic equations is Newton's method. Combined with a direct method, i.e., computing the exact Newton step, Newton's method shows excellent performance with respect to positivity of the solution. Additionally, per nonlinear iteration a few line-search iterations are required to obtain Newton convergence due to the strong nonlinear reaction terms present.

The computational effort to solve the interior linear algebra problem in Newton's method increases cubically in the number of mesh points and the number of species in the gas mixture. The fact that the linear systems are large and sparse motivates the introduction of preconditioned Krylov Subspace methods to compute the solution of the interior system of linear equations. Again, one might encounter two problems. First, by allowing iterative linear solution methods the positivity of all species concentrations is no longer ensured on the nonlinear solution level. Secondly, the stiff

reaction terms give rise to huge condition numbers. Applying iterative linear solvers to ill-conditioned linear systems usually results into no or very poor convergence.

In order to maintain the unconditional positivity of Euler Backward time integration, combined with Inexact Newton solvers, a projected version of such Newton methods is applied. If this projected Newton method converges, then the obtained solution on the new time level is guaranteed to be positive. It has been illustrated that this approach gives more accurate results with respect to mass conservation than alternative methods such as clipping.

On the linear algebra level of these solvers, the influence of various preconditioners on the linear convergence has been studied. Effective preconditioners cluster the eigenvalues of the linear system such that, hopefully, the condition number drops significantly as well. Consequently, fast convergence of the preconditioned Krylov method is achieved. By means of numerical experiments it has been illustrated that incomplete factorization type preconditioners do this much more effectively than block diagonal type preconditioners. The block incomplete factorization, where a block corresponds to all species per grid point, has been found to be the most effective preconditioner.

Combining these iterative solution techniques resulted in a considerable reduction of computional costs, compared to nonstationary solvers equipped with direct linear solvers, and the steady state solvers developed bij Kleijn and co-workers. Further, these numerical techniques enabled us to perform three-dimensional time accurate transient simulations from inflow conditions until steady state on a $70 \times 70 \times 70$ mesh in reasonable time. The chemistry model in this case consists of 17 species that satisfy a reaction mechanism of 26 gas phase reaction and 14 surface reactions.

10.3 Evolution of Computational Costs

It is particularly interesting to see how the total computational effort, measured in wall clock time, evolved. Recall, that the instationary computations are running from inflow conditions until steady state. For two-dimensional simulations on a 35×32 spatial mesh it evolved as:

- 25,000 CPU sec with steady state solver of Kleijn,
- 20,000 CPU sec with direct linear solver,
- 6,500 CPU sec with direct linear solver and efficient ordering,
- 3,000 CPU sec with preconditioned iterative linear solver,
- 300 CPU sec with preconditioned iterative linear solver and an effective ordering of the unknowns.

With our current solvers the CPU time for the time accurate transient solution on a 70×82 grid has been reduced to approximately one hour, compared to approximately a few days with the software of Kleijn. The effectivity of these numerical techniques in the optimal configuration, that is Euler Backward time integration, Projected Newton methods and the block ILU preconditioner, is illustrated in the computation of the three-dimensional nonstationary solutions:

- about 5 hours computation time on a $35 \times 35 \times 35$ grid, and
- about $4^1/_2$ days on a $70 \times 70 \times 70$ mesh.

In all cases the chemistry model consisted of 17 species, 26 gas phase and 14 surface reactions.

10.4 Future Research

The primary focus of the research presented in this dissertation has been on efficient and robust solution techniques for time accurate laminar reacting gas flow solvers. A first start has been made on the incorporation of iterative linear solvers in nonstationary solvers for these type of problems. The investigated preconditioning techniques are all of the tradional-type. Further improvements of these techniques could include multigrid based preconditioners and/or second level preconditioning, such as deflation.

The majority of the detailed CVD chemistry models contain besides gas phase reactions and species, also surface species and reactions. Then, partial differential equations describing the gas phase chemistry have to solved throughout the whole computational domain and coupled to a set of ordinary differential equations considering the surface reactions. Extending the current software with more sophisticated surface reaction models is of great importance, because then the majority of the published CVD chemistry models can be used.

From a practical point of view, the existing code has to be parallelized, in order to keep the computing times reasonable when using larger chemistry models. The same is probably true for memory requirements. It is believed that parallelization of this code is rather straightforward, except for the preconditioned Krylov solver.

APPENDIX A

Positive Krylov Methods

Related to the question what conditions spatial discretization and time integration methods need to fulfill in order to maintain all components of the solution positive, the following problem statement is put forward. We start by recalling the definition of an M-matrix, which has been taken from Saad (2003). Berman & Plemmons (1994) wrote a text on the subject of M-matrices in the mathematical sciences.

Definition A.1. *An $n \times n$ matrix A is said to be an M-matrix if it satisfies the following four properties:*

1. $(A)_{ii} > 0$ *for* $i = 1, \ldots, n$,
2. $(A)_{ij} \leq 0$ *for* $i \neq j$, $i, j = 1, \ldots, n$,
3. *A is nonsingular, and,*
4. $A^{-1} \geq 0$.

Suppose that A is an M-matrix and that $b \geq 0$ (component-wise). Then, the solution x of the linear system

$$Ax = b, \tag{A.1}$$

is component-wise positive as well.

When solving equation (A.1) by means of a preconditioned Krylov Subspace method, then it is a priori not known whether the approximated solution is positive.

In this chapter we investigate the case that A is symmetric positive definite and satisfies the M-matrix property, and the (preconditioned) linear system is solved by means of the Conjugate Gradient method. Details of the Conjugate Gradient method are found in standard texts on this subject, like that of Saad (2003).

A.1 Does the Conjugate Gradient Method Return Positive Approximations?

Assume that A is a symmetric positive definite matrix and that $b \geq 0$ (component-wise). Then, it is known that

$$x = A^{-1}b, \tag{A.2}$$

is positive as well. When approximating the solution of (A.1) by means of the Conjugate Gradient method, then each approximated solution that satisfies an arbitrary accuracy criterium (or stop criterium), should be positive. In other words, each Conjugate Gradient iterate should be positive.

If the initial iterate is equal to the zero-vector, then it is an easy exercise to show that the first Conjugate Gradient iterate is always positive. Consider the Conjugate Gradient Algorithm as stated in Algorithm 7, see Saad (2003).

Algorithm 7: Conjugate Gradient

1: Compute $r^0 = b - Ax^0$ and $p^0 = r^0$,
2: **for** $j = 1, 2, \ldots,$ until convergence **do**
3: $\quad \alpha_j = (r^j, r^j)/(Ap^j, p^j)$,
4: $\quad x^{j+1} = x^j + \alpha_j p^j$,
5: $\quad r^{j+1} = r^j - \alpha_j A p^j$,
6: $\quad \beta_j = (r^{j+1}, r^{j+1})/(r^j, r^j)$,
7: $\quad p^{j+1} = r^{j+1} + \beta_j p^j$.
8: **end for**

Hence, if x^0 equals the zero-vector, then $r^0 = p^0 = b$. Since A is positive definite we have that $(Ap, p) > 0$ for all nonzero p. It follows that $\alpha_1 \geq 0$, and thus that $x^1 \geq 0$.

For the second Conjugate Gradient it is not clear whether it remains positive, because the updated residual r^2 is not necessarily positive (see line 5 of Algorithm 7). Consequently, the same holds for all subsequent iterates.

On the other hand, we could not find a counterexample for which this is true. Numerical experiments with symmetric positive definite matrices A revealed that for (arbitrary) positive right-hand sides b no negative entries in the approximate solution vector x are observed. To the author's knowledge, this question has not been answered.

A.2 What about Preconditioning?

Logically, the second question to be put forward is which preconditioners maintain these positivity properties. Since the first question has not been answered (yet), there is not much to mention on this topic.

When using the preconditioned Conjugate Gradient algorithm, then the coefficient α_1 is computed as

$$\alpha_1 = (r^0, z^0)/(Ap^0, p^0), \tag{A.3}$$

where $z^0 = P^{-1}r^0$ and P the preconditioner. Sufficient conditions for the first iterate to be positive is that α_1 is positive. This is true if and only if z^0 is positive. Under the assumption that the initial iterate x^0 is the zero-vector, the initial residual is equal to $r^0 = b$. A sufficient condition for the preconditioner P to let z^0 be positive is that P is an M-matrix.

Since we are not able to proof that subsequent iterates remain positive, we are also not able to derive conditions on the preconditioner to remain positive solutions for the preconditioned Conjugate Gradient method.

However, numerical experiments show that for incomplete factorization preconditioners and diagonal preconditioners all Conjugate Gradient iterates are positive. Therefore, it is conjectured that preconditioners that have the M-matrix property return positive preconditioned Conjugate Gradient iterates.

A.3 Other Krylov Subspace Methods

So far, only Krylov methods are discussed for symmetric positive matrices. For non-symmetric matrices there are generally two Krylov methods available: the family of Bi-CGSTAB methods and the family of GMRES methods. For both families of methods it is beforehand not even clear whether the first iterate is positive, if it is assumed that A is an M-matrix and $b \geq$ component-wise.

To the author's knowledge, this is still an unanswered question. However, since we are not able to proof identical properties for the Conjugate Gradient method, it is not expected that this is easily done for Krylov methods for general matrices.

Curriculum Vitae

- 2004-2008: PhD student in Numerical Analysis Group, Department of Applied Mathematical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands, financially supported by the Delft Centre for Computational Science and Engineering, subject: *Efficient numerical methods for the instationary solution of laminar reacting gas flow problems*, advisors: prof.dr.ir. C. Vuik and prof.dr.ir. C.R. Kleijn
- 2007: Research Associate, Division of Engineering, Colorado School of Mines, Colorado, United States of America, supervisor: prof.dr. R.J. Kee
- 2006-2008: Representative of the Numerical Analysis Group of Delft University of Technology in the J.M. Burgerscentrum PhD contact group.
- 2007: Organization of the J.M. Burgerscentrum PhD contact group outing
- 2007: Organization of PhDays 2007
- 2006: Organization of PhDays 2006
- 1999-2004: Delft University of Technology, Delft, MSc degree in Applied Mathematics
- 1992-1999: Caland Lyceum, Rotterdam, VWO
- Born on May 26, 1980 in Rotterdam, The Netherlands

List of Publications

Refereed Journal Papers

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *On Projected Newton-Krylov solvers for instationary laminar reacting gas flows*, submitted, (2008)

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Comparison of ODE Methods for Laminar Reacting Gas Flow Simulations*, Numerical Methods for Partial Differential Equations 24, pp. 1037-1054, (2008)

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Numerical Methods for Reacting Gas Flow Simulations*, International Journal for Multiscale Computational Engineering 5, issue 1, pp. 1-10, (2007)

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Transient Chemical Vapor Deposition Simulations*, Surface Coatings and Technology 201, pp. 8859-8862, (2007)

Contribution to Books

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *On Numerical Issues in Time Accurate Laminar Reacting Gas Flow Solvers*, in: B. Koren and C. Vuik (eds.) Advanced Computational Methods in Science and Engineering. Springer Series Lecture Notes in Computational Science and Engineering. (to appear).

Conference Proceedings

- R.J. Kee, S. van Veldhuizen and H. Zhu, *Modeling Segmented-in-Series SOFCs with Distributed Charge Transfer and Internal Reforming*, in: R. Steinberger-Wilckens (Ed.): 8th European Fuel Cell Forum, Paper A0203, (2008)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Numerical Methods for Reacting Gas Flow Simulations*, in: V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp.10-17, (2006)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Numerical Methods for Reacting Gas Flow Simulations*, in: P. Wesseling, E. Oñate, J. Périaux (Eds.): ECCOMAS CFD 2006, TU Delft, CDROM ISBN 90-9020970-0, (2006)

Technical Reports

- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *A class of Projected Newton methods to solve laminar reacting gas flow problems*, Technical Report at Delft University of Technology, Report 08-03, Delft, (2008)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Transient Chemical Vapor Deposition Simulations*, Technical Report at Delft University of Technology, Report 07-08, Delft, (2007)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *A Note on the Numerical Simulation of Kleijn's Benchmark Problem*, Technical Report at Delft University of Technology, Report 06-15, Delft, (2006)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Numerical Methods for CVD Simulation*, Technical Report at Delft University of Technology, Report 06-07, Delft, (2006)
- S. van Veldhuizen, C. Vuik and C.R. Kleijn, *Efficient Solution Methods for Stiff Systems of Advection-Diffusion-Reaction Equations*, Literature Study, Technical Report at the Delft University of Technology, Report 05-05, Delft, (2005)

Other

- Highlight in Annual Report & Research Programme 2007-2008, JM Burgerscentrum, Research School for Fluid Mechanics, S. van Veldhuizen, C. Vuik and C.R. Kleijn, *On numerical issues in time accurate laminar reacting gas flow solvers*

Nomenclature

A	pre-exponential factor for homogeneous reaction rate	mol, m^3 and s
A_{eq}	pre-exponential factor for fitted equilibrium constant	mol, m^3 and s
B_g	permeability	m^2
c	molar concentration	$mol \cdot m^{-3}$
c_p	specific heat at constant pressure	$J \cdot mol^{-1} \cdot K^{-1}$
d_p	particle diameter	m
D	binary diffusion coefficient	$m^2 \cdot s^{-1}$
D^e	effective binary diffusion coefficient	$m^2 \cdot s^{-1}$
D^e_{Kn}	effective Knudsen diffusion coefficient	$m^2 \cdot s^{-1}$
$\mathbb{D}^T$	multicomponent thermal diffusion coefficient	$kg \cdot m^{-1} \cdot s^{-1}$
$\mathbb{D}'$	effective multicomponent diffusion coefficient	$m^2 \cdot s^{-1}$
E	activation energy of gas phase reaction	$J \cdot mol^{-1}$
E_a	electric potential difference in the anode	V
E_c	electric potential difference in the cathode	V
E_{eq}	activation energy for fitted equilibrium constant	$J \cdot mol^{-1}$
f	species mole fraction	
F	Faradaic constant	$C \cdot mol^{-1}$
$\mathbf{g}$	gravity vector	$m \cdot s^{-2}$
H^0	standard heat of formation	$J \cdot mol^{-1}$

i_0	exchange current density	$A \cdot m^{-2}$
$i_{e,\mathrm{BV}}$	charge transfer rate in Butler-Volmer equation	$A \cdot m^{-2}$
$\mathbf{I}$	unity tensor	
$\mathbf{j}$	mass diffusive flux	$kg \cdot m^{-2} \cdot s^{-1}$
$\mathbf{J}$	molar diffusive flux	$mol \cdot m^{-2} \cdot s^{-1}$
K^g	homogeneous reaction equilibrium constant	
$k^g_{k,\mathrm{forward}}$	forward gas phase reaction rate constant	mol, m^3 and s^{-1}
$k^g_{k,\mathrm{backward}}$	backward gas phase reaction rate constant	mol, m^3 and s^{-1}
m	average molecular weight	$kg \cdot mol^{-1}$
m_i	molecular weight of species i	$kg \cdot mol^{-1}$
N	number of gas-phase species	
P	pressure	Pa
$\mathcal{P}$	net mass production rate	$kg \cdot m^{-2} \cdot s^{-1}$
q_a	charge of the anode phase	$C \cdot m^{-3}$
q_c	charge of the cathode phase	$C \cdot m^{-3}$
q_e	charge of the eletrolyte phase	$C \cdot m^{-3}$
r	radial coordinate	m
R	universal gas constant	$J \cdot mol^{-1} \cdot K^{-1}$
R^g	net molar gas phase reaction rate	$mol \cdot m^{-2} \cdot s^{-1}$
R^S	net molar surface reaction rate	$mol \cdot m^{-2} \cdot s^{-1}$
$\dot{s}$	molar production rate	$mol \cdot m^{-2} \cdot s^{-1}$
$\dot{s}_{a,e}$	faradic charge-transfer rate in the anode	$A \cdot m^{-3}$
$\dot{s}_{c,e}$	faradic charge-transfer rate in the cathode	$A \cdot m^{-3}$
S	number of surface reactions	
S^0	standard entropy of formation	$J \cdot mol^{-1} \cdot K^{-1}$
t	time	s
T	temperature	K
T_{in}	inlet temperature	K
$\mathbf{v}$	mass averaged gas velocity	$m \cdot s^{-1}$
z	axial coordinate	m

Greek Symbols

α_a	anodic syymetric factor in Butler-Volmer equation	
α_c	cathodic syymetric factor in Butler-Volmer equation	
β	temperature coefficient for reaction rate expression	
β_{eq}	temperature coefficient for fitted equilibrium constant	
γ	reactive sticking coefficient	
Γ	diffusion coefficient in general transport equation for	$\text{m}^2 \cdot \text{s}^{-1}$
$\eta_{\text{act},a}$	local activation overpotential in the anode	V
$\eta_{\text{act},c}$	local activation overpotential in the cathode	V
λ	thermal conductivity of the gas mixture	$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$
μ	viscosity of the gas mixture	$\text{kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$
μ^0	standard-state chemical potential	$\text{J} \cdot \text{mol}^{-1}$
ν	net stoichiometric coefficient for gas phase reaction	
ν'	forward stoichiometric coefficient for gas phase reaction	
ν''	backward stoichiometric coefficient for gas phase reaction	
ξ	mean free path length	m
ρ	density	$\text{kg} \cdot \text{m}^{-3}$
σ	net stoichiometric coefficient for surface reaction	
σ'	forward stoichiometric coefficient for surface reaction	
σ''	backward stoichiometric coefficient for surface reaction	
σ_a^e	effective electric conductivity of anode phase, $\text{S} \cdot \text{m}^{-1}$	
σ_c^e	effective electric conductivity of cathode phase, $\text{S} \cdot \text{m}^{-1}$	
σ_c^e	effective electric conductivity of electrolyte phase, $\text{S} \cdot \text{m}^{-1}$	
τ_g	tortuosity of gas phase	
ϕ	porosity	
Φ_a	anode electric potential	V
Φ_c	cathode electric potential	V
$\Phi_{e,a}$	electrolyte electric potential in the anode	V
$\Phi_{e,c}$	electrolyte electric potential in the cathode	V
χ	stoichiometric coefficient for surface reaction	
ω	species mass fraction	

Subscripts

C	in grid point C
center	at the cell center of control point
dep	with respect to deposition
i, j	with respect to the i^{th} / j^{th} gas species
ij	with respect to gas pair $i - j$
in	at the inflow
k	with respect to the k^{th} gas phase reaction
n, s, e, w	at the north, south, east or west wall of grid cell
N, S, E, W	at the north, south, east or west neighbor grid point
s	at the wafer surface
out	at the outflow
wall	at the wall of the reactor

Superscripts

C	due to concentration gradients
g	with respect to a gas phase reaction
T	due to temperature gradients

References

Agnew, G. D., Collins, R. D., Jorger, M., Pyke, S. H., & Travis, R. P. (2007). The components of a Rolls-Royce 1 MW SOFC system. *ECS Trans.*, *7*, 105–111.

Andersen, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., & Sorensen, D. (1995). *LAPACK users' guide* (2 ed.). Philadelphia: SIAM.

Barret, R., Berry, M., Chan, T. F., Demmel, J., Donato, J. M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., & van der Vorst, H. A. (1994). *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia: SIAM.

Berman, A. & Plemmons, R. J. (1994). *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia: SIAM.

Bertsekas, D. P. (1982). Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.*, *20*, 221–246.

Bolley, C. & Crouzeix, M. (1973). Conservation de la positivité lors de la discrétisation des problèmes d'évolution paraboliques. *RAIRO Anal. Numer.*, *12*, 237–245.

Bouteville, A. (2005). Numerical simulation applied to Chemical Vapour Deposition process. Rapid Thermal CVD and Spray CVD. *J. Optoelectronics and Advanced Materials*, *7*, 599–606.

Brown, P. N. & Saad, Y. (1990). Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, *11*, 450–481.

Coltrin, M. E., Kee, R. J., & Evans, G. H. (1989). A mathematical model of the fluid mechanics and gas-phase chemistry in a rotating Chemical Vapor Deposition reactor. *J. Electrochem. Soc*, *136*, 819–829.

Coltrin, M. E., Kee, R. J., Evans, G. H., Meeks, E., Rupley, F. M., & Grcar, J. F. (1993). SPIN (version 3.83): A FORTRAN program for modeling one-dimensional rotatingdisk/stagnation-flow Chemical Vapour Deposition reactors. Technical Report SAND91-8003.UC-401, Sandia National Laboratories, Albuquerque, NM/Livermore, CA, USA.

Coltrin, M. E., Kee, R. J., & Miller, J. (1984). A mathematical model of the coupled fluid mechanics and chemical kinetics in a Chemical Vapor Deposition reactor. *J. Electrochem. Soc*, *131*, 425–434.

Coltrin, M. E., Kee, R. J., Rupley, F. M., & Meeks, E. (1996). Surface Chemkin III. Technical Report SAND96-8217, SANDIA National Laboratories, Albuquerque, NM/Livermore, CA, USA.

Costamagna, P., Selimovic, A., Borghi, M. D., & Agnew, G. (2004). Electrochemical model of the integrated planar solid oxide fuel cell (IP-SOFC). *Chem. Eng. J.*, *102*, 61–69.

Dahlquist, G. (1963). A special stability problem for linear multistep methods. *BIT*, *3*, 27–43.

Dembo, R. S., Eisenstat, S. C., & Steihaug, T. (1982). Inexact Newton methods. *SIAM J. Numer. Anal.*, *19*, 400–408.

Dembo, R. S. & Steihaug, T. (1983). Truncated newton algorithms for large scale optimization. *Math. Programming*, *26*, 190–212.

Dennis, J. E. & Schnabel, R. B. (1983). *Numerical methods for unconstrained optimization and nonlinear equations*. Series in Automatic Computing. Englewood Cliffs, NJ: Prentice-Hall.

Duff, I. S. & Koster, J. (1999). The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM J. Matrix Anal. Appl.*, *20*(4), 889–901.

Eisenstat, S. C. & Walker, H. F. (1994). Globally convergent Inexact Newton methods. *SIAM J. Optimization*, *4*, 393–422.

Eisenstat, S. C. & Walker, H. F. (1996). Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, *17*, 16–32.

Faber, V. & Manteuffel, T. (1984). Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM J. Num. Anal.*, *21*, 356–362.

Fluent (1995). Fluent is a product of Fluent Inc., 10 Cavendish Court, Lebanon, NH, USA. `http://www.fluent.com`.

Gardner, F. J., Day, M. J., Brandon, N. P., Pashley, M. N., & Cassidy, M. (2000). SOFC technology development at Rolls-Royce. *J. Power Sources*, *86*, 122–129.

Geyling, F. T. (1994). Benchmarking computational fluid dynamics (CFD) codes for rapid thermal processing (RTP) simulation. Technical Report 94012188A-ENG, SEMATECH.

Goodwin, D. G. (2008). Cantera. `http://www.cantera.org`.

Gottlieb, S., Shu, C. W., & Tadmor, E. (2001). Strong stability-preserving high-order time discretization methods. *SIAM Review*, *43*, 89–112.

Haberman, B. A. & Young, J. B. (2004). Three-dimensional simulation of chemically reacting gas flows in the porous support structure of an integrated planar solid oxide fuel cell. *Int. J. Heat Mass Transfer*, *47*, 3617–3629.

Haberman, B. A. & Young, J. B. (2006). Diffusion and chemical reaction in the porous structures of solid oxide fuel cells. *J. Fuel Cell Sci. Technol.*, *3*, 312–321.

Haberman, B. A. & Young, J. B. (2008). A detailed three-dimensional simulation of an IP-SOFC stack. *J. Fuel Cell Sci. Technol.*, *5*, 1–12.

Hairer, E., Nørsett, S. P., & Wanner, G. (1987). *Solving ordinary differential equations I: nonstiff problems*. Number 8 in Springer Series in Computational Mathematics. Berlin: Springer.

Hairer, E. & Wanner, G. (1996). *Solving ordinary differential equations II: stiff and differential-algebraic problems*. Number 14 in Springer Series in Computational Mathematics. Berlin: Springer.

Hecht, E. S., Gupta, G. K., Zhu, H., Dean, A. M., Kee, R. J., Maier, L., & Deutschmann, O. (2005). Methane reforming kinetics within a Ni-YSZ SOFC anode. *Appl. Catal. A*, *295*, 40–51.

Hertz-Fischler, R. (1998). *A mathematical history of the golden number*. Mineola, NY: Dover Publications, Inc.

Hitchman, M. L. & Jensen, K. F. (1993). *Chemical Vapor Deposition- Principles and Applications*. London: Academic Press.

Hundsdorfer, W., Ruuth, S. J., & Spiteri, R. J. (2003). Monoticity-preserving linear multistep methods. *SIAM J. Numer. Anal.*, *41*, 605–623.

Hundsdorfer, W. & Verwer, J. G. (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Number 33 in Springer Series in Computational Mathematics. Berlin: Springer.

Jensen, K. F. (1988). Modeling of Chemical Vapor Deposition reactors. In *Modeling of Chemical Vapor Deposition reactors for semiconduction fabrication*. Berkeley. Course notes.

Jensen, K. F. & Graves, D. B. (1983). Modelling and analysis of low pressure CVD reactors. *J. Electrochem. Soc.*, *130*, 1950–1957.

Kee, R. J., Coltrin, M. E., & Glarborg, P. (2003). *Chemically reacting flow: theory and practice*. New Jersey: Wiley.

Kee, R. J., Rupley, F. M., & Miller, J. A. (1989). Chemkin II : A FORTRAN chemical kinetics package for the analysis of gas-phase kinetics. Technical Report SAND89-8009B.UC-706, SANDIA National Laboratories, Albuquerque, NM, USA.

Kee, R. J., van Veldhuizen, S., & Zhu, H. (2008). Modeling segmented-in-series SOFCs with distributed charge transfer and interna reforming. In Steinberger-Wilckens, R. (Ed.), *8th European Fuel Cell Forum*, Oberrohrdorf, Swiss. European Fuel Cell Forum. Paper A0203.

Kelley, C. T. (1995). *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia: SIAM.

Kelley, C. T. (2003). *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. Philadelphia: SIAM.

Kleijn, C. R. (1991). *Transport phenomena in Chemical Vapor Deposition reactors*. PhD thesis, Delft University of Technology, Delft.

Kleijn, C. R. (1995). Chemical Vapor Deposition processes. In M. Meyyappan (Ed.), *Computational modeling in semiconductor processing* chapter 4, (pp. 97–229). Boston: Artech House.

Kleijn, C. R. (2000). Computational modeling of transport phenomena and detailed chemistry in Chemical Vapor Deposition- A benchmark solution. *Thin Solid Films*, *365*, 294–306.

Kleijn, C. R., van der Meer, T. H., & Hoogendoorn, C. J. (1989). A mathematical model for LPCVD in a single wafer reactor. *J. Electrochem. Soc.*, *136*, 3423–3433.

Knoll, D. A. & Keyes, D. E. (2004). Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, *193*, 357–397.

Kuijlaars, K. J., Kleijn, C. R., & van den Akker, H. E. A. (1995). Multi-component diffusion phenomena in multiple-wafer chemical vapour deposition reactors. *Chem. Eng. J.*, *57*, 127–136.

Lankhorst, A. M., Paarhuis, B. D., Terhorst, H. J. C. M., Simons, P. J. P. M., & Kleijn, C. R. (2007). Transient ALD simulations for a multi-wafer reactor with trenched wafers. *Surf. Coat. Technol.*, *201*, 8842–8848.

LeVeque, R. J. (2002). *Finite Volume methods for conservation laws*. Cambridge Texts in Applied Mathematics. Cambridge University Press.

Mason, E. A. & Malinauskas, A. P. (1983). *Gas Transport in Porous Media: The Dusty Gas Model*. New York: American Elsevier.

Nakamura, K., Yamashita, S., Tsutomu, S., & Seyama, T. (2005). Development of SOFC power generation system using segmented-in-series cell stacks operating at low temperatures. In *Proceedings of the 1st European Fuel Cell Technology and Applications Conference*, (pp.41). ASME.

Ortega, J. M. & Rheinboldt, W. C. (2000). *Iterative solution of Nonlinear equations in several variables*. Number 30 in Classics in Applied Mathematics. Philadelphia: SIAM. Reprint of the 1970 original.

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow*. Washington DC: Hemisphere Publishing Corp.

Phoenics-CVD (1995). Phoenics-CVD is a product of CHAM Ltd. It was developed under EC-ESPRIT project 7161, and is described in detail in The Phoenics Journal 8 (4),1995.

Rosenbrock, H. H. (1963). Some general implicit processes for the numerical solution of differential equations. *Comput. J.*, *5*, 329 – 330.

Saad, Y. (2003). *Iterative methods for sparse linear systems* (2nd ed.). Philadelphia: SIAM.

Shadid, J. N., Tuminaro, R. S., & Walker, H. F. (1997). An inexact Newton method for fully coupled solution of the Navier-Stokes equations with mass and heat transport. *J. Comput. Phys.*, *137*, 155–185.

Shampine, L. F. (1994). *Numerical solution of ordinary differential equations*. New York: Chapmann & Hall.

Shampine, L. F., Sommeijer, B. P., & Verwer, J. G. (2005). IRKC: an IMEX solver for stiff diffusion-reaction PDEs. Technical Report MAS-*E0513, CWI, Amsterdam, The Netherlands.

Singhal, S. C. & Kendall, K. (Eds.). (2003). *High Temperature Solid Oxide Fuel Cells: Fundamentals, Design, and Applicatons*. Elsevier.

Sommeijer, B. P., Shampine, L. F., & Verwer, J. G. (1997). RKC: An explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, *88*, 315–326.

Sonneveld, P. & van Gijzen, M. (2007). IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems. Technical Report 07-07, TU Delft, Delft, The Netherlands.

Sportisse, B. (2000). An analysis of operator splitting in the stiff case. *J. Comput. Phys.*, *161*, 140–168.

Stoer, J. & Bulirsch, R. (1980). *Introduction to Numerical Analysis*. New York: Springer-Verlag.

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, *5*, 506–517.

Strang, G. (2003). *Introduction to Linear Algebra*. Massachusetts: Wellesley-Cambridge Press.

Tang, T. (1998). Convergence analysis of operator-splitting methods applied to hyperbolic conservation laws with stiff source terms. *SIAM J. Numer. Anal.*, *35*, 1939–1968.

TNO Science and Industry (2007). *CVD-X User Manual* (Version 4.0 ed.). TNO Science and Industry.

Tuminaro, R. S., Walker, H. F., & Shadid, J. N. (2002). On backtracking failure in Newton-GMRES methods with a demonstration for the Navier-Stokes equations. *J. Comput. Phys.*, *180*, 549–558.

van der Houwen, P. J. (1996). The development of Runge-Kutta methods for partial differential equations. *Appl. Numer. Math.*, *20*, 261–273.

van der Vorst, H. A. (1992). Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, *13*(2), 631–644.

van Veldhuizen, S. (2005). Efficient solution methods for stiff systems of advection-diffusion-reaction equations, literature study. Technical Report 05-05, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2006a). A note on the numerical simulation of Kleijn's benchmark problem. Report 06-15, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2006b). Numerical methods for CVD simulation. Report 06-07, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2006c). Numerical methods for reacting gas flow simulations. In Alexandrov, V., van Albada, G., Sloot, P., & Dongarra, J. (Eds.), *Computational Science–ICCS 2006: 6th International Conference, Reading, UK, May 28-31, 2006. Proceedings, Part II*, (pp. 10–17)., Berlin. Springer. Lecture Notes in Computer Science 3992.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2006d). Numerical methods for reacting gas flow simulations. In Wesseling, P., Onate, E., & Periaux, J. (Eds.), *European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, Delft. TU Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2007a). Comparison of numerical methods for transient CVD simulations. *Surf. Coat. Technol.*, *201*, 8859–8862.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2007b). Numerical methods for reacting gas flow simulations. *Internat. J. Multiscale Eng.*, *5*, 1–10.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2007c). Transient Chemical Vapor Deposition simulations. Report 07-08, Delft University of Technology, Delft Institute of Applied Mathematics, Delft.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2008a). A class of projected Newton methods to solve laminar reacting flow problems. Report 08-03, Delft University of Technology, Delft Institute of Applied Mathematics, Delft. Submitted to J. Sci. Comput.

van Veldhuizen, S., Vuik, C., & Kleijn, C. R. (2008b). Comparison of ODE methods for laminar reacting gas flow simulations. *Num. Meth. Part. Diff. Eq.*, *24*, 1037–1054.

Verwer, J. G. & Sommeijer, B. P. (2004). An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations. *SIAM J. Sci. Comput.*, *25*, 1824–1835.

Verwer, J. G., Sommeijer, B. P., & Hundsdorfer, W. (2004). RKC time-stepping for advection-diffusion-reaction problems. *J. of Comp. Physics*, *201*, 61–79.

Verwer, J. G., Spee, E. J., Blom, J. G., & Hundsdorfer, W. (1999). A second order Rosenbrock method applied to photochemical dispersion problems. *SIAM J. Sci. Comput.*, *20*, 1456–1480.

Verwer, J. G. & Sportisse, B. (1998). A note on operator splitting in a stiff linear case. Technical Report MAS-R9830, CWI, Amsterdam.

Wahl, G. (1977). Hydrodynamic description of CVD processes. *Thin Solid Films*, *40*, 13–26.

Wesseling, P. (1992). *An introduction into Multigrid Methods*. Chichester: John Wiley and Sons.

Wesseling, P. (1996). Von Neumann stability conditions for the convection-diffusion equation. *IMA J. of Num. Anal.*, *16*, 583–598.

Wesseling, P. (2001). *Principles of Computational Fluid Dynamics*. Number 29 in Springer Series in Computational Mathematics. Berlin: Springer.

Zhu, H. & Kee, R. J. (2008). Modeling distributed charge-transfer processes in SOFC membrane electrode assemblies. *J. Electrochem. Soc.*, *155*, B715–B729.

Zhu, H., Kee, R. J., Janardhanan, V. M., Deutschmann, O., & Goodwin, D. G. (2005). Modeling elementary heterogeneous chemistry and electrochemistry in solid-oxide fuel cells. *J. Electrochem. Soc.*, *152*, A2427–A2440.