Inaugural Dissertation

Erlangung der Doktorwürde

der

Naturwissensschaftlich-Matematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Diplom-Ingenieur Cyrille Waguet

September, 2000

Adaptive Finite Element Computation of Chemical Flow Reactors

Gutachter: Professor Dr. Rolf Rannacher

Contents

Introduction

Flow reactors are used in many applications in industry and research. Complex interactions in the reactor, such as superposition of convection and diffusion processes with chemical reactions in the gas phase or at the walls, make it difficult for experimental data to be correctly interpreted. By means of a detailed numerical simulation, these various effects can be distinguished and the interacting processes occurring within reactive mixing flows are easier to understand.

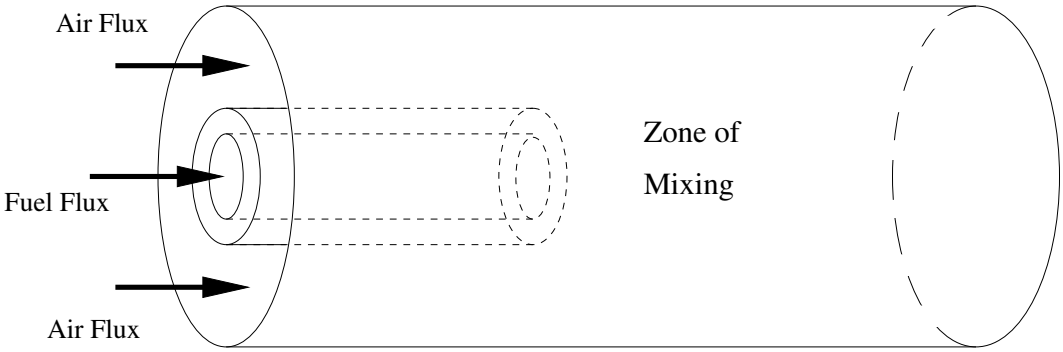

Figure 1: *Flow reactor made of an inner and an outer tube where two gases enter and get in contact at the outlet of the central tube.*

Thus the main interest in the simulation of flow reactors is the comprehension of the complex interplay between flow, mixing processes and reaction processes. To describe the chemical and physical processes taking place in reactive flows, many chemical species are to be considered with often a few hundred elementary reactions. Considering the equations for velocities, pressure, temperature, and each species, the system of PDEs modelling the reactive flow contains usually between 10 and 50 equations and is highly non-linear. The leading terms in these equations may vary in space and time. In the reaction zones, the system may become reaction-dominated through stiff source terms. In other parts of the domain where chemical reactions are weak, either the convection terms (by high Reynolds number) or the diffusive terms (as in non-reactive boundary layers) may be predominant.

Several methods for the simulation of reactive flows have already been implemented, usually based on finite difference or finite volume discretizations on tensor product meshes (see for instance [18], [55]). A code based on finite differences has recently been applied to the simulation of a low-pressure flow reactor for kinetic studies in [46], in order to improve existing methods (as plug-flow techniques) for evaluating data from isothermal flow kinetic measurements. It has been developed for the low Mach-number regime and makes use of splitting techniques for variables and spatial dimensions thereby reducing the computational effort. Numerical results of full reactive flow simulation have been compared with the measurement of elementary relaxation and vibrational energy transfer processes. As a model system for a simple kinetic process the heterogeneous relaxation of vibrationally excited hydrogen ($H_2(v''=1)$) and its energy transfer in collisions with deuterium ($D_2(v''=0)$) was considered (see Chapter 6):

$$\begin{aligned} H_2(v''=1) &\xrightarrow{\text{wall}} H_2(v''=0), \\ H_2(v''=1) + D_2(v''=0) &\longrightarrow H_2(v''=0) + D_2(v''=1). \end{aligned}$$

This made it possible to evaluate species wall deactivation probabilities and reaction rate constants for vibrational energy transfer. However, this simulation did not bring enough information about the precision on the computed quantities, which could assure that the error done on these quantities was lower than a given tolerance. Nor did the tensor-product mesh allow to efficiently control the accuracy of the calculation locally in the zones of the flow tube that were of interest. Moreover, due to some instabilities in the method, it was necessary to use pseudo-time stepping to obtain steady solutions, which could have been avoided in some cases.

In order to eliminate these weaknesses and achieve better accuracy in the solution with reasonable computational effort, we develop in this work a new method for the simulation of chemical flow reactors with precise evaluation of some physical quantities. We derive this method from recent techniques for adaptive mesh refinement which allows to reduce the numerical effort and nevertheless achieve good or even better accuracy in the data that may be of interest compared to a straightforward tensor product approach. This makes possible on the one hand to simulate flow reactors on simple workstations or PCs without any compromise with respect to the quality of the computed solution, and on the other hand, on super-computers, to reach an accuracy that could not be achieved on simple tensor product meshes or on locally adapted meshes constructed according to *ad hoc* criteria, usually justified on physical grounds, whose impact on the accuracy of the numerical solution is difficult to assess.

Chapter 1 discusses the dimension reduction of the computational domain. For the simulation of circular flow tubes assuming an axial symmetry, it is

sufficient to consider only half of an meridional section of the tube to describe the reactive flow. We discuss here problems invariant under rotation, and the derivation of weighted Sobolev spaces needed in the weak formulation of the system to be solved.

The model considered consists of the compressible Navier-Stokes equations with additional convection-diffusion-reaction equations for the chemical species. The goal is the simulation of stationary or quasi-stationary reactive flows at low Mach number for the evaluation of kinetic reaction parameters as well as process optimization of chemical reaction systems in flow reactors. The complete model for multispecies flows is presented in Chapter 2 and then restricted by simplifying the diffusive part of the species transport as well as taking into account the low-Mach number flow state, in order to make fast computations possible without too much loss in the model accuracy according to the physics. The reaction model is also presented and the form of the chemical source terms is discussed. Further the physical constraints on the model are explained.

The discretization of the equations is discussed in Chapter 3. We use a finite element method based on bilinear elements defined on rectangles (Q_1 elements). The standard Galerkin discretization using Q_1 elements is not stable and has to be stabilized. Details are given about the pressure stabilization and the streamline diffusion methods for steady and unsteady compressible flows at low Mach number.

The highly non-linear system obtained requires very efficient numerical methods. Therefore a robust non-linear solver is needed. A defect correction method with step size control is developed by approximating the Newton matrix. The degree of approximation required is assessed according to consistence and solvability of the corresponding linear systems.

In Chapter 4 the solver is described. The outer iteration is based on defect correction and the inner large linear problems are solved by an iterative method GMRES with the help of a multigrid method as preconditioner. GMRES and multigrid methods are among the most efficient modern techniques for solving large scale algebraic systems resulting from finite element discretizations of PDEs. The multigrid method needs an appropriate smoother for reactive flow problems on locally refined meshes. The development of a Vanka smoother for the Navier-Stokes part of the system and the use of Gauss-Seidel or ILU smoothing for the chemical part lead to an efficient and robust method.

Another important part of this work deals with error control and mesh adaptivity. The aim is to achieve reliability in the sense that physically relevant derived quantities, which can be thought of as functionals of the solution, are approximated to within a given tolerance. The use of duality arguments leads to the control of the error in functionals of the solution, which can be quantities such as point values of the temperature or line averages of mass fractions

(which corresponds to a CARS signal for instance, see Chapter 6). The mesh adaptivity based on an a posteriori error estimate gives us the possibility to refine the mesh locally only in the zones where it is necessary in order to compute these quantities with the required accuracy. We treat this problem of adaptivity and accurate quantity computations in Chapter 5. The concept of error estimation for functionals of the solution is explained and we apply this method to produce "optimal" meshes for reliable and efficient computation of reactive flows in flow reactors. A quantitative error estimation of functionals is especially important for comparison between simulation and experiment to validate the underlying model. The model and numerical method developed in this work are indeed validated through experimental measurement which also provides the data essential for parameter estimation, such as deactivation probabilities for vibrationally-excited H_2 molecules.

In order to test the efficiency of the adaptive method and of the solvers, we consider in Chapter 6 three relevant problems in flow reactors:

- CARS (Coherent Antistokes Raman Spectroscopy) measurement of deactivation reactions and reaction rate for energy transfer of vibrationally-excited H_2 molecules,
- LIF-Spectroscopy for the kinetic analysis of reactions between NH and NO molecules as well as between NH and O_2 molecules in the case of high temperatures, and
- CA-CVD (Combustion Aided Chemical Vapor Deposition) for the optimization of a diamond deposition process.

In the first case, the mixture consist of 9 species with heterogeneous reactions of deactivation on the wall as well as gas-phase reactions between H_2 and D_2 molecules. The complete chemical model consist of 27 gas-phase reactions and 5 wall reactions. The evolution of the concentration of some species is measured along the axis of the tube on well defined measurement points. The solution method with adaptive mesh refinement is applied to compute the evolution of the species concentration along the axis with optimal precision on these measurement points. We are then able to compare accurate simulation results with measurements and thus derive reaction rates.

In the second case, the mixture considered (based on products of reactions between NO_2 and H_2) consists of 8 species with homogeneous and heterogeneous reactions with heated walls (Dirichlet boundary conditions for the temperature at the wall). The temperature range to be considered is 300K (temperature of the incoming gas flow) to 1700K. These high temperature gradients induce some numerical instabilities in the inflow region so that only a quasi-stationary solution can be found. We have to use here a time step method to be able to converge to a solution.

A CA-CVD experiment (see [32] and [23]) has also been simulated. The aim is to optimize the quality and quantity of diamond deposition on a substrate. The system to be solved is more complex than the former system for the simulation of the CARS experiment. The mixture contains 39 species and the reaction model consists of 358 chemical reactions. An injection of methane is done from a pipe into a gas mixture made of products of a H_2/O_2 flame. It has been shown that the deposition of diamond strongly depends on the concentration of CH_3 near the substrate. Working with such a large system of equations does not allow to use simple structured meshes without error control on the values we are interested in. The adaptive process developed in this work not only allow us to compute accurately physical values - such as the CH_3 concentration near the substrate - but also to deal with more complicated chemical processes. This was made possible by improving the performance of the simulation process with respect to already existing codes. Using an adaptive refinement process based on error functionals allows us to get higher accuracy on some physical value of interest with a given number of cells, and thus drastically reduce memory requirements. Moreover, the implementation of robust and efficient solvers make it possible to reduce the computation time. All computations here can be done on a workstation.

The basic principles of finite element methods is assumed to be known. Some references are given for an introduction to finite element discretizations.

Chapter 1

Axisymmetric Problems and Dimension Reduction

Most physical problems are naturally formulated as boundary problems in three dimensional domains. However three dimensional computations are very expensive and sometimes practically impossible on workstations. It is therefore necessary to rewrite the problem with two dimensional equations. This is obtained by assuming that the dependency of the parameters, data and solution with respect to one of the three variables can be neglected, which is justified in many situations. Here we are interested in the case where the three-dimensional computation domain is invariant under rotation around an axis. Thus, without any approximation, the problem can be transformed into a family of two dimensional equations on the Fourier coefficients (cf. [9]). Moreover, if the data satisfy suitable axisymmetry properties, only the Fourier coefficient of order 0 subsists, so that the three dimensional problem can be reduced to a two dimensional one. We will deal with this later case in this work. The problems we are interested in are indeed invariant under rotation (see later).

The axisymmetric functions which belong to standard Sobolev spaces on the three dimensional domain can be mapped onto functions in the corresponding two dimensional domain. These new functions belong to weighted Sobolev spaces, the weight being the distance to the symmetry axis. We characterize these functions as the elements of the weighted spaces such that suitable traces vanish on the rotation axis.

All this leads to transform an axisymmetric boundary value problem on the three dimensional domain into an equivalent problem on the corresponding two dimensional domain. For more details see [11], [41] and [2].

1.1 Description of Axisymmetric Problems

For a generic point in $\mathbb{R}^3$, we use both cartesian coordinates (x, y, z) and cylindrical coordinates (r, θ, z) in $\mathbb{R}_+ \times]-\pi, \pi] \times \mathbb{R}$, with

$$r = \sqrt{x^2 + y^2} \qquad \text{and} \qquad \theta = \begin{cases} -\arccos \frac{x}{r} & \text{if } y < 0, \\ \arccos \frac{x}{r} & \text{if } y \geq 0. \end{cases} \tag{1.1}$$

In $\mathbb{R}^2$ we use the cartesian coordinates (r, z) and we define the half-space $\mathbb{R}^2_+$ as the set of points in $\mathbb{R}^2$ with positive coordinate r.

Let Ω denote a bounded domain contained in $\mathbb{R}^2_+$. The axisymmetric domain $\breve{\Omega}$ is the three-dimensional set obtained by rotating Ω around the axis $r = 0$.

We are interested in two-dimensional domains of the following types for the reactive flow computations in Chapter 6:

- CARS flow reactor:

Figure 1.1: $\Omega =$ *half axial section of the CARS flow reactor shown in Fig. 1.*

- CVD flow reactor:

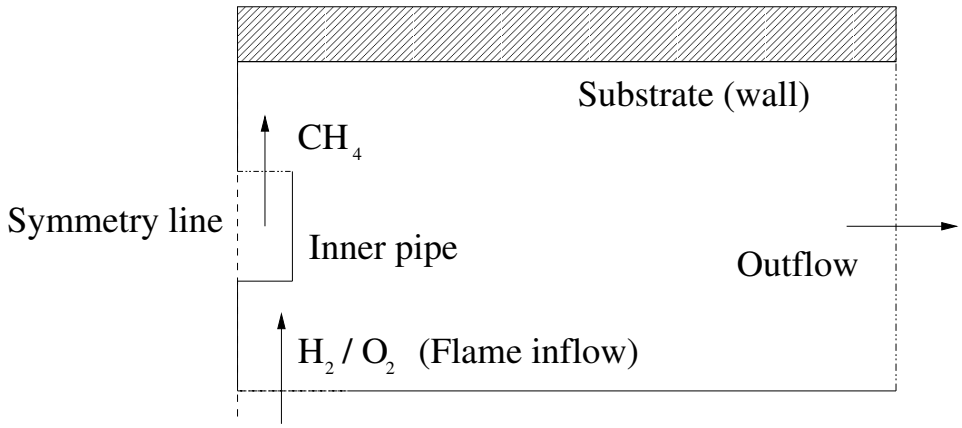

Figure 1.2: $\Omega =$ *half axial section of the CVD reactor*

We denote by Γ_0 the part of the boundary $\partial\Omega$ contained in the axis $r = 0$, i.e. the symmetry line. We set $\Gamma = \partial\Omega \backslash \Gamma_0$. The boundary $\partial\Omega$ is a polygon, i.e. the union of a finite number of segments.

The corresponding three-dimensional domain $\breve{\Omega}$, corresponding to the whole flow reactor shown in Fig. 1, is defined as:

$$\breve{\Omega} = \{(x, y, z) \in \mathbb{R}^3 \quad | \quad (r, z) \in \Omega \cup \Gamma_0 \quad \text{and} \quad -\pi < \theta \leq \pi\}. \tag{1.2}$$

Let $\mathcal{R}_\eta$ denote the rotation with angle η with respect to the axis $r = 0$ in $\mathbb{R}^3$, i.e.

$$\mathcal{R}_\eta(x, y, z) = (x\cos\eta - y\sin\eta,\ x\sin\eta + y\cos\eta,\ z). \tag{1.3}$$

Of course, $\breve{\Omega}$ is invariant by any rotation $\mathcal{R}_\eta$. The unit outward normal vector $\breve{n}$ to $\breve{\Omega}$ is obtained by rotating the unit outward vector n to Ω on Γ.

1.2 Problems Invariant under Rotation

The problems which are considered in this work are invariant under rotation. Let us consider the boundary value problem $[\breve{A}, \breve{B}]$ on $\breve{\Omega}$ where the unknown is a vectorial function $\breve{v}$ with M components:

$$\begin{cases} \breve{A}\breve{v} = \breve{f} & \text{in} \quad \breve{\Omega}, \\ \breve{B}\breve{v} = \breve{g} & \text{on} \quad \partial\breve{\Omega}. \end{cases} \tag{1.4}$$

The symbol $\breve{}$ over a letter means that the corresponding function, distribution or operator is defined on $\breve{\Omega}$. Here $\breve{A}$ is a linear system of partial differential operators and $\breve{B}$ is a system of boundary differential operators.

Definition 1. *Problem $[\breve{A}, \breve{B}]$ is said to be invariant under rotation if the following property holds for any smooth function $\breve{v}$ from $\breve{\Omega}$ into $\mathbb{R}^M$:*

$$\forall \eta \in [-\pi, \pi] : \qquad \begin{cases} \breve{A}(\breve{v} \circ \mathcal{R}_\eta) = (\breve{A}\breve{v}) \circ \mathcal{R}_\eta, \\ \breve{B}(\breve{v} \circ \mathcal{R}_\eta) = (\breve{B}\breve{v}) \circ \mathcal{R}_\eta. \end{cases} \tag{1.5}$$

Equivalently, problem $[\breve{A}, \breve{B}]$ is invariant under rotation if the operators $\breve{A}$ and $\breve{B}$ can be written in the following form in cylinder coordinates (r, θ, z):

$$\begin{aligned} \breve{A}(x, y, z; \partial_x, \partial_y, \partial_z) &= \breve{A}(r, z; \partial_r, \partial_\theta, \partial_z), \\ \breve{B}(x, y, z; \partial_x, \partial_y, \partial_z) &= \breve{B}(r, z; \partial_r, \partial_\theta, \partial_z), \end{aligned} \tag{1.6}$$

i.e. with coefficients independent of the variable θ. A basic example is the Laplace operator

$$\triangle = \partial_x^2 + \partial_y^2 + \partial_z^2 = \partial_r^2 + \frac{1}{r}\,\partial_r + \frac{1}{r^2}\,\partial_\theta^2 + \partial_z^2 \tag{1.7}$$

Dirichlet boundary conditions or, more generally, conditions which only depend on the normal derivative $\partial_{\check{n}}$ to the boundary, are invariant under rotation.

1.3 Data and Solutions Invariant under Rotation

Definition 2. *A function $\breve{v}$ is said to be invariant under rotation if the following property holds*

$$\forall \eta \in [-\pi, \pi] : \qquad \breve{v} \circ \mathcal{R}_\eta = \breve{v}. \tag{1.8}$$

Problems which are invariant under rotation are associated with functions invariant under rotation: if problem $[\breve{A}, \breve{B}]$ satisfies (1.5) and if $\breve{v}$ is invariant under rotation, so are $\breve{f}$ and $\breve{g}$; the converse property holds when problem $[\breve{A}, \breve{B}]$ has at most one solution.

When the operators $\breve{A}$ and $\breve{B}$ as well as the data $\breve{f}$ and $\breve{g}$ are invariant under rotation, we easily see that this problem is closely linked to the two-dimensional problem

$$\begin{cases} Av = f & \text{in} \quad \Omega, \\ Bv = g & \text{on} \quad \Gamma, \end{cases} \tag{1.9}$$

where

$$\begin{aligned} f(r,z) &= \breve{f}(x,y,z), \qquad & A(r,z;\partial_r,\partial_z) &= \tilde{A}(r,z;\partial_r,0,\partial_z), \\ g(r,z) &= \breve{g}(x,y,z), \qquad & B(r,z;\partial_r,\partial_z) &= \tilde{B}(r,z;\partial_r,0,\partial_z), \end{aligned} \tag{1.10}$$

$\tilde{A}$ and $\tilde{B}$ being defined in (1.6).

Thus in the case of a problem invariant under rotation, we have actually reduced the number of variables from 3 to 2.

When problem $[\breve{A}, \breve{B}]$ is invariant under rotation, and if the data $\breve{f}$ and $\breve{g}$ are invariant under rotation, it is readily checked that the following propositions are equivalent:

- $\breve{v}$ is a solution of $[\breve{A}, \breve{B}]$ and is invariant under rotation,
- v is a solution of $[A, B]$.

1.4 Basic Formulas

With each coordinate system, we associate an orthonormal basis: (e_x, e_y, e_z) for the cartesian system, and (e_r, e_θ, e_z) for the cylindrical system. The derivative with respect to each of these coordinates is denoted by ∂ indexed by the coordinate. From the basic identities

$$\partial_x = \partial_r \ \cos\theta - \frac{1}{r}\,\partial_\theta \ \sin\theta, \qquad \partial_y = \partial_r \ \sin\theta - \frac{1}{r}\,\partial_\theta \ \cos\theta$$

we derive the formulas for operators acting on scalar functions and on vectorial functions. A function $\breve{v}$ with values in $\mathbb{R}^3$ is written either in cartesian coordinates $v_x\,e_x + v_y\,e_y + v_z\,e_z$ or in cylindrical coordinates $v_r\,e_r + v_\theta\,e_\theta + v_z\,e_z$.

The problems we are interested in are invariant under rotation. Thus the derivative according to the variable θ as well as the component v_θ of the vector defined above vanish, which leads to the following formulas:

- For scalar functions:

	cartesian coordinates	cylindrical coordinates
∇v	$\partial_x v\,e_x + \partial_y v\,e_y + \partial_z v\,e_z$	$\partial_r v\,e_r + \partial_z v\,e_z$
$\triangle v$	$\partial_x^2 v + \partial_y^2 v + \partial_z^2 v$	$\partial_r^2 v + \frac{1}{r}\,\partial_r v + \partial_z^2 v$

- For vectorial functions:

	cartesian coordinates	cylindrical coordinates
$\nabla.\breve{v}$	$\partial_x v_x + \partial_y v_y + \partial_z v_z$	$\partial_r v_r + \frac{1}{r}\,v_r + \partial_z v_z$
$\triangle\breve{v}$	$(\partial_x^2 v_x + \partial_y^2 v_x + \partial_z^2 v_x)\,e_x$ $+(\partial_x^2 v_y + \partial_y^2 v_y + \partial_z^2 v_y)\,e_y$ $+(\partial_x^2 v_z + \partial_y^2 v_z + \partial_z^2 v_z)\,e_z$	$(\partial_r^2 v_r + \frac{1}{r}\,\partial_r v_r + \partial_z^2 v_r - \frac{1}{r^2}\,v_r)\,e_r$ $+\,(\partial_r^2 v_z + \frac{1}{r}\,\partial_r v_z + \partial_z^2 v_z)\,e_z$
$\nabla\breve{v}$	$\begin{bmatrix} \partial_x v_x & \partial_x v_y & \partial_x v_z \\ \partial_y v_x & \partial_y v_y & \partial_y v_z \\ \partial_z v_x & \partial_z v_y & \partial_z v_z \end{bmatrix}$	$\begin{bmatrix} \partial_r v_r & 0 & \partial_r v_z \\ 0 & v_r/r & 0 \\ \partial_z v_r & 0 & \partial_z v_z \end{bmatrix}$

1.5 Weighted Sobolev Spaces

In the problems we consider, the solution is sought in a Sobolev space or a product of Sobolev spaces. From the space $L^2(\breve{\Omega})$ of square integrable functions for the measure $dx\,dy\,dz$, the Sobolev spaces $H^s(\breve{\Omega})$ for any positive integer s are defined. Then we derive the spaces $H_0^s(\breve{\Omega})$ as the closure in $H^s(\breve{\Omega})$ of the space $C_0^\infty(\breve{\Omega})$ and finally the spaces $H^{-s}(\breve{\Omega})$ as the dual spaces of $H^s(\breve{\Omega})$.

1.5.1 Definition and Properties of the Weighted Spaces

The space $L^2_\alpha(\Omega)$ is defined as the set of measurable functions w such that

$$\|w\|_{L^2_\alpha(\Omega)} = \left(\int_\Omega w^2(r,z)\, r^\alpha\, dr\, dz\right)^{\frac{1}{2}} < +\infty. \tag{1.11}$$

For any positive integer s, $H^s_\alpha(\Omega)$ is the space of functions w in $L^2_\alpha(\Omega)$ such that their partial derivatives of order $\leq s$ belong to $L^2_\alpha(\Omega)$. It is provided with the semi-norm

$$|w|_{H^s_\alpha(\Omega)} = \left(\sum_{l=0}^{s} \|\partial_r^l \partial_z^{s-l} w\|^2_{L^2_\alpha(\Omega)}\right)^{\frac{1}{2}}, \tag{1.12}$$

and with the norm

$$\|w\|_{H^s_\alpha(\Omega)} = \left(\sum_{l=0}^{s} |w|^2_{H^l_\alpha(\Omega)}\right)^{\frac{1}{2}} \tag{1.13}$$

Thus it is a Hilbert space.

We state the principal results in the following propositions. We first define a mapping for scalar functions. We are interested in the characterization of the functions in $H^s(\breve{\Omega})$ which are invariant under rotation in the sense (1.8). We denote the corresponding subspace by $\breve{H}^s(\breve{\Omega})$. Any element $\breve{v}$ in $\breve{H}^s(\breve{\Omega})$ is completely characterized by the function v defined by

$$v(r,z) = \breve{v}(x,y,z).$$

Proposition 1. *Let s be a positive integer. The mapping: $\breve{v} \to v$ is one-to-one from $\breve{H}^s(\breve{\Omega})$ onto the space $H^s_+(\Omega)$ defined as follows:*

- *If s is not an even integer,*

$$H^s_+(\Omega) = \left\{ w \in H^s_1(\Omega); \quad \partial_r^{2j-1} w|_{\Gamma_0} = 0, \quad 1 \leq j \leq \frac{s}{2} \right\}, \tag{1.14}$$

endowed with the natural norm

$$\|w\|_{H^s_+(\Omega)} = \|w\|_{H^s_1(\Omega)}; \tag{1.15}$$

- *if s is an even integer,*

$$H^s_+(\Omega) = \big\{ w \in H^s_1(\Omega); \quad \partial_r^{2j-1} w|_{\Gamma_0} = 0, \quad 1 \le j \le \frac{s}{2}, \\ \text{and} \quad \partial_r^{s-1} w \in L^2_{-1}(\Omega) \big\}, \tag{1.16}$$

endowed with the natural norm

$$\|w\|_{H^s_+(\Omega)} = \left(\|w\|^2_{H^s_1(\Omega)} + \|\partial_r^{s-1} w\|^2_{L^2_{-1}(\Omega)} \right)^{1/2}. \tag{1.17}$$

And then a mapping for vectorial functions. We are interested in triple of functions $\breve{v} = (v_x, v_y, v_z)$ in cartesian coordinates in $H^s(\breve{\Omega})^3$ which also satisfy (1.8) with $\mathcal{I}_\eta = \mathcal{R}_{-\eta}$. This space is also denoted by $\breve{H}^s(\breve{\Omega})$. We define, as in section (1.4), the radial component v_r, the angular component v_θ, and the axial component v_z of the vector field $\breve{v}$. Then the following proposition holds:

Proposition 2. *Let s be a positive integer number. The mapping: $\breve{v} \to (v_r, v_\theta, v_z)$ is well defined and one-to-one from $\breve{H}^s(\breve{\Omega})$ onto the product space $H^s_-(\Omega) \times H^s_-(\Omega) \times H^s_+(\Omega)$ where the space $H^s_+(\Omega)$ is defined in proposition* (1) *and the space $H^s_-(\Omega)$ is defined as follows:*

- *If s is not an odd integer,*

$$H^s_-(\Omega) = \left\{ w \in H^s_1(\Omega); \quad \partial_r^{2j} w|_{\Gamma_0} = 0, \quad 0 \le j \le \frac{s-1}{2} \right\}; \tag{1.18}$$

- *if s is an odd integer,*

$$H^s_-(\Omega) = \big\{ w \in H^s_1(\Omega); \quad \partial_r^{2j} w|_{\Gamma_0} = 0, \quad 0 \le j \le \frac{s-1}{2}, \\ \text{and} \quad \partial_r^{s-1} w \in L^2_{-1}(\Omega) \big\}. \tag{1.19}$$

The proof of these theorems may be found in [2].

1.6 Special Case

From these results we can derive the special case $s = 1$ which we need in chapter 3 to write the variational formulation.

$\breve{H}^1(\breve{\Omega})$ is the space of functions in $H^1_1(\breve{\Omega})$ which are invariant under rotation. According to the previous propositions, the space $H^1_+(\Omega)$ coincides with $H^1_1(\Omega)$. And $H^1_-(\Omega)$ is the space of functions w in $H^1_1(\breve{\Omega})$ such that $w|_{\Gamma_0} = 0$ and $w \in L^2_{-1}(\Omega)$.

To take boundary conditions into account, we must introduce the subspace of functions in $H^1_-(\Omega)$ which vanish on a certain part Γ_1 of the boundary of Ω which is not on the axis:

$$H^1_{-,0}(\Omega) = \left\{ v \in H^1_-(\Omega); \quad v = 0 \text{ on } \Gamma_1 \right\} \tag{1.20}$$

We define in the same way the subspace of functions in $H^1_+(\Omega)$ which vanish on a certain part Γ_1 of the boundary of Ω which is not on the axis:

$$H^1_{+,0}(\Omega) = \left\{ v \in H^1_+(\Omega); \quad v = 0 \text{ on } \Gamma_1 \right\} \tag{1.21}$$

Chapter 2

Equations

The intention for the numerical simulation presented in this work is to provide profiles for concentration, temperature, density and velocity fields. The equations governing chemical reactive flows are based on the compressible formulation of the Navier-Stokes equations, for the global behavior of the mixture flow, with additional convection-diffusion-reaction equations for the temperature and the chemical species. The equations are written in the primitive form, i.e. with the variables ρ (density) or p (pressure), u (velocity), T (temperature), and w (mass fractions). The set of coupled partial differential equations considered describes the convective motion of the fluid, the chemical reactions among the constituent species, and the diffusive transport processes such as thermal conduction and molecular diffusion. Its origin is the conservation of the physical variables ρ, ρu, ρE, ρw. While using these variables to write the equations, the formulation is said to be conservative. For smooth solutions, both formulations (conservative or primitive) are equivalent. In many applications, the formulation with primitive variables has the advantage of simpler boundary conditions and determination of transport coefficients (most of them are given as functions of the primitive variables).

2.1 Navier-Stokes Equations

The most general description of a fluid flow is obtained from the full system of Navier-Stokes equations. These are obtained by writing the mass and momentum conservation. For multicomponent flows, they describe the evolution in time and space of the density and velocity of the whole mixture, i.e. averaged quantities for the global flow. They are the following:

- *Mass conservation* : The law of mass conservation is a general statement of kinematic nature. It is independent of the nature of the fluid or of

the forces acting upon it. It expresses the empirical fact that, in a fluid system, mass cannot disappear from the system nor be created. The mass conservation equation is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\, u) = 0, \tag{2.1}$$

with ρ the density of the fluid, which could not be considered as constant in the case of multicomponent flows, even in the case of low-Mach-number flows, since the mixture is not usually homogeneous. u is the velocity of the flow.

- *Momentum conservation* : The sources for the variation of momentum in a physical system are the forces acting on it. These forces consist of the external volume forces f_e and the internal forces f_i. The latter are dependent on the nature of the fluid considered, and result from the assumptions made about the properties of the internal deformations within the fluid and their relation to the internal stresses. We will assume that the fluid is Newtonian, and therefore the total internal stresses $\overline{\overline{\sigma}}$ are taken to be

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}}, \tag{2.2}$$

where $\overline{\overline{I}}$ is the unit tensor and p the isotropic pressure. $\overline{\overline{\tau}}$ is the viscous shear stress tensor. With the exception of very high temperatures or pressures, the stress tensor for Newtonian fluids has the following form (see [26]):

$$\overline{\overline{\tau}} = \mu \left[\nabla u + (\nabla u)^T - \frac{2}{3}(\nabla \cdot u)\,\overline{\overline{I}} \right], \tag{2.3}$$

where μ is the dynamic viscosity of the fluid. In the case of multicomponent flows, it is a function of the partial viscosities and mole fraction of each species (see section 2.4).

The equation of motion then becomes

$$\rho \frac{\partial u}{\partial t} + \rho\,(u \cdot \nabla)\, u + \nabla p - \nabla \cdot \overline{\overline{\tau}} = \rho\, f_v, \tag{2.4}$$

with f_v the external volume forces.

2.2 Energy Conservation

The profile of temperature of the multispecies flow can be obtained through energy conservation. The energy content of a system is measured by its internal

energy per unit mass e. This internal energy is a state variable of a system and hence its variation during a thermodynamical transformation depends only on the final and initial states. In a fluid the total energy to be considered in the conservation equation is the sum of its internal energy e and its kinetic energy per unit mass $u^2/2$. The first law of thermodynamics states that the sources for the variation of the total energy are the work of the forces acting on the system plus the heat transmitted to this system. A distinction has to be made between the surface and volume sources. The volume sources are the sum of the work of the volume forces f. Hence we have, $Q_v = \rho\, f \cdot u$. The surface sources are the result of the work done on the fluid by the internal shear stresses acting on the surface of the volume considering that there are no surface heat sources:

$$Q_s = \overline{\overline{\sigma}} \cdot u = -p\, u + \overline{\overline{\tau}} \cdot u. \tag{2.5}$$

The diffusive flux q of heat due to molecular thermal conduction is given by the Fourier's law of heat conduction

$$q = -\lambda\, \nabla T, \tag{2.6}$$

with λ the thermal conductivity coefficient and T the temperature.

Writing the conservation of the total energy and considering the mass and momentum conservation equation as described in [44] or [26], we obtain

$$\rho\, \frac{de}{dt} + p\, \nabla \cdot u = \overline{\overline{\tau}} : \nabla u + \nabla \cdot (\lambda\, \nabla T), \tag{2.7}$$

with $\frac{de}{dt} = \frac{\partial e}{\partial t} + u \cdot \nabla e$ the total derivative of the intern energy according to time.

We define the specific enthalpy as

$$h = e + \frac{p}{\rho} \tag{2.8}$$

For an ideal gas (see Section 2.6, [58]), the enthalpy is a function of the temperature T and gas chemical state which can be represented by the mass fraction of each component $w = (w_i)_{i=1,\dots,n_s}$, with n_s the number of species in the mixture. The total variation of enthalpy for an ideal gas can be then expressed as follow:

$$dh = \left(\frac{\partial h}{\partial T}\right)_{p,w} dT + \sum_{i=1}^{n_s} \left(\frac{\partial h}{\partial w_i}\right)_{p,T} dw_i. \tag{2.9}$$

By definition the variation of enthalpy according to the temperature at constant pressure and chemical state is called c_p, specific heat capacity:

$$c_p = \left(\frac{\partial h}{\partial T}\right)_{p,w}. \tag{2.10}$$

We derive the total variation of internal energy:

$$de = c_p\, dT + \frac{p}{\rho^2}\, d\rho - \frac{1}{\rho}\, dp + \sum_{i=1}^{n_s} \left(\frac{\partial h}{\partial w_i}\right)_{p,T} dw_i. \tag{2.11}$$

Using the continuity equation (2.1), it yields

$$\rho \frac{de}{dt} = \rho\, c_p \frac{dT}{dt} + p\, \nabla \cdot u - \frac{dp}{dt} + \sum_{i=1}^{n_s} \left(\frac{\partial h}{\partial w_i}\right)_{p,T} \frac{dw_i}{dt} \tag{2.12}$$

Since h, the averaged enthalpy of the mixture considered as an ideal gas (see [58]), fulfills the relation

$$h = \sum_{i=1}^{n_s} h_i\ w_i, \tag{2.13}$$

with h_i the specific enthalpy of species i, equation (2.12) can be written as follow:

$$\rho \frac{de}{dt} = -\frac{dp}{dt} + p\, \nabla \cdot u + \rho\, c_p \frac{dT}{dt} + \sum_{i=1}^{n_s} h_i \frac{dw_i}{dt}. \tag{2.14}$$

The total time derivative of w_i can be expressed with a diffusion and a reaction terms (cf. Section 2.3 for the characteristics of these terms). This result together with equation (2.7) leads to an equation which describes the temperature evolution:

$$\rho\, c_p \frac{dT}{dt} = \frac{dp}{dt} + \overline{\overline{\tau}} : \nabla u + \nabla \cdot (\lambda\, \nabla T) + \sum_{i=1}^{n_s} h_i\ [\nabla \cdot j_i - f_i(T, w)]. \tag{2.15}$$

We use a simplified form of this equation because several terms may usually be neglected. Since we consider only flows at low-Mach number, the energy source due to internal stresses can be neglected. We are interested in this work in low pressure flow reactor. For such flows the pressure is considered as quasi-constant in time and space. Therefore we do not take into account in the following the pressure variation term in this equation. Moreover the term $\sum_i h_i\, \nabla \cdot j_i$, which represents the diffusion of species with different enthalpies,

is usually omitted, considering that the partial enthalpies h_i are nearly identical. Taking these simplifications into account, the equation for temperature becomes

$$\rho\, c_p \frac{\partial T}{\partial t} + \rho\, c_p\, u \cdot \nabla T - \nabla \cdot (\lambda\, \nabla T) = f_T(T, w). \tag{2.16}$$

The coefficients c_p and λ are the specific heat capacity at constant pressure and the heat conductivity of the mixture, respectively. The source term f_T depends on the temperature and the chemical state. Let us denote by h_i the specific enthalpy of species i, and by c_{p_i} the specific heat capacity of species i. The source term is then

$$f_T(T, w) = -\sum_{i=1}^{n_s} h_i(T)\, f_i(T, w). \tag{2.17}$$

The enthalpy h_i of species i is given by

$$h_i(T) = h_{i,T^0} + \int_{T^0}^{T} c_{p,i}(T')\, dT', \tag{2.18}$$

with an enthalpy h_{i,T^0} for a reference temperature T^0. The partial heat capacity of species i is represented by $c_{p,i}$. The temperature dependence of these partial heat capacities is modelled empirically. A fourth order polynomial fit in T, with coefficients determined by experiments, is widely used in numerical computations:

$$c_{p,i}(T) = \sum_{j=0}^{k} \alpha_j\, T^j \qquad i = 1, \ldots, n_s. \tag{2.19}$$

We use the coefficients from data bases developed at the Sandia National Laboratories [36] for the computations in chapter 6.

The heat conductivity λ corresponds to an average value for the mixture according to the chemical state of the gas and is defined in Section 2.4.

The factors $f_i(T, w_j)$ are chemical production terms and are defined in the next section.

2.3 Species Mass Conservation

The evolution of the chemical state of the gas in multicomponent flows can be described with the mass conservation of each chemical species. These latter can be represented by their mass fraction or by their mole fraction. We present here the formulation in mass fractions w_i. Both formulations are equivalent

although the formulation with mole fractions leads to a slightly more complicated transport term, while the formulation with mass fractions leads to a slightly more complicated diffusion term. Another difference is found in the calculation of the Jacobian matrix of the resulting non-linear system. We refer here to Chapter 4 for more details. The mass conservation of each species can be written with the help of a diffusion flux j_i, a source term (creation or destruction) f_i and the convective transport of the species. For a mixture of n_s chemical species, the corresponding equations are

$$\rho \frac{\partial w_i}{\partial t} + \rho\,(u \cdot \nabla) w_i + \nabla \cdot j_i = f_i(w, T), \qquad i = 1, \ldots, n_s, \tag{2.20}$$

with w the vector of all mass fractions w_i, characterizing the chemical state, and T the temperature. The source term f_i depends on both the temperature and chemical state.

This section deals further with the non-linearities brought by the multispecies character of the flow. In some regions of the domain, the flow may be dominated by reaction source terms that couple all the chemical variables with each other as well as with the temperature. Also in regions where the chemical reactions are weak, the non-constant diffusion coefficients cause another non-linearity and a coupling between all the chemical equations.

2.3.1 Modelling of Chemical Reactions and Source Terms

For the description of the chemical conversion in the gas phase, the chemical mechanisms are made up of elementary reactions. An elementary reaction can be generally described by

$$\sum_{i=1}^{n_s} a_{ir}\,\chi_i \quad \xrightarrow{k_r} \quad \sum_{i=1}^{n_s} \tilde{a}_{ir}\,\chi i \quad , \tag{2.21}$$

where χ_i represents the ith species and k_r the reaction rate of the reaction number r. a_{ir} and $\tilde{a}_{ir}$ are the stoichiometric coefficients of species i respectively as educt and product in the reaction r. In order to conserve the mass, these coefficients must fulfill the equation

$$\sum_{i=1}^{n_s} M_i(\tilde{a}_{ir} - a_{ir}) = 0, \tag{2.22}$$

with M_i being the molar mass of species i. In each reaction r of the above type, up to three species are involved on each side. Therefore, only up to three coefficient a_{ir} do not vanish for each r.

The production rate for species i, denoted $\dot{w}_i$, is obtained by adding the participation of all the reactions considered to the creation or destruction of species i. Defining n_r as the total number of reactions,

$$\dot{w}_i(T,w) = \sum_{l=1}^{n_r} \left\{ (\tilde{a}_{il} - a_{il})\, k_l(T) \prod_{j=1}^{n_s} c_j^{a_{jl}}(w) \right\}, \tag{2.23}$$

with c_j the concentration of species j, given by

$$c_j = \frac{\rho\, w_j}{M_j}. \tag{2.24}$$

The chemical source terms for the species equations in mass fractions have the form

$$f_i(T,w) = M_i\, \dot{w}_i(T,w), \qquad i = 1, \ldots, n_s. \tag{2.25}$$

Due to the property (2.22) on the stoichiometric coefficients we conclude that the sum over all the n_s source terms vanishes:

$$\sum_{i=1}^{n_s} f_i = 0. \tag{2.26}$$

The dependence on temperature for the reaction rate is given by the following Arrhenius-law

$$k_r(T) = A_r\, T^{\beta_r}\, \exp\left(-\frac{E_{ar}}{RT}\right). \tag{2.27}$$

This law is empirically validated. The constants A_r, β_r and the activation energy E_{ar} are usually determined through experiments. R is the ideal gas constant.

2.3.2 Surface Reactions

The reaction model used in this work for surface reactions introduces a reaction probability γ (named s̈ticking coefficientf̈or particles in the gas phase which hit a wall surface (see [56] and [17] for more information about surface reactions and their modelization). These particles can react (recombination, decomposition) or diffuse further unchanged in the gas phase. We consider here the case of surface reactions in which there is only one gas-phase reactant. These reactions are described by the following scheme:

$$a_{jr}\, \chi_j \quad \xrightarrow{\gamma_r} \quad \sum_{i=1}^{n_s} \tilde{a}_{ir}\, \chi i, \qquad j = 1, \ldots, n_s. \tag{2.28}$$

The corresponding reaction rate per surface unit for species i over all the n_r^0 surface reactions is given by

$$\dot{w}_i^0(T, w) = \sum_{r=1}^{n_r^0} \left\{ \gamma_r \, \frac{1}{4} \sqrt{\frac{8RT}{\pi \, M_j}} \, c_j \, (\tilde{a}_{ir} - \delta_{ij} \, a_{jr}) \right\}, \tag{2.29}$$

j being the single educt species of the reaction r. In this wall reaction model, there is indeed exactly one educt species for each surface reaction.

The probability coefficients are taken to be

$$\gamma_r = a_r \, T^{b_r} \, \exp\left(-\frac{c_r}{RT}\right), \qquad r = 1, \ldots, n_r^0, \tag{2.30}$$

with a_r, b_r and c_r usually determined by experiments. One goal of these simulations is precisely to determine the value of wall decomposition probabilities by comparing numerical with experimental results. In our applications (see Chapter 6) we have considered only constant probability coefficients.

From a numerical point of view, we must be careful to correctly evaluate the surface as well as the gas-phase production terms. Numerical experiments showed us that a good local convergence in the reaction zones have to be reached in order to get an accurate solution. Indeed the production or destruction of species anywhere in the domain may have influence on the whole flow. Hence a convergence statement on the global residuum is generally not sufficient.

Since the surface reactions occur only locally on the walls, i.e. on some domain boundaries, the numerical contribution of these reactions to the residuum and jacobian matrix is only restricted to the edges corresponding to a wall, i.e. on a few one-dimensional elements (for two-dimensional computations). These source terms influence the boundary conditions at walls for the temperature and the species mass fractions (see Chapter 3). For the temperature, energy is given to or taken from the gas phase depending on whether the reactions have created or consumed energy. For the species boundary conditions, a balance between the diffusion flux at the wall and the species creation or destruction rates is considered.

The influence of the surface reaction terms on the flow is of importance even if their participation to the global residuum might be small (due to their local existence). The accuracy on the solution needed locally to resolve these terms reinforce the importance of the adaptive mesh-refinement process (cf. Chapter 5).

2.3.3 Transport Coefficients

Transport property evaluation plays an important and often time-consuming role in the computational modelling of gaseous multicomponent reacting flows.

Two approaches are mostly considered for evaluating transport coefficients. In a first approach, a direct numerical inversion of the transport linear systems derived from kinetic theory is considered. This strategy often becomes computationally expensive. In a second approach, an empirical average expression is used, which yields less accurate transport coefficients but allows to deal with complex reactive systems with smaller computational efforts.

The diffusion flux, $\nabla \cdot j_i$, in (2.20) can be written with the help of the species diffusion velocity V_i as

$$j_i = \rho\, w_i\, V_i, \qquad i = 1, \ldots, n_s, \tag{2.31}$$

the species diffusion velocities being defined by the kinetic theory of dilute polyatomic gas mixture (see [54]) as

$$V_i = \frac{1}{x_i\, \overline{M}} \sum_{j \neq i}^{n_s} M_j\, D_{ij}\, d_j - \frac{D_i^T}{\rho\, w_i}\, \frac{1}{T}\, \nabla T, \tag{2.32}$$

with D_{ij} the multicomponent diffusion coefficients (see [27]), D_i^T the thermal diffusion coefficients and d_i the diffusion driving force of the ith species. The vectors d_i incorporate the effects of various state-variable gradients and are given by

$$d_i = \nabla x_i + (x_i - w_i)\, \frac{\nabla p}{p}, \qquad \forall\, i = 1, \ldots, n_s. \tag{2.33}$$

x_i denotes the mole fraction of the ith species, M_i the species molar mass of the ith species and $\overline{M}$ the mean molar mass of the mixture, which depends for multicomponent flows on the mixture chemical state:

$$\frac{1}{\overline{M}} = \sum_i \frac{w_i}{M_i}. \tag{2.34}$$

The mass fractions w_i and mole fractions x_i are related as follows:

$$x_i = w_i\, \frac{\overline{M}}{M_i}.$$

Thus we see from equations (2.32) and (2.33) that the diffusion flux from the species mass conservation equation (2.20) is composed of three parts: mass diffusion (Fick's law) due to gradients in molar fractions, thermo-diffusion due to temperature gradients (Soret effect), and pressure diffusion due to pressure gradients.

It follows from the above equations that the detailed modelling of a polyatomic gas mixture requires the evaluation of its transport coefficients, i.e.

the multicomponent and the thermal diffusion coefficients. These coefficients are functions of the state of the mixture as given by the variables p, T, and $w_1, \ldots, w_{n_s}$. Their evaluation requires solving linear systems, referred to as the transport linear systems (for more details on this see [20] and [35]).

In order to reduce the computational effort, mixture-averaged formulations may be used, which allows to avoid solving linear systems. Mixture-averaged diffusion coefficients can be defined with the help of the multicomponent properties. By definition, in the mixture, the diffusion velocities are then related to the species gradients by a Fickian formula as

$$V_i = -\frac{1}{x_i} D_i \, d_i - \frac{D_i^T}{\rho \, w_i} \frac{1}{T} \nabla T, \qquad i = 1, \ldots, n_s. \tag{2.35}$$

The mixture diffusion coefficients (see [12]) are computed as

$$D_i = \frac{1 - x_i}{\sum_{j \neq i}^{n_s} x_j / \mathcal{D}_{ji}}, \qquad i = 1, \ldots, n_s, \tag{2.36}$$

with $\mathcal{D}_{ji}$ the binary diffusion coefficient of species pair (j, i) (see [27]). These coefficients are nearly proportional to the square-root of the temperature and inversely proportional to the pressure.

A potential problem with this expression is that it is not mathematically well-defined in the limit of the mixture becoming a pure species. Considering equation (2.36), this modelling is not able to handle the special case of pure species. Even though diffusion itself has no real meaning in the case of a pure species, a computer-program implementation should ensure that the diffusion coefficients behave reasonably and that the code does not "blow up" when the pure species condition is reached. To overcome this difficulty we always maintain a residual amount of each species. Specifically, we assume in the above formulas that

$$x_i = \hat{x}_i + \delta, \tag{2.37}$$

where $\hat{x}_i$ is the actual mole fraction and δ is a small number that is numerically insignificant compared to any mole fraction of interest, yet which is large enough in order to be represented in computer arithmetic. We have experienced reasonable numerical behavior considering $\delta = 10^{-12}$.

A further problem is that this latter diffusion model does not necessarily fulfill the mass conservation constraint which implies that the species diffusion velocities satisfy the mass conservation relation

$$\sum_{i=1}^{n_s} w_i \, V_i = 0. \tag{2.38}$$

This topic will be considered in Section 2.5 in more details.

Finally we have restricted in this work the diffusion flux to the fickian diffusion. As result we obtain the following species mass conservation equations:

$$\rho \frac{\partial w_i}{\partial t} + \rho\,(u \cdot \nabla) w_i + \nabla \cdot (\rho\, D_i\, \nabla w_i) - \nabla \cdot (\rho\, D_i \frac{w_i}{\overline{M}} \nabla \overline{M}) = f_i(w, T)\,, \qquad i = 1, \ldots, n_s. \tag{2.39}$$

2.4 Mixture-Averaged Flow Properties

Our objective in this section is to determine mixture properties from the pure species properties. In the case of viscosity and heat conductivity, we use the empirical laws given in [56]. The viscosity μ of a mixture can be modelled with an accuracy of approximately 10% by the partial viscosities μ_i and the mole fractions x_i of the species:

$$\mu(T, w) = \frac{1}{2} \left[\sum_{i=1}^{n_s} x_i\, \mu_i + \left(\sum_{i=1}^{n_s} \frac{x_i}{\mu_i} \right)^{-1} \right]. \tag{2.40}$$

The $\mu_i = \mu_i(T)$ are nearly proportional to the square-root of the temperature. We use a polynomial fit with coefficients determined by experiments [36]. The heat conductivity λ has a similar representation:

$$\lambda(T, w) = \frac{1}{2} \left[\sum_{i=1}^{n_s} x_i\, \lambda_i + \left(\sum_{i=1}^{n_s} \frac{x_i}{\lambda_i} \right)^{-1} \right], \tag{2.41}$$

with λ_i the partial heat conductivity, which are also calculated as a polynomial of the temperature.

2.5 Physical constraints

By definition, the sum over all mass fractions must be one and the mass conservation implies that the sum over the diffusive fluxes should vanish:

$$\sum_{i=1}^{n_s} w_i = 1\,, \qquad \sum_{i=1}^{n_s} j_i = 0. \tag{2.42}$$

Moreover each mass fraction w_i must, also by definition, have a value between zero and one:

$$0 \le w_i \le 1, \qquad \forall\, i = 1, \ldots, n_s. \tag{2.43}$$

Some care needs to be taken in using the mixture-averaged diffusion coefficients as described above. The mixture formulas are approximations and they are not constrained to require that the sum over all species diffusion fluxes is zero, i.e. condition (2.38) needs not be satisfied. Therefore, one must expect that applying these mixture diffusion relationships in the solution of a system of species conservation equations should lead to some non-conservation, i.e. the resultant mass fractions will not sum to one. Therefore one of a number of corrective actions must be invoked to ensure mass conservation.

One possible approach is to define a "conservation diffusion velocity" as recommended in [16]. In this approach it is assumed that the diffusion velocity vector is given as

$$V_k = \hat{V}_k + V_c, \tag{2.44}$$

where $\hat{V}_k$ is the ordinary diffusion velocity given by equation (2.35) and V_c is a constant correction factor (independent of species, but spatially varying) introduced to satisfy equation (2.38). The correction velocity is defined by

$$V_c = -\sum_{k=1}^{n_s} w_k \hat{V}_k. \tag{2.45}$$

An alternative is based on excluding the conservation equation for one species. Its mass fraction is then computed simply by subtracting the sum of the remaining mass fractions from unity. This is an attractive method for problems having one species that is always present in excess. A similar approach involves determining locally at each computational cell, which species is in excess. The diffusion velocity for that species is then computed to require satisfaction of equation (2.38).

But even though the complete multicomponent formulation is theoretically forced to conserve mass, and so should also be corrected methods for the simplified formulation, numerical implementations and resolution errors can cause some slight non-conservation. Depending on the numerical method, even slight inconsistencies can lead to difficulties. Therefore a third approach may be used that ensures (2.38) but also (2.43). This latter basic condition must absolutely be fulfilled to avoid inconsistencies with the physics and that the resolution method suffers computational inefficiencies or convergence failures. A correction can be made directly on the mass fractions $\hat{w}_i$ that are calculated with the mixture-averaged diffusion model. This model can deliver slightly negative or greater-than-one mass fractions. The correction is then

$$\tilde{w}_i = \begin{cases} 10^{-12} & \text{if } \hat{w}_i \leq 10^{-12}, \\ \hat{w}_i & \text{otherwise,} \end{cases}$$

$$w_i = \frac{\tilde{w}_i}{\sum_{k=1}^{n_s} \tilde{w}_k}.$$

This allows to avoid the pure species problem and leads to physically reasonable values for the mass fractions. Nevertheless the w_i obtained are not solution of the multicomponent-flow system anymore. One should ensure that this correction is not too strong according to the solution $\hat{w}_i$ obtained by the resolution of the system of partial differential equations. Therefore we may apply this method as complementary corrective measure to the methods described above since, in this case, we can be sure that the magnitude of this correction will be significantly smaller.

In this work only the latter correction is applied to the solution at every nonlinear step of the solving process (see Chapter 4). Numerical tests showed us that the other corrections did not have much influence on the solution for our application cases. The order of the correction in our tests was locally at most 10% on the species mass fractions.

2.6 Ideal Gas Law

Usually an algebraic equation of state for the mixture closes the system. In many instances a compressible fluid can be considered as a perfect gas, even if viscous effects are taken into account. The ideal gas law gives a relation between the pressure and the density:

$$p = \frac{\rho R T}{\overline{M}}, \tag{2.46}$$

where R is the universal gas constant and $\overline{M}$ the mean molar mass of the mixture. While considering the low-Mach-number approximation, the pressure which is to be found in this later state equation is the constant thermodynamical pressure p_{th}.

Defining $\gamma = c_p/c_v$, the speed of sound c is given by

$$c^2 = \left(\frac{\partial p}{\partial \rho}\right)_s = \frac{\gamma R T}{\overline{M}} = \frac{\gamma p}{\rho}, \tag{2.47}$$

We can then define the Mach number by

$$M = \frac{|u|}{c}. \tag{2.48}$$

For our applications, it is supposed to be small. For example in the flow reactor for the CARS experiment presented in Chapter 6, with a fluid velocity of 50 m/s, the Mach number is 0.018. Under a value of 0.3, the fluid may be considered as hydrodynamically incompressible. However in the case of multicomponent flows, this does not mean that the density of the flow is constant.

For ideal gases, the continuity equations can be rewritten in a form independent of the variable ρ. From the relation 2.46, dividing the equation 2.1 by ρ yields to the following form of the continuity equation:

$$\frac{1}{p}\frac{dp}{dt} + \frac{1}{\overline{M}}\frac{d\overline{M}}{dt} - \frac{1}{T}\frac{dT}{dt} + \nabla \cdot u = 0, \tag{2.49}$$

with the definition of the total derivative $\frac{d}{dt} = \frac{\partial}{\partial t} + u \cdot \nabla$.

In the following section we will see that the pressure term can be neglected for the pressure remains constant in first approximation. The continuity equation is finally

$$\frac{1}{\overline{M}}\frac{d\overline{M}}{dt} - \frac{1}{T}\frac{dT}{dt} + \nabla \cdot u = 0. \tag{2.50}$$

2.7 Low-Mach-Number approximation

In low-Mach-number flows, the pressure field can be split in two parts, one constant and the other variable in space and time. The first one is called the thermodynamical part and the second one the hydrodynamical part:

$$p = p_{th} + p_{hyd}. \tag{2.51}$$

The hydrodynamical part p_{hyd} is negligible according to the thermodynamical part p_{th}. Rewriting the ideal gas law with these conditions leads to an equation for the density:

$$\rho = \frac{\overline{M}\, p_{th}}{RT}. \tag{2.52}$$

This splitting has been used in many publications (see for instance [42], [39], [40]) and we sketch here the method which leads to it.

We must first write the governing conservation equations with non-dimensional variables, taking the Mach number into account. The Mach number used to make the variables dimensionless is evaluated at the initial state. For the sake of simplicity, we write here only the non-dimensional momentum equation:

$$\gamma M^2 \rho \frac{d\hat{u}}{dt} = -\nabla \hat{p} + \frac{\gamma M^2}{R_e} \nabla \cdot \hat{\tau}. \tag{2.53}$$

The $\hat{}$ means that the corresponding variable is in non-dimensional form. $Re = \frac{L\,\rho\,u}{\mu}$ is the Reynolds number of the flow (L is a characteristic length of the problem) and $\frac{d}{dt} = \frac{\partial}{\partial t} + \hat{u} \cdot \nabla$. Since the Mach number is small and since it appears in the equations as $\epsilon = \gamma M^2$, all the gas dynamic variables may be

expanded in terms of ϵ. That is, any variable $\zeta \in [\rho, u, p, T, w]$ can be expanded as follow:

$$\zeta(x,t) = \zeta_0(x,t) + \epsilon\,\zeta_1(x,t) + \epsilon^2\,\zeta_2(x,t) + O(\epsilon^3). \tag{2.54}$$

Considering the variable p and substituting into (2.53), the momentum equation reads

$$\epsilon\rho\,\frac{D\hat{u}}{Dt} = -\nabla\hat{p_0} - \epsilon\nabla\hat{p_1} - \epsilon^2\nabla\hat{p_2} + \frac{\epsilon}{R_e}\,\nabla\cdot\hat{\tau}. \tag{2.55}$$

Gathering terms that are independent of M, one finds $\nabla p_0 = 0$, which shows immediately that

$$p_0 = p_0(t) \tag{2.56}$$

This is the main result of the low Mach number approximation. The largest component of the pressure is constant throughout the field and changes only with time. p_0 is the thermodynamic pressure. The second component of the pressure appears in the ϵ-component of the expansion of the momentum equation:

$$\rho_0\,\frac{Du_0}{Dt} = -\nabla p_1 + \frac{1}{R_e}\,\nabla\cdot\tau_0. \tag{2.57}$$

p_1 is the hydrodynamic pressure and is generated to balance the changes in momentum within the flow field. Its contribution to the total pressure is restricted to ϵ.

2.8 Cylinder Coordinates

As we saw in Chapter 1, the operators in cylinder coordinates involve supplementary terms that are not to be found in cartesian coordinates. In this section we describe the equations discussed in the previous sections developed in cylinder coordinates and focus on these supplementary terms. Some information about generalized curvilinear coordinates can be found in [24] or [1], and about the Navier-Stokes equations in cylinder coordinates in [44].

2.8.1 The Stress Tensor

The stress tensor written in canonical form in Section 2.3 depends on the velocity-gradient tensor. Considering the symmetry condition, just as in Chapter 1, this latter tensor can be written in cylinder coordinates in the basis (e_r, e_θ, e_z):

$$\nabla u = \begin{pmatrix} \frac{\partial u_r}{\partial r} & 0 & \frac{\partial u_z}{\partial r} \\ 0 & \frac{u_r}{r} & 0 \\ \frac{\partial u_r}{\partial z} & 0 & \frac{\partial u_z}{\partial z} \end{pmatrix}$$

The stress tensor is

$$\overline{\overline{\sigma}} = \mu \left(\nabla u + \nabla^T u \right) - \left(\frac{2}{3} \mu \nabla \cdot u + p \right) \overline{\overline{I}}.$$

Defining a generalized pressure by

$$p^* = \frac{2}{3} \mu \nabla \cdot u + p \tag{2.58}$$

and again taking into account the symmetry condition, the stress tensor becomes

$$\overline{\overline{\sigma}} = \begin{pmatrix} 2\mu u_r - p^* & 0 & \mu (w_r + u_z) \\ 0 & 2\mu \frac{u}{r} - p^* & 0 \\ \mu (w_r + u_z) & 0 & 2\mu w_z - p^* \end{pmatrix}.$$

In the cylinder system of coordinates, which is defined in this work with the orthonormal base (e_r, e_θ, e_z), the first and third components of the divergence of a symmetric tensor $\overline{\overline{t}}$ of second order is :

$$\begin{aligned} (\nabla \cdot \overline{\overline{t}})_1 &= \frac{1}{r} t_{11} + \frac{\partial t_{11}}{\partial r} + \frac{1}{r} \frac{\partial t_{12}}{\partial \theta} + \frac{\partial t_{13}}{\partial z} - \frac{t_{22}}{r}, \\ (\nabla \cdot \overline{\overline{t}})_3 &= \frac{\partial t_{33}}{\partial z} + \frac{1}{r} t_{31} + \frac{\partial t_{31}}{\partial r} + \frac{\partial t_{32}}{\partial \theta}. \end{aligned}$$

Thus the first component of the divergence of the stress tensor in cylinder coordinates with axial symmetry is

$$(\nabla \cdot \overline{\overline{\sigma}})_1 = \nabla \cdot (\mu \nabla u_r) + \mu \frac{\partial}{\partial r} (\nabla \cdot u) + \nabla \mu \cdot \frac{\partial u}{\partial r} - \mu \frac{u}{r^2} - \frac{\partial p^*}{\partial r}.$$

The second component of the divergence of the stress tensor vanishes, due to axial symmetry. It remains the third component:

$$(\nabla \cdot \overline{\overline{\sigma}})_3 = +\nabla \cdot (\mu \nabla u_z) + \mu \frac{\partial}{\partial z} (\nabla \cdot u) + \nabla \mu \cdot \frac{\partial u}{\partial z} - \frac{\partial p^*}{\partial z}.$$

One has to remember that the divergence in cylinder coordinates is

$$\nabla \cdot u = \frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial u_z}{\partial z}.$$

2.8.2 The Equations in Cylinder Coordinates

Additional terms appear in cylinder coordinates for the vectorial equations. Taking into account the results of the previous chapter, we can then write the momentum conservation equations (2.4) in cylinder coordinates. Writing the velocity in cylinder coordinates $u = (u_r, u_z)$, the system of equations is

$$\frac{1}{\overline{M}} \frac{d\overline{M}}{dt} - \frac{1}{T} \frac{dT}{dt} + \nabla \cdot u = 0, \tag{2.59}$$

$$\begin{aligned} \rho \frac{\partial u_r}{\partial t} + \rho \left(u \cdot \nabla\right) u_r - \nabla \cdot \left(\mu \nabla u_r\right) \\ - \mu \frac{\partial}{\partial r}(\nabla \cdot u) - \nabla \mu \cdot \frac{\partial u}{\partial r} + \mu \frac{u_r}{r^2} + \frac{\partial p^*}{\partial r} = \rho f_v^{(r)}, \end{aligned} \tag{2.60}$$

$$\begin{aligned} \rho \frac{\partial u_z}{\partial t} + \rho \left(u \cdot \nabla\right) u_z - \nabla \cdot \left(\mu \nabla u_z\right) \\ - \mu \frac{\partial}{\partial z}(\nabla \cdot u) - \nabla \mu \cdot \frac{\partial u}{\partial z} + \frac{\partial p^*}{\partial z} = \rho f_v^{(z)}, \end{aligned} \tag{2.61}$$

$$\rho \, c_p \frac{\partial T}{\partial t} + \rho \, c_p \left(u \cdot \nabla\right) T + \nabla \cdot \left(\lambda \nabla T\right) = f_T(w, T), \tag{2.62}$$

$$\rho \frac{\partial w_i}{\partial t} + \rho \left(u \cdot \nabla\right) w_i + \nabla \cdot j_i = f_i(w, T), \qquad \forall \, i = 1, \ldots, n_s. \tag{2.63}$$

Chapter 3

Discretization

This chapter presents and analyzes a finite element scheme for simulating the three major processes in reactive flows: chemical reactions, diffusion and convection.

The methods used in simulation of reactive flows are usually based on either finite differences for its simple implementation and mathematical background as in [3] and [46] or finite volumes which are a range of methods widely spread in the engineering field (see [19] for a study of some schemes). The method used in this work is based on conforming "Q_1/Q_1" Galerkin finite elements. The basics on the mathematical theory of finite element methods used in this work can be found in the books of Johnson [30] and Brenner/Scott [15].

The choice of a finite element method is principally motivated by the flexibility it offers with respect to adaptive mesh refinement. It can be coupled with error control based on a posteriori error estimates provided by the orthogonality property of the method as explained in Chapter 5. Thus accuracy for some physical quantities which are to be precisely known can be guaranteed.

In this chapter, we discuss the discretization of the unsteady and steady multispecies low-Mach-number compressible Navier-Stokes equations with advection-diffusion-reaction equations for chemical species. The aim is to simulate quasi-stationary low-Mach-number flows in flow reactors.

The application of conforming finite elements to the incompressible or compressible Navier-Stokes equations is standard (see for instance [4], [49] or [10]). Extensions to thermally coupled flows or multispecies reactive flows have also been developed in the last decade. The reader can find some examples in [38], [50], [37] or in the more recent work [13].

In the case of axisymmetric flows, the three-dimensional problem can be transformed to a two-dimensional one (see chapter 1). Although such a transformation reduces the computation time, we have to deal with the following problems:

- The differential operators in the axisymmetric formulation have singularities on the axis. We have to work with weighted Sobolev spaces (see chapter 1 or [41], [11]).
- The radial and the axial components of the velocity belong to different Sobolev spaces.

We discretize the equations modelling axisymmetric multispecies reactive flows with *stabilized* Q1 elements for all variables. The equations considered have indeed two different sources of difficulties that a stable discretization must overcome.

The first difficulty is the velocity-pressure coupling brought by the saddle-point structure of the Stokes system of equations. It is well known that this approach does not lead to a stable discretization unless the finite dimensional spaces fulfill the "inf-sup" condition (see [25]). In order to get a stable discretization, we add weighted mesh-dependent least-squares terms to the standard Galerkin formulation as proposed by Hughes et al. in [29].

The second kind of instability occurs in the case of high Reynolds numbers, when the system becomes convection-dominated. The standard Galerkin method for convection dominated problems produce approximations which contain "spurious" oscillations in case of non-smooth exact solutions. The oscillations result from a lack of stability of the method. A standard finite element technique to deal with scalar convection-diffusion equations is the streamline diffusion method (see [30], [60]). The stabilization is done by adding further weighted least-square terms to the discrete equations. The stabilizing perturbation term can be physically thought as a numerical diffusion term in the direction of the streamlines. This modification enhances stability without a strong effect on the accuracy because the terms added are based on the residual.

3.1 Definitions

Using the notations of chapter 1, we denote the inner scalar product in $L^2_1(\Omega)$ by

$$(u, v) = \int_\Omega u(r, z)\, v(r, z)\, r\, dr\, dz \tag{3.1}$$

We also denote by X the solution vector of the system presented in the next section, that is

$$X = [u_r, u_z, p^*, T]^T. \tag{3.2}$$

For simplicity, in the following the notation p will replace p^*. We will call it the generalized pressure.

3.2 The Variational Formulation

In this section, we consider the continuity equation (2.59) as well as the momentum equations (2.60) and (2.61). We also consider a diffusion-convection-reaction equation modelling the evolution equations of temperature and species mass fractions. It can be written as follow:

$$\rho \frac{\partial T}{\partial t} + \rho \left(u \cdot \nabla\right) T + \nabla \cdot (\lambda \nabla T) = f_T(w, T). \tag{3.3}$$

The variational formulation of the resulting system is obtained by writing the equations in weak form and integrating by parts. We define the energy forms for each equation:

- The continuity equation:

$$a_1(X, q) = \left(\frac{1}{\bar{M}} \frac{d\bar{M}}{dt}, q\right) - \left(\frac{1}{T} \frac{dT}{dt}, q\right) + (\nabla \cdot u, q), \tag{3.4}$$

- The first momentum conservation equation:

$$\begin{aligned} a_2(X, \varphi) =& (\rho \frac{du_r}{dt}, \varphi) - \left(p, \frac{\partial \varphi}{\partial r} + \frac{\varphi}{r}\right) + (\mu \nabla u_r, \nabla \varphi) \\ &+ \left(\mu \frac{u_r}{r^2}, \varphi\right) - \left(\mu \frac{\partial}{\partial r}(\nabla \cdot u), \varphi\right) - \left(\nabla \mu \cdot \frac{\partial u}{\partial r}, \varphi\right), \end{aligned} \tag{3.5}$$

- The second momentum conservation equation:

$$\begin{aligned} a_3(X, \psi) = & \left(\rho \frac{du_z}{dt}, \psi\right) - \left(p, \frac{\partial \psi}{\partial z}\right) + (\mu \nabla u_z, \nabla \psi) \\ & - \left(\mu \frac{\partial}{\partial z}(\nabla \cdot u), \psi\right) - \left(\nabla \mu \cdot \frac{\partial u}{\partial z}, \psi\right), \end{aligned} \tag{3.6}$$

- The energy or species-mass conservation equations

$$a_4(X, \phi) = \left(\rho \frac{dT}{dt}, \phi\right) + (\lambda \nabla T, \nabla \phi), \tag{3.7}$$

with $\frac{d}{dt} = \frac{\partial}{\partial t} + u \cdot \nabla$ the total time derivative.

Using the notations of Chapter 1, we denote by $V_- = H^1_{-,0}$ and $V_+ = H^1_{+,0}$ the spaces for the velocity field, by $Q = L^2_1(\Omega)$ the space for the pressure and by $S = H^1_{+,0}(\Omega)$ the space for the temperature and mass fractions.

We define the vectorial energy form corresponding to the whole system by

$$a(X,\xi) = a_1(X,q) + a_2(X,\varphi) + a_3(X,\psi) + a_4(X,\phi), \tag{3.8}$$

with the vectorial test function $\xi = [q, \varphi, \psi, \phi]^T \in V = V_- \times V_+ \times Q \times S$.
The right hand side vector f of the system is

$$f = [0; f_v^{(r)}; f_v^{(z)}; f_T]^T. \tag{3.9}$$

The variational formulation consists then in finding $X \in V = V_- \times V_+ \times Q \times S$ such that

$$a(X,\xi) = (f,\xi) \qquad \forall\, \xi \in V \tag{3.10}$$

holds.

3.3 Boundary Conditions

3.3.1 General Boundary Conditions

For this problem, the boundary conditions are on the four different boundaries the following:

$$\left.\begin{aligned} \text{symmetry on } \Gamma_0 &: u_r = 0, \\ \text{inflow on } \Gamma_1 &: u = u_0, \quad T = T_0, \\ \text{wall on } \Gamma_2 &: u = 0, \quad \frac{\partial T}{\partial n} = f_T^0, \\ \text{outflow on } \Gamma_3 &: \mu \frac{\partial u}{\partial n} - p \cdot n = 0, \quad \frac{\partial T}{\partial n} = 0, \end{aligned}\right\} \tag{3.11}$$

where $\partial\Omega = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$, and f_T^0 is a surface source terms. Since the integration is weighted by the factor r, the natural boundary condition on the symmetry boundary Γ_0 vanishes. Nevertheless, according to the proposition 2 of Section 1.5.1, the radial velocity u_r is zero on the symmetry line Γ_0. The Neumann or mixed conditions on the other domain boundaries are obtained through the natural boundary conditions supplied by the variational formulation.

3.3.2 Supplementary Conditions

Other conditions coming directly from the equations for a steady-state solution can be taken into account.

A condition on u_z can be found in the case of a steady-state solution through the continuity equation (2.1) which leads to the relation

$$\int_{\Gamma_3} \rho\, u \cdot n \, d\Gamma + \int_{\Gamma_0} \rho\, u \cdot n \, d\Gamma + \int_{\Gamma_1} \rho\, u \cdot n \, d\Gamma = 0, \tag{3.12}$$

since the velocity is zero on the wall boundary Γ_2. On the symmetry line, the normal n is in the radial direction. For the outflow and inflow it is in the axial direction. At the symmetry line the integration weight r is zero. We then obtain

$$\int_{\Gamma_3} \rho\, u_z \, r \, dr = \int_{\Gamma_1} \rho\, u_z \, r \, dr. \tag{3.13}$$

The integral upon the inflow boundary is known for u_z which is set by a Dirichlet condition. It physically means that the mass that flows into the tube goes out.

Again for the outflow, a condition on the generalized pressure can be found by considering the natural boundary condition on the outflow boundary. The relation

$$\int_{\Gamma_3} (\mu \frac{\partial u_z}{\partial z} - p)\, r \, dr = 0 \tag{3.14}$$

is completed by the mass conservation property

$$\frac{\partial u_z}{\partial z} = -\frac{\partial (r\, u_r)}{\partial r}. \tag{3.15}$$

The continuity equation in strong formulation may be written in this way only if the density ρ remains constant. This should be the case on the outflow boundary. Therefore, to be sure that this relation is respected, we must assume that no chemical reaction take place on the outflow and that the mixing process is complete. If additionally the viscosity μ is also constant on the outflow (the same hypothesis should lead to such a situation), a direct integration yields

$$\int_{\Gamma_3} p \, r \, dr = 0, \tag{3.16}$$

since $r = 0$ on the symmetry line and $u_r = 0$ on the wall.

Another condition can be derived from the continuity equation at least in the case of a strong solution of equation (2.1). We must here consider the three-dimensional domain and remember that the symmetry boundary corresponds to the middle of the flow reactor. Thus if the solution is smooth enough, the mass conservation in strong form may be fulfilled, particularly in the middle of the tube where no singularity is found. Lets consider the following integral:

$$I_0 = \int_{\Gamma_0} \phi \, \nabla \cdot (\rho\, u)\, r \, dz = 0 \qquad \forall \phi \in L_1^2(\Gamma_0), \tag{3.17}$$

if the above hypothesis is fulfilled. This integral can be decomposed as follows

$$I_0 = \int_{\Gamma_0} r\,\phi\,\frac{\partial \rho\, u_r}{\partial r}\,dz + \int_{\Gamma_0} r\,\phi\,\frac{\partial \rho\, u_z}{\partial z}\,dz + \int_{\Gamma_0} r\,\phi\,\frac{u_r}{r}\,dz. \tag{3.18}$$

We have $I_0 = 0$, since $r = 0$ on this boundary. The first and second integrals of the right hand side are zero for the same reason. We can then deduce that

$$\int_{\Gamma_0} \phi\, u_r\, dz = 0 \qquad \forall \phi \in L^2_1(\Gamma_0), \tag{3.19}$$

which means that the radial component of the velocity is zero on the symmetry line. Therefore, if the above conditions are fulfilled, no Dirichlet boundary condition needs to be set on the symmetry line for the radial velocity.

3.3.3 Symmetry Boundary Condition

Depending on the space which the three-dimensional solution belongs to, the mapping between the three-dimensional and the two-dimensional problems can also lead to supplementary boundary conditions which are contained within the finite element spaces considered. Once again according to Proposition 2, the solution may indeed fulfill supplementary conditions on the symmetry boundary if it has enough regularity. In the case of a three-dimensional solution which belongs to $\breve{H}^2(\breve{\Omega})$, with regard to the definition of the space H^2_+, the normal derivative to the symmetry boundary of the solution components u_r, p and T vanishes. If the solution is sought in H^1_+, these boundary conditions on the symmetry line for the variable cited above are not valid anymore.

3.4 Discretization in Space

Starting from the variational formulation (3.10) supplemented by the boundary conditions (3.11), we choose the finite element subspaces $V_h \subset V$ to obtain the standard Galerkin discretization. We consider in this work an approximation by piecewise bi-linear shape functions on meshes $\mathbb{T}_h = \{K\}$ made of quadrilaterals and satisfying the usual regularity conditions (quasi-uniformity). The width of the mesh $\mathbb{T}_h$ is characterized in terms of the mesh size function $h = h_{max} = \max_{K\in\mathbb{T}_h}(h_K)$ with $h_K = \mathrm{diam}(K)$. In order to ease the refinement and coarsening processes, one hanging node per element edge is allowed.

Considering the vectorial energy form defined in (3.8), the discrete solution $X_h \in V_h$ is determined by the equation

$$a(X_h, \xi_h) = (f, \xi_h) \qquad \forall\, \xi_h \in V_h, \tag{3.20}$$

with V_h the set of piecewise bi-linear shape functions on $\mathbb{T}_h$, which is a subset of V defined in (3.8).

3.5 Stabilization

As mentioned before, the standard Galerkin discretization obtained for the Navier-Stokes equations does not yield a stable algorithm unless the spaces fulfill the discrete LBB-condition (cf. [14], [25]). This condition is a compatibility condition for the velocity-pressure coupling. An alternative, presented by Hughes et al. in [29], is to modify the discrete bilinear form in order to get a stable discretization.

Moreover the convection terms in any convection-diffusion equation lead to supplementary instabilities. Non-physical oscillations can appear in numerical solutions of the Navier-Stokes equations. Therefore the approach is modified. The stability of the Galerkin finite element method has to be improved, but it has to be done carefully since additional stability is often obtained at the price of decreased accuracy. We consider two ways of enhancing the stability of the standard Galerkin finite element method:

- introduction of weighted least-squares terms;
- introduction of artificial viscosity based on the residual.

We refer to the Galerkin finite element method with these modifications as the streamline diffusion method. The first modification adds stability through least squares control of the residual and the second modification adds stability by the introduction of an elliptic term with the size of the diffusion coefficient depending on the residual with the effect that diffusion is added where the residual is large, i.e. typically where the solution is non-smooth. Both modifications enhance stability without a strong effect on the accuracy because both modifications use the residual.

3.5.1 The Galerkin-Least-Squares Method

Let a be a linear operator on a vector space V with inner product (.,.) and corresponding norm $\|.\|$. Typically, A is a convection-diffusion differential operator, and (.,.) is the L_2 inner scalar product over some domain Ω. We consider the linear problem of finding u such that

$$Au = f, \tag{3.21}$$

for which the variational formulation reads:

$$\begin{aligned} &\textit{Find } u \in V \textit{ such that} \\ &(Au, \varphi) = (f, \varphi) \qquad \forall \varphi \in V. \end{aligned} \tag{3.22}$$

The least squares method for (3.21) is to find $u \in V$ that minimizes the residual over V, that is

$$\|Au - f\|^2 = \min_{v \in V} \|Av - f\|^2. \tag{3.23}$$

This is a convex minimization problem (because it is quadratic) and the solution is characterized by

$$(Au, A\varphi) = (f, A\varphi) \qquad \forall \varphi \in V \tag{3.24}$$

The problem is symmetric positive definite (A is considered regular), and thus can be solved without difficulties. Equation (3.22) may be more difficult to solve, but may be more accurate than equation (3.24), for the test-function space used in the second problem may contain less information (for instance if A contains a differential operator and the ansatz functions are linear). Therefore a combination of the 2 systems is taken. The resulting system should still be accurate enough but easier to solve.

We now formulate the Galerkin-least-squares finite element method for (3.21) by taking a weighted formulation of (3.22) and (3.24):

$$\begin{aligned} &\textit{Find } u \in V \textit{ such that} \\ &\qquad (Au, \varphi) + (Au, \delta A\varphi) = (f, \varphi) + (f, \delta A\varphi) \qquad \forall \varphi \in V. \end{aligned} \tag{3.25}$$

We can alternatively formulate the Galerkin-least-squares method as a Petrov-Galerkin method, which is a Galerkin method with the space of test functions being different from the space of trial functions. In our case the test functions have the form $\varphi + \delta A\varphi$ with $\varphi \in V$.

3.5.2 Artificial Viscosity

Adding artificial viscosity yields the streamline diffusion method in the form:

$$\begin{aligned} &\textit{Find } u \in V \textit{ such that} \\ &\qquad (Au, \varphi + \delta A\varphi) + (\epsilon \nabla u, \nabla \varphi) = (f, \varphi + \delta A\varphi) \qquad \forall \varphi \in V, \end{aligned} \tag{3.26}$$

where ϵ is the artificial viscosity. It is defined in the discretization process in terms of the residual $R(u) = Au - f$ through

$$\epsilon = \gamma\, h^2\, \|R(u)\|, \tag{3.27}$$

with γ a positive constant to be chosen, and h the local mesh size.

3.5.3 Application to Scalar Convection-Diffusion Equations

Applying this stabilization to any scalar convection-diffusion equation of type (3.7), the streamline diffusion method introduces a stabilizing term by the use of an additional test function of the form $\delta\, u \cdot \nabla\phi$. The introduction of the additional least-squares terms is done in an element-wise fashion. This implies that the weighting parameter δ depends on the element. It will be subscribed correspondingly. We obtain the following equation:

$$\begin{aligned}\left(\rho\,\frac{dT_h}{dt},\phi_h\right) + (\lambda\,\nabla T_h, \nabla\phi_h) + \sum_{K\in\mathbb{T}_h}\left(\rho\,\frac{dT_h}{dt} - \nabla\cdot(\lambda\,\nabla T_h), \delta_K\, u_h\cdot\nabla\phi_h\right)\\ = (f_T,\phi_h) + \sum_{K\in\mathbb{T}_h}(f, \delta_K\, u_h\cdot\nabla\phi_h) \qquad \forall\,\phi_h\in S_h.\end{aligned} \tag{3.28}$$

The least-squares terms correspond to the addition of viscosity in the direction of the streamline. The method is consistent in the sense that the stabilizing terms vanish for a strong solution of system (2.59) - (2.63). We discuss later the choice of the parameter δ_K. The introduction of artificial viscosity is straightforward. However it should be brought into operation only if additional cross-wind diffusion is really necessary to avoid oscillations. In many cases the least-squares terms are sufficient. The discretization still remains of second order (see [60]) and stable for a wide range of diffusion parameters.

3.5.4 Pressure Stabilization

The spaces Q_h and V_h used in this work are based on piecewise bi-linear functions on quadrilateral elements, namely Q1/Q1-elements. For these spaces the LBB-condition is not satisfied (see [14], [25]). The stabilization of the Navier-Stokes equations for our discretization with bi-linear conforming elements is done in the same way as the streamline diffusion, i.e. by adding mesh-dependent least-squares terms to the Galerkin formulation. The discretization then reads as follow:

$$\left.\begin{aligned}(\nabla\cdot(\rho\,u_h), q_h) + \sum_{K\in\mathbb{T}_h}(\,R_u,\,\alpha_K\,\nabla q_h\,)_K \\ = \sum_{K\in\mathbb{T}_h}(f_v, \alpha_K\,\nabla q_h)_K \qquad \forall\, q_h\in Q_h,\\ \left(\rho\,\frac{du_h}{dt},\varphi_h\right) + \left(\mu\,\frac{u_{r,h}}{r^2},\varphi_{r,h}\right) + (\mu\,\nabla u_h, \nabla\varphi_h)\\ -(\mu\,\nabla(\nabla\cdot u_h),\varphi_h) - (\nabla\mu\cdot\nabla u_h,\varphi_h)\\ -(p_h,\nabla\varphi_h) = (f_v,\varphi_h) \qquad \forall\,\varphi_h\in V_-^h\times V_+^h,\end{aligned}\right\} \tag{3.29}$$

where

$$R_u = \rho \frac{du_h}{dt} - \nabla \cdot (\mu \nabla u_h) + \nabla p_h - \nabla \mu \cdot \nabla u_h \\ -\mu \nabla(\nabla \cdot u_h) + \frac{\mu}{r^2} \begin{pmatrix} u_{r,h} \\ 0 \end{pmatrix}, \tag{3.30}$$

and α_K is proportional to h_K^2. That amounts to additionally testing the momentum conservation equations by $q_h + \alpha_K \nabla q_h$. The structure of the system is changed by the appearance of a pressure stabilization term $\sum_K \alpha_K (\nabla p_h, \nabla q_h)$. Due to the change of the bilinear form, stability for pressure is now implied by a generalized LBB condition (cf. [4]). As for the streamline diffusion method, the pressure stabilization vanishes for a strong solution u and p, since the stabilizing term is based on the residual of the momentum equation. This pressure stabilization process for the Navier-Stokes equations must also be completed for the momentum equations by the convection stabilization process that has been presented in previous sections.

3.5.5 Stabilization Weights

We define in a first step some forms that describe the stabilizing terms. The pressure stabilization is denoted by

$$c(X_h, q) = \sum_{K \in \mathbb{T}_h} (R_u,\, \alpha_K \nabla q\,)_K, \tag{3.31}$$

with R_u defined in (3.30).

The streamline diffusion method for the velocities involves the term

$$s_u(X_h, \phi) = \sum_{K \in \mathbb{T}_h} (R_u,\, \delta_K\, u_h \cdot \nabla \phi\,)_K \tag{3.32}$$

And the stabilization for the temperature equation consists of the following term:

$$s_T(X_h, \phi) = \sum_{K \in \mathbb{T}_h} \left(\rho \frac{dT_h}{dt} - \nabla \cdot (\lambda \nabla T_h), \gamma_K\, u_h \cdot \nabla \psi_h \right)_K. \tag{3.33}$$

From the energy form (3.20), we define the energy form of the system augmented by the least-squares terms by

$$a_\delta(X_h, \xi_h) = a(X_h, \xi_h) + c(X_h, q_h) + s_u(X_h, \phi_h) + s_T(X_h, \psi_h). \tag{3.34}$$

This discretization has been analyzed for example in [28], [31] or [51]. An error analysis clarifies the role of the parameters and motivates their choice. The

parameters α_K, δ_K and γ_K have to be chosen depending on the local mesh size h_K, the convection u and the viscosity μ or λ on each cell K. Error estimates obtained in [13] allow to derive values for the stabilization parameters for the compressible-low-Mach-number-flow system for which the error of discretization $e = X - X_h$ can be minimized. This study leads to the following values for the velocity stabilization:

$$\delta_K = \frac{h_K}{\triangle t + \mu/(\rho\, h_K) + |u|_\infty}. \tag{3.35}$$

Analogously, minimization of the error in temperature gives

$$\delta_K = \frac{h_K}{\triangle t + \lambda/(c_p\, \rho\, h_K) + |u|_\infty}. \tag{3.36}$$

$\triangle t$ represents the time step. We discuss time discretization in next section. A short analysis of the limit cases helps to understand this stabilization parameter. In the case of convection dominance, the velocity u is greater as the viscosity or time step and $\delta \sim \frac{h}{|u|_\infty}$. If diffusion rules the flow, there is no need to add much stabilization. δ being then proportional to $\frac{h^2}{\mu}$, the second order of the method is assured. For unsteady solutions, when the time step processes are dominant, we have $\delta \sim 1/\triangle t$.

3.6 Time discretization

In this work we are interested in stationary solutions of the system described in Chapter 2. However severe non-linearities in reactive flows may imply a non-stationary behavior of the solution, with small instabilities in time which make a steady-state not exactly reachable. The solution may be considered as quasi-stationary but the system can then only be solved using a non-stationary solution algorithm.

In order to take into account time variation of the solution, we use the expansion $u_h(t,x) = \sum u_i(t)\, \phi_i(x)$. We divide the time interval considered into N parts of size $k_n = t_n - t_{n-1}$. We denote the value of any variable ζ at time t_n by $\zeta^n = \zeta(t_n)$.

The implicit Euler method leads to a system analogous to the following system:

$$b_\delta(X_h^n, \xi_h) + k_n\, a_\delta(X_h^n, \xi_h) = b_\delta(X_h^{n-1}, \xi_h), \tag{3.37}$$

with b_δ being the L_2 scalar product augmented by stabilization terms, i.e. $b_\delta(X, \xi) = (X, \xi) + \sum_{K \in \mathbb{T}_h} (X, \delta\, u \cdot \nabla \xi)$.

The additional term for the stabilization in the form b_δ may be neglected if we are actually looking for a quasi-stationary solution, as said above, and are therefore not interested in the exact evolution in time. This term does not bring more stability to the scheme and makes the process more time-consuming.

3.7 Full Discretization for Reactive Flows

We can now write the discretization of the whole system (2.59)- (2.63). We have the following boundary conditions:

$$\left.\begin{aligned} &\text{symmetry on } \Gamma_0 : u_r = 0, \\ &\text{inflow on } \Gamma_1 : u = u_0, \quad T = T_0, \quad w_i = w_0^{(i)}, \\ &\text{wall on } \Gamma_2 : u = 0, \quad \frac{\partial T}{\partial n} = \sum_{i=1}^{n_s} h_i \, M_i \, \dot{w}_i^0, \quad \frac{\partial w_i}{\partial n} = M_i \, \dot{w}_i^0, \\ &\text{outflow on } \Gamma_3 : \mu \frac{\partial u}{\partial n} - p \cdot n = 0, \quad \frac{\partial T}{\partial n} = 0, \quad \frac{\partial w_i}{\partial n} = 0. \end{aligned}\right\} \tag{3.38}$$

The weak formulation can be written as

$$\left.\begin{aligned} &(\nabla \cdot u_h, q) + (L(u_h, w_h), q) + c(p_h, u_h, q) = N_h(q) \qquad \forall q \in Q_h, \\ &\frac{1}{k_n}(u_h, \phi) + (\rho \, u_h \cdot \nabla u_h, \phi) + (\mu \, \nabla u_h, \phi) \\ &-(\nabla \mu \cdot \nabla u, \phi) - (\mu \, \nabla(\nabla \cdot u), \phi) - (p_h, \nabla \cdot \phi) \\ &\quad +(\mu \frac{u_{r,h}}{r^2}, \phi_r) + s_u(p_h, u_h; \phi) = F_h(\phi) \qquad \forall \phi \in V_h, \\ &\frac{1}{k_n}(w_h^{(i)}, \psi) + (\rho \, u_h \cdot \nabla w_h^{(i)}, \psi) + (\rho \, D_i \, \nabla w_h^{(i)}, \psi) \\ &+(\rho \, D_i \, w_h^{(i)} \nabla \overline{M}, \psi) + s_i(w_h^{(i)}, u_h; \psi) = P_h(w_h, \psi) \qquad \forall \psi \in S_h, \end{aligned}\right\} \tag{3.39}$$

where N_h, F_h and P_h are the corresponding functionals formed by the right-hand side variational formulation and the stabilization. P_h contains the volume chemical source terms but also the surface source terms $\int_{\Gamma_3} M_i \, \dot{w}_i^0 \, \phi \, d\Gamma$. The operator $L(u_h, w_h; q)$ consists of the variational formulation of the convection terms from the continuity equation (2.59). The temperature is considered here as an additional species w_0, since the structure of its evolution equation is the same as the structure of a mass conservation equation for any species. We have $D_0 = \lambda/(\rho \, c_p)$. The density is defined by an algebraic equation $\rho = \rho(w_h)$.

Since diffusion coefficients for each species can differ strongly, one has to define a stabilization parameter for each species:

$$\delta_K^{(i)} = \frac{h_K}{k_n + D_i/h_K + |u|_\infty}. \tag{3.40}$$

and the least-squares stabilization term:

$$\begin{aligned} s_i(w_h^{(i)}, u_h; \psi) = \sum_{K \in \mathbb{T}_h} \Big(&\frac{1}{k_n} w_h^{(i)} + \rho \, u_h \cdot \nabla w_h^{(i)} \\ &-\nabla \cdot \Big(\rho \, D_i \, \nabla(\overline{M} \, w_h^{(i)}) \Big) - M_i \dot{w}_h^{(i)}, \delta_K^{(i)} \, u_h \cdot \nabla \psi \Big)_K . \end{aligned} \tag{3.41}$$

Chapter 4

Numerical Solution

To solve the strongly non-linear system coming from the finite element discretization of multispecies reactive flows, we consider the classical approach based on a linearization of the system with the help of its jacobian matrix. The iterative method used in this work is a defect correction method which requires to solve a linear system in each non-linear step. In such an algorithm for computing complex reactive flows, two ingredients are decisive for the efficiency of the total solution process: an economical storage technique which fully exploits the very special structure of the jacobian matrices, and an efficient and robust solver for the large coupled linear systems.

This chapter discusses the linear systems obtained from a simplification of the jacobian matrix, which may be efficiently solved. This iteration matrix has to provide enough accuracy according to the non-linear system to obtain an acceptable convergence rate of the defect correction process. We will also describe methods to solve the resulting linear systems.

To solve the linear systems we have chosen a Generalized Minimal Residual (GMRES) algorithm. GMRES is appropriate for non-symmetric and indefinite matrices. In order to obtain an efficient solver with a rate of convergence independent of the mesh size, we use a multigrid scheme as a preconditioner. The locally-refined structure of the mesh makes the preconditioning through a multigrid method necessary to avoid the dependence of the condition number on the mesh width.

The grids under consideration are obtained as follows: The coarsest mesh does not contain any hanging node and consists of cells belonging to the level $l = 0$. The cells of level $l \geq 0$ are obtained by refinement of some cells belonging to level $(l - 1)$. Due to this hierarchical refinement strategy the required smoothing operations in a multigrid cycle on level l are restricted to the cells belonging to this level. We use in this work different smoothing operators. For the Navier-Stokes part of the system, we have implemented a method similar to the smoother proposed by Vanka in [52] for staggered grid finite

difference discretizations, which consists of a block Gauss-Seidel iteration loop. The decomposition in blocks is done patch-wise on each level of the grid and corresponds to a local grouping of velocities and pressure unknowns. For the smoothing of the temperature and species equations we use two methods; the first one is based on point-Gauss-Seidel iterations, while the second one may be used in the case of stiffer systems and is based on a block-ILU decomposition.

4.1 Defect Correction

As mentioned above, the non-linear system of equations is solved by a defect-correction method. The iteration matrix is an approximation of the jacobian of the non-linear equations. This method is based on the Newton iteration which consists of the following fix-point iteration

$$X^{n+1} = \begin{pmatrix} u \\ p \\ w_i \end{pmatrix}^{n+1} = \begin{pmatrix} u \\ p \\ w_i \end{pmatrix}^{n} - \omega \left(D_R\right)^{-1} R^n, \tag{4.1}$$

with the following notations:

$$\begin{aligned} D_R &= \text{derivative of R with respect to the variables u, p, T, } w_i, \\ R &= \text{residual of the system that is to be solved,} \\ \omega &= \text{relaxation parameter.} \end{aligned}$$

For the sake of simplicity we will consider the temperature in this chapter as the first term of the vector defining the chemical state of the flow, i.e. $w_0 = T$, since the equations for temperature and those for the species have exactly the same characteristics.

We also denote the increments for our solution vector by

$$d_X^{n+1} = \begin{pmatrix} d_u \\ d_p \\ d_{w_i} \end{pmatrix} = \begin{pmatrix} u^{n+1} - u^n \\ p^{n+1} - p^n \\ w_i^{n+1} - w_i^n \end{pmatrix}, \tag{4.2}$$

$n+1$ being the number of the current non-linear step.

In the defect-correction process, D_R is actually not computed exactly since a suitable approximation of this derivative is often sufficient to solve the system. For this reason, with the additional use of a relaxation parameter ω, this method is called *quasi*-Newton method, when the computed D_R converges to the exact final D_R, or defect correction method otherwise.

Damping the iteration step with the parameter w leads to a stabilization of the algorithm. ω is chosen to be $w = 2^{-i}$ where i is the lowest integer greater

than 0 such that the monotonicity $|R(X^n - 2^{-i}\, d_X^{n+1})| < |R(X^n)|$ is fulfilled. $X^{n+1} = X^n - 2^{-i}\, d_X^{n+1}$ is then chosen as the update. This stabilization is necessary to have a robust solver and avoid oscillations in the convergence of the method. An example of divergence in the case without damping can be found in [48].

4.2 Newton Matrix

The aim of this section is to describe the construction of the jacobian matrix and its approximation. We present the jacobian matrix and its approximation used in this work in order to reduce storage requirements and computation time. We introduce the following form which is the residual of the system:

$$\begin{aligned} R(\{p,u,w\},\{q,\phi,\psi\}) =& R_u(\{p,u,w\},\phi) + R_p(\{p,u,w\},q) \\ &+ \sum_{i=0}^{n_s} R_{w_i}(\{p,u,w\},\psi), \end{aligned} \tag{4.3}$$

where R_u, R_p are the partial residuals according to the Navier-Stokes equations and R_w the partial residual according to the temperature-species equations:

$$\begin{aligned} R_u(\{p,u,w\},\phi) &= \left(\rho\,\frac{du}{dt},\phi\right) + (\mu\nabla u,\nabla\phi) - (p,\nabla\cdot\phi) \\ &\quad + (\mu\,\frac{u_r}{r^2},\phi) - (\nabla\mu\cdot\nabla u,\phi) - (\mu\,\nabla(\nabla\cdot u),\phi) - (f_v,\phi), \\ R_p(\{p,u,w\},q) &= (\nabla\cdot u,q) + (L(u,w),q) + (\nabla p,\delta\,\nabla q), \\ R_{w_i}(\{p,u,w\},\psi) &= \left(\rho\,\frac{dw_i}{dt},\psi\right) + (\rho\,D_i\nabla w_i,\nabla\psi) - (f_{w_i},\psi), \\ & \qquad\qquad i = 0,\ldots,n_s. \end{aligned} \tag{4.4}$$

Taking into account the stabilization terms would not change the structure of this system. The only stabilizing term which changes the characteristics of the system is the term $(\nabla p,\delta\,\nabla q)$ in the operator $c(X,q)$ defined in relation (3.31).

The jacobian matrix corresponding to the residual given above is

$$D_R = \begin{bmatrix} \frac{\partial R_u}{\partial u} & \frac{\partial R_u}{\partial p} & \frac{\partial R_u}{\partial w_j} \\ \frac{\partial R_p}{\partial u} & \frac{\partial R_p}{\partial p} & \frac{\partial R_p}{\partial w_j} \\ \frac{\partial R_{w_i}}{\partial u} & \frac{\partial R_{w_i}}{\partial p} & \frac{\partial R_{w_i}}{\partial w_j} \end{bmatrix}, \tag{4.5}$$

with $i = 0, \ldots, n_s$ and $j = 0, \ldots, n_s$. For the approximation of this matrix, we must take the physics of the flow into consideration as well as the ability to efficiently calculate the derivatives and solve the system at low computational cost. The flow variables u, p are coupled with the chemical state w through the mixture viscosity μ, the density ρ and the mean molar mass $\bar{M}$ in the Navier-Stokes equations and through the velocity of the fluid in the convection-diffusion equations for the temperature and species. For our application to flow reactors, no rapid variation of the physical quantities should be observed in almost the whole domain. Therefore, in order to be able to bring efficient smoothers into play, we decide to keep only a weak coupling between the Navier-Stokes equations and the temperature/species equations. The system is correspondingly linearized at each non-linear step. In the approximate jacobian we neglect the blocks $\frac{\partial R_u}{\partial w}$, $\frac{\partial R_w}{\partial u}$ and $\frac{\partial R_p}{\partial w}$. The term $\frac{\partial R_w}{\partial p}$ is also not taken into account since the temperature is almost independent of the pressure for low-Mach-number flows. The density couples the equations through the ideal gas law (2.46). Viscosity, mean molar mass and species or temperature convection velocities are calculated in each non-linear step with the previous-non-linear-iteration value of the solution vector.

With these simplifications, the approximation of the operator D_R has the following block-form:

$$\widetilde{D}_R = \begin{bmatrix} A_{pp} & A_{pu} & 0 \\ A_{up} & A_{uu} & 0 \\ 0 & 0 & G \end{bmatrix}. \tag{4.6}$$

While denoting the test and trial functions by ψ and ζ, respectively, we can write the approximated operators defining $\widetilde{D}_R$ using overlined variables as the linearized variables calculated with their values taken from the previous non-linear step.

For the continuity equation, A_{pu} corresponds to the sum of the divergence operator with the element-wise least-squares terms stemming from the streamline-diffusion stabilization and A_{pp} results from the pressure velocity stabilization:

$$A_{pp} = \sum_{K \in \mathbb{T}_h} \left(\nabla\zeta,\, \alpha_K \nabla\psi \right)_K , \tag{4.7}$$

$$\begin{aligned} A_{pu} &= (\nabla \cdot (\rho\,\zeta), \psi) + \sum_{K \in \mathbb{T}_h} \left(\rho \frac{d\zeta}{dt} - \nabla \cdot (\mu \nabla\zeta) \right. \\ &+ \left. \frac{\mu}{r^2} \begin{pmatrix} u_r \\ 0 \end{pmatrix} - \mu \nabla(\nabla \cdot \zeta) - \nabla\mu \cdot \nabla\zeta,\, \alpha_K \nabla\psi \right)_K , \end{aligned} \tag{4.8}$$

with the total time derivative $\frac{d}{dt} = \frac{\partial}{\partial t} + \bar{u} \cdot \nabla$. The variable $\bar{u}$ is here the velocity evaluated at the previous step of the iterative process. We neglect the other part of the derivative of the transport term with regards to u.

Furthermore the operator A_{up} represents the influence of the pressure in the momentum conservation equation, and A_{uu} corresponds to the convection-diffusion terms in this equation:

$$A_{up} = -(\zeta, \nabla \cdot \Psi) + \sum_{K \in \mathbb{T}_h} (\nabla\zeta, \alpha_K \, \bar{u} \cdot \nabla\Psi)_K, \tag{4.9}$$

$$\begin{aligned} A_{uu} = & \left(\frac{d\zeta}{dt}, \Psi\right) + (\mu \, \nabla\zeta, \nabla\Psi) + (\mu \frac{u_r}{r^2}, \Psi_r) \\ & -(\nabla\mu \cdot \nabla u, \Psi) - (\mu \, \nabla(\nabla \cdot u), \Psi) \\ & + \sum_{K \in \mathbb{T}_h} \left(\frac{d\zeta}{dt} - \mu \, \nabla\zeta, \, \alpha_K \, \bar{u} \cdot \nabla\Psi\right)_K . \end{aligned} \tag{4.10}$$

Considering (4.6), we see that the linearized system is split into two independent parts. One part determines the evolution of the flow, the other part corresponds to the chemistry and the behavior of species within the flow. The operator G corresponds to the convection-diffusion-reaction terms of the species mass conservation equations and to the temperature evolution equation, which have the same structure. While considering the interactions between the species, the block-matrix G can be decomposed into $(n_s+1)\times(n_s+1)$ matrices, the temperature being considered as a separate species. The diagonal matrices G_{ii} correspond to the convection-diffusion of the mass fraction of the species i, as well as the reaction of this species in the gas-phase or at the wall. For all $i = 0, \ldots, n_s$ we have

$$\left.\begin{aligned} G_{ii} = & \left(\frac{d\zeta}{dt}, \psi\right) + (\rho \, D_i \, \nabla(\overline{M}\,\zeta), \nabla\psi) - \left(M_i \frac{\partial \dot{w}_i}{\partial w_i}, \psi\right) \\ & - \left(M_i \frac{\partial \dot{w}_i^0}{\partial w_i}, \psi\right)_{\Gamma_{\text{wall}}} + \sum_{K \in \mathbb{T}_h} \left(\frac{d\zeta}{dt} + \rho \, D_i \, \nabla(\overline{M}\,\zeta), \, \delta_K^{(i)} \, \bar{u} \cdot \nabla\psi\right)_K \\ & - \sum_{K \in \mathbb{T}_h} \left(M_i \frac{\partial \dot{w}_i}{\partial w_i}, \, \delta_K^{(i)} \, \bar{u} \cdot \nabla\psi\right)_K . \end{aligned}\right\} \tag{4.11}$$

The non-diagonal block-matrices elements of the matrix G, denoted by G_{ij} where $i, j = 0, \ldots, n_s$ and $i \neq j$, correspond to the coupling between the species through chemical reactions: which species are created while others are chemically transformed. These block-matrices contain only derivatives of gas-phase or wall production terms. For all $i, j = 0, \ldots, n_s$ with $i \neq j$, we have

$$\begin{aligned} G_{ij} = & -\left(M_i \frac{\partial \dot{w}_i}{\partial w_j}, \psi\right) - \left(M_i \frac{\partial \dot{w}_i^0}{\partial w_j}, \psi\right)_{\Gamma_{\text{wall}}} \\ & - \sum_{K \in \mathbb{T}_h} \left(M_i \frac{\partial \dot{w}_i}{\partial w_j}, \, \delta_K^{(i)} \, \bar{u} \cdot \nabla\psi\right)_K . \end{aligned} \tag{4.12}$$

As noted in Section 2.3.2, we want here to emphasize the importance of the gas-phase and wall production terms in the jacobian matrix. Even if these production terms may have small influence on the residuum (the surface reactions occur on 1D domains – for 2D computations), the convergence largely depends on their presence in the jacobian matrix. Actually the difference on the convergence between two methods using different approximations of the jacobian matrix may be noticed only very late in the convergence process. The convergence criterion (often residuum smaller than a certain tolerance) has to be chosen carefully. Indeed numerical experiments showed us that, for some approximations, a residuum drop which could seem to be sufficient according to accepted tolerances for Navier-Stokes solver, is actually not enough for the convergence of the chemical processes, principally for surface reactions. Some surface reactions may not be taken into account at this point in the convergence process. This means that we must be careful about local convergence for chemical reactions or accept to solve the system with a convergence to the zero machine. We have tested several approximations of the jacobian matrix in order to understand which terms were necessary. Comparison for the wall reaction terms can be found in Chapter 6.

If one decided to delete one species, as proposed in Section 2.5, in order to make the approximated solution automatically fulfill the constraint (2.42), the jacobian matrix has to be calculated in a slightly different manner. The reader can find a complete explanation of this method in [13]. A chemical component can be deleted and its mass conservation equation substituted by the relation (2.42). The jacobian matrix of the resulting system is then calculated.

4.3 Implementational Constraints

The size of G depends on the number of species and the number of degree of freedom in the discretization. The latter is controlled through an adaptive process which will be discussed in Chapter 5; it is typically in a range between 3000 to 20000. The sparse matrix type we use in the implementation is supplied by the DEAL library and is usually used for solving large linear system resulting from a finite element discretization. The reader can find a description of this sparse matrix structure in [43]. In our test applications, in Chapter 6, the maximal number of species considered is 39. Due to memory restrictions, with so many species, if we want to achieve enough approximation accuracy, we cannot keep the whole matrix G in memory. Thus, with regards to the memory available, we decide to keep the whole matrix G or reduce it to its block-diagonal part, i.e. not to take the matrices G_{ij} into account. This simplification is reasonable only if the reaction terms are smooth. We will see that the resulting defect correction method still converges for our applications with an acceptable convergence rate with regard to calculation time. For problem

with more intense reactions, we may be forced to take the whole matrix into account.

4.4 Solvers

The global solution process for steady nonlinear systems used for our purpose can be seen as a nested process (see Fig. 4.1) involving, within a defect-correction scheme based on a Newton iterative method, a preconditioned Generalized Minimal Residual method (GMRES) as linear solver (see [45]), where the preconditioner is chosen to be a multigrid method. Our implementation is based on the multigrid method developed by Becker in [4], which offers the ability to handle locally refined grids. For our multigrid method we use several smoothers depending on the systems we have to solve. For unsteady problems a loop over time steps wraps again the whole process.

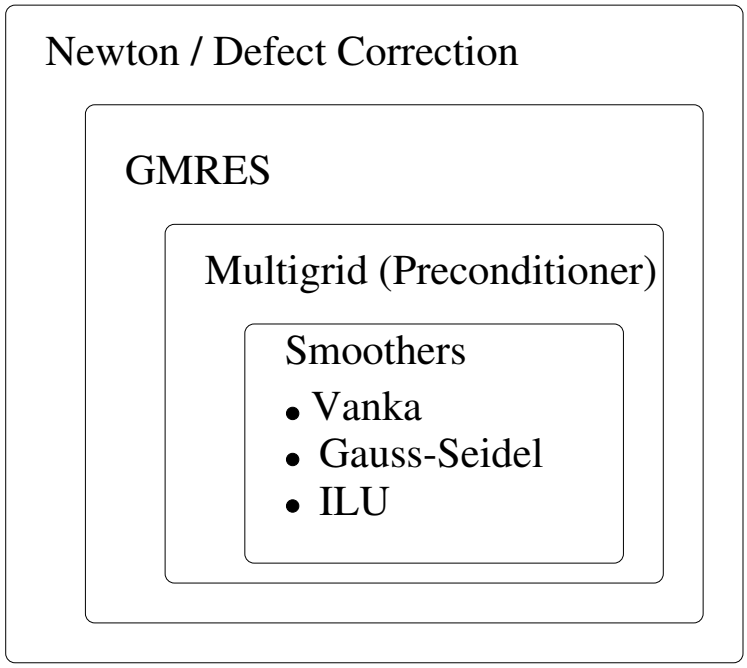

Figure 4.1: *Nested solution process.*

In each nonlinear step of the defect-correction method, a linear system is to be solved. Since the linearized system is decoupled due to (4.6), we may implement two linear solvers: one for the mixture-averaged flow (i.e. Navier-Stokes), the other for the species convection-diffusion-reaction process. This requires two different strategies for the smoothing iteration. In our implementation we have chosen a "Vanka smoother" for the Navier-Stokes part of the system

and a Gauss-Seidel smoother or an ILU smoother for the chemical part of the system.

4.4.1 Multigrid

The mesh we use for the discretization comes from a refinement process (see Chapter 5) which makes the hierarchical structure of the triangulation available. The idea is to use this structure to implement an efficient preconditioner based on multi-level techniques.

The application of multigrid methods on locally refined mesh is not trivial. The reader can find a detailed explanation in the work of Becker [4] and an implementation in the DEAL finite-element library (see [6]). We only sketch here the essential steps of such a method.

The multigrid process we use for our purpose is based on a V-cycle. On the coarsest grid $\mathbb{T}_0$ the system is solved exactly. On other levels $\mathbb{T}_l$, a pre-smoothing is done and the residual is then restricted on a coarser grid $\mathbb{T}_{l-1}$ where this process is recursively repeated until the coarsest grid is reached. Then the solution is prolongated from the coarser grid $\mathbb{T}_{l-1}$ to the grid $\mathbb{T}_l$ and a post-smoothing is carried out.

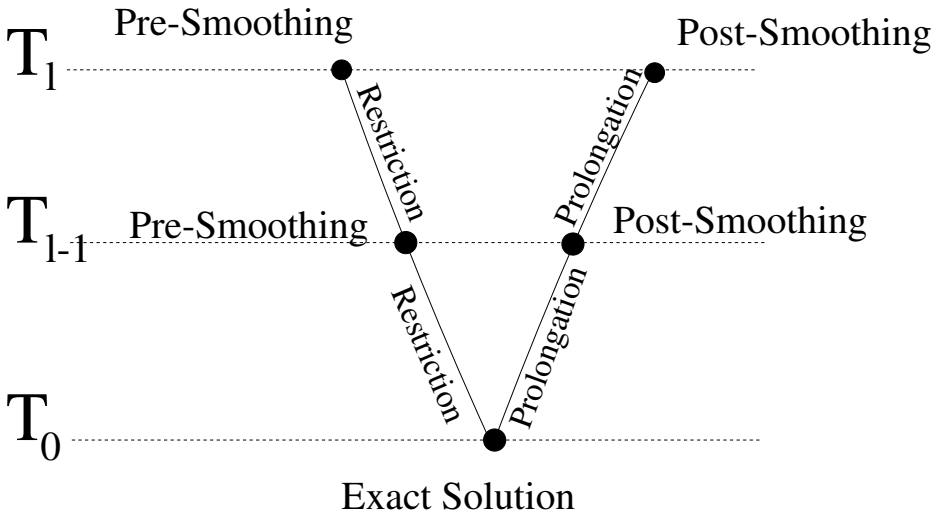

Figure 4.2: *Multigrid V-cycle.*

In the following subsections we describe the smoothing operators. The smoothing of the residual is done level-wise. Smoothing the residual on each level of the mesh means eliminating its high frequencies in order to approximate it accurately on a coarser grid. A possibility is to smooth the system with a fixed number of GMRES steps on each level of the triangulation. Nevertheless

this leads to bad performance of the multigrid method, especially when the mesh contains more than four or five levels. We need to use methods which have good smoothing properties (not compulsorily a solver) without demanding too much computational effort since the smoother works on each level of the mesh. For the Navier-Stokes equations, we have therefore implemented a Vanka-type smoother, which is a block-Gauss-Seidel iterative method. The block are constructed by considering a patch-wise grouping of pressure and velocities unknowns. The species equations are smoothed with the help of a point-Gauss-Seidel iterative method or an ILU method. In order to obtain good smoothing properties, it is well known that these two methods require a renumbering of the grid nodes in the direction of the flow. We will shortly discuss this point as final remark.

4.4.2 Vanka Smoothing Operator

As smoothing operator for the Navier-Stokes equations we employ a block-Gauss-Seidel iteration similar to the one proposed by Vanka in [52]. A smoothing step consists of a loop over the pressure degrees of freedom, where we simultaneously update the corresponding pressure value together with the velocity unknowns which are coupled with it, by solving a local system derived from the Navier-Stokes equations.

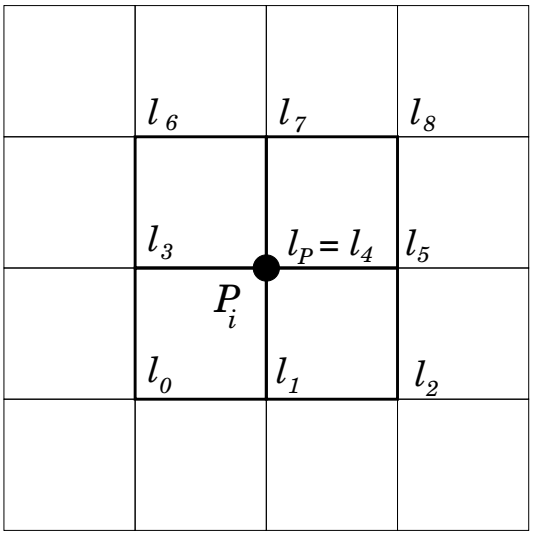

Figure 4.3: *Patch defining the local problems for the Vanka smoother.*

To this end, we associate with each pressure point P_i of the considered level a patch consisting of the cells having P_i in common (see Fig. 4.3). On each patch we define the indices l_i with $0 \leq i \leq 8$ for velocity degrees of freedom and the

local index l_p corresponding to the pressure point. The discrete operator for the stabilized Navier-Stokes system of equations can be written as follows:

$$\begin{pmatrix} A & B \\ E & C \end{pmatrix}. \tag{4.13}$$

Having calculated the residuals r_i and r_p for the velocity and the pressure respectively, we obtain, after simplification, the following local system for the velocity and pressure updates d_i and d_p:

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 & b_1 \\ 0 & a_{22} & \dots & 0 & b_2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & a_{nn} & b_n \\ e_1 & e_2 & \dots & e_n & c \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ d_p \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \\ r_p \end{bmatrix}. \tag{4.14}$$

This system has been simplified by neglecting the coupling terms between the velocity degrees of freedom (i.e. $a_{12}, \dots$). It can be easily solved with two passes over the involved unknowns. This construction provides velocity updates which satisfy the mass conservation equation with respect to the test function on the patch.

The Vanka smoother showed more robustness than a simple Gauss-Seidel smoothing during tests done on the Navier-Stokes equations with constant viscosity. It is well known for saddle-point problems that by increasing the Reynolds number of the flow, the linearized systems may still be solved with the Vanka smoother, while when using the Gauss-Seidel smoother the whole process shows poor convergence rates or even diverges. Numerical tests on our application cases for flow reactors led us to set the number of pre- and post-smoothing steps with the Vanka smoother each to four. Less iteration steps could handicap the efficiency of the multigrid method as preconditioner.

4.4.3 Chemical System Smoothing

The chemical system formed by the species mass conservation equations and the temperature evolution equations is solved with the help of Gauss-Seidel iterations or, for more stiff systems, with an ILU method, a description of which can be found in [13]. We use an ILU(0) from the MV++ and IML++ packages (see [43] and [21]). MV++ implements efficient matrix/vector classes designed for high performance numerical computing and IML++ is a collection of algorithms for solving or preconditioning linear systems of equations. The idea of the ILU method is to compute a factorization of the form

$$A = LU, \tag{4.15}$$

where A is the matrix of our system, L and U are a lower and an upper triangular matrix respectively. In general L and U will be dense matrices. The incomplete LU method of order zero provides approximations of these two matrices, $\tilde{L}$ and $\tilde{U}$, which have the same sparse structure as the matrix A. This allows to reduce memory requirements and to calculate the decomposition with low computational costs. The factorization remains accurate enough to ensure the robustness of the method. Some examples of application of incomplete LU methods may be found in [59] and [13].

The block Gauss-Seidel iterative method is not as robust as ILU methods but is less time-consuming and can be used as an efficient smoother for linear systems which do not contain too strong convection and source terms. With regard to the implementation of a smoother for the chemical system, one must only be aware of the limits of this method and should make an ILU method also available. The Gauss-Seidel smoother is used as pre- and post-smoother for the multigrid method with a number of steps typically each between two and five.

The efficiency of these two methods is extremely dependent on the numbering of the mesh points. To be able to use the information transport within the flow, the degrees of freedom have to be numbered in streamline direction. Since we need the smoother on each level of the mesh, the numbering of the nodes has to be realized independently on each refinement level. A renumbering method based on the minimization of a functional depending on the velocity of the flow is described in [13]. However this sort of renumbering method might demand some computational effort and if the direction of the flow is known in advance, one may prefer to make the numbering simply dependent on this direction, which is done very quickly. We used the latter method in our applications on low-pressure flow reactors.

Chapter 5

Adaptivity

5.1 Introduction

It is frequently the case in engineering problems that the main quantity of concern is not the solution of a partial differential equation, but a secondary quantity which is a scalar functional of the solution.

The strategies for mesh refinement conventionally used in finite element methods are mostly based on *a posteriori* error estimates in global norms involving local terms corresponding to residuals of the computed solution. The mesh refinement process aims at equilibrating these local error indicators. However meshes generated on the basis of such global error estimates may not be appropriate for controlling the accuracy in approximating local quantities such as point values or contour integrals. More detailed information is needed on the mechanism of error propagation with regard to these quantities depending on the solution. This can be obtained by employing suitable duality arguments. The corresponding dual solution is approximated on the current mesh and is used to derive local weight factors which are used in the *a posteriori* error estimates.

Our aim in this chapter is to propose an approach to the derivation of *a posteriori* bounds on the error in linear functionals for reactive flows in order to be able to compute some physical quantities with best accuracy. A functional $J(\cdot)$ of the solution is defined, which may represent for example local values of the temperature, contour average of species mass fractions or point values of certain components of the system. In these cases the error control is applied only to a part of the solution. When such well defined quantities are to be calculated with precision, an error indicator allows to control the approximation error on these quantities for the calculated solution.

We present in this chapter an adaptive algorithm leading to reliable and efficient error control in our context, according to a functional as described above.

It allows to calculate the solution with a controlled accuracy for the value of the functional $J(\cdot)$ on "optimal" meshes for our FEM Ansatz according to the corresponding error estimator. "Optimal" means either "most economical for achieving a prescribed accuracy" or "most accurate for a given number N of mesh points". The functional is assumed in this work to be linear although the approach can be extended to non-linear functionals (see [8]).

The error estimation is based on duality arguments. The dual problem is obtained from a linearization of the primal problem. Since the dual problem is linear, the additional cost induced by the computation of the error estimator corresponds to only one Newton step of the solution of the non-linear primal problem on each mesh level.

In contrast to the error bound obtained by duality arguments, a classical approach to adaptivity for reactive flows supplies error indicators usually based on the estimation of a global stability constant, independently of any quantity derived from the solution (see [53]). For scalar equations, such an indicator η_{ind} has the form

$$\eta_{ind} = \sum_{K \in \mathbb{T}_h} \omega_K \, \gamma_K, \tag{5.1}$$

where ω_K is a weight depending on the cell K and γ_K is a suitable difference quotient of the discrete solution approximating some derivative. In reactive flow computations, the situation is more complicated since we deal with a system of equations. For systems the corresponding indicator reads

$$\eta_{ind} = \sum_{K \in \mathbb{T}_h} \sum_{i=1}^{n} \omega_K^i \, \gamma_K^i. \tag{5.2}$$

In order to sum over all the equations, a scaling of the computed variables (for instance the mass fraction) may be necessary, since the concentration of the species in the mixture may sensibly differ by many orders.

Through the approximation of derivatives by differential quotients, such an indicator will capture the strong variations in concentration and therefore will lead to a refinement in reaction zones. However the absence of information on global error propagation as well as on the coupling between the different components may have negative influence on the quality of the discrete solution by not refining the mesh where the error is actually created. Moreover there is no possibilities to control the error on quantities depending on the solution.

Other traditional approaches to the construction of locally adapted meshes often resort to *ad hoc* criteria, often gradients of physical quantities, whose impact on the accuracy of the numerical solution is difficult to assess.

In the first section of this chapter an error estimate for a functional in the simple case of a linear convection-diffusion equation is developed. This concept

is then applied to a non-linear PDE. We finally apply the error estimation to reactive-flow problems and then discuss how to organize a mesh refinement process with the help of the computed estimator.

5.2 Error Estimation for a Linear Scalar Equation

We consider the scalar convection-diffusion equation with homogeneous Dirichlet boundary conditions. Let β be a given vector field. The variational formulation consists in finding $u \in V = H_0^1(\Omega)$ such that

$$a(u, \phi) = (\beta \cdot \nabla u, \phi) + (\mu \nabla u, \nabla \phi) = (f, \phi) \qquad \forall\, \phi \in V. \tag{5.3}$$

This problem is approximated by a Galerkin finite element method using a sequence of test and trial spaces $V_h \subset V$ parameterized by a discretization parameter h. The discrete problem reads: find u_h in V_h such that

$$a(u_h, \phi) = (f, \phi) \qquad \forall\, \phi \in V_h. \tag{5.4}$$

For the sake of simplicity, the modification due to the stabilization of the equation by the streamline diffusion method is not taken into account; it will be included later.

Subtracting (5.4) from (5.3), we obtain the Galerkin orthogonality relation for the error $e = u - u_h$,

$$a(e, \phi) = 0 \qquad \forall \phi \in V_h. \tag{5.5}$$

The error e is orthogonal to the space V_h with respect to the bilinear form a, which is a characteristic property of Galerkin methods.

We now define the functional of the solution that is to be accurately known. Let $J : V \to \mathbb{R}$ be a linear functional. The aim of the adaptive process is to construct an appropriate triangulation $\mathbb{T}_h$ and to compute u_h with the condition that

$$|J(e)| = |J(u) - J(u_h)| \leq TOL \tag{5.6}$$

for a given tolerance TOL.

To know if $J(u_h)$ is calculated accurately enough, one must be able to bound the error $J(e)$ defined above. Hence it must be expressed only in terms of the approximated solution u_h, since the continuous solution u is not known.

We consider therefore the solution $z \in V$ of a corresponding dual problem

$$a(\phi, z) = J(\phi) \qquad \forall\, \phi \in V, \tag{5.7}$$

where trial and test functions are interchanged with respect to the primal problem (5.3). The corresponding continuous operator of this dual problem is by definition the adjoint of the operator of the primal problem. Integration by parts yields the following representation of this operator:

$$L^* = -\beta \cdot \nabla - \mu \triangle. \tag{5.8}$$

This means that convection occurs in the opposite direction as for the primal problem. The dual problem carries information upstream.

The Galerkin orthogonality argument (5.5) and the dual problem (5.7) together lead to an error representation in terms of the dual solution z:

$$J(e) = a(e, z) = a(e, z - i_h z) = (f, z - i_h z) - a(u_h, z - i_h z) \tag{5.9}$$

for an arbitrary interpolation $i_h z \in V_h$ of the dual solution $z \in V$. We will see later the aim of the introduction of this interpolation of the dual solution in the space V_h.

From (5.3) we get

$$J(e) = (f - \beta \cdot \nabla u_h, z - i_h z) - (\mu \nabla u_h, \nabla(z - i_h z)) \tag{5.10}$$

Thus we have reached a formulation of the functional where the unknown continuous solution does not appear. Expressing the scalar product element-wise, an integration by parts leads to the exact error representation as a function of the residual of the primal system and $[\nabla u_h]$, the jumps of the first derivatives over the cell edges:

$$\begin{aligned} J(e) = &\sum_{K \in \mathbb{T}_h} (f - \beta \cdot \nabla u_h + \mu \triangle u_h, z - i_h z)_K \\ &- \frac{1}{2} \sum_{K \in \mathbb{T}_h} (\mu \, n \cdot [\nabla u_h], z - i_h z)_{\partial K}, \end{aligned} \tag{5.11}$$

with n the external normal vector to the edge ∂K. Note that the normal derivatives of u_h are discontinuous over the cell edges.

Although equation (5.11) is independent of u, it still contains the unknown continuous dual solution z. Therefore the error on the functional $J(e)$ cannot be evaluated numerically in this form and the term $z - i_h z$ must be approximated in an appropriate way. Several methods for this are presented in [8]. One usually uses the cell-wise approximation of the expression $\|z - i_h z\|_K$. Indeed by applying the Cauchy-Schwarz inequality on (5.11) in order to get an upper bound for $J(e)$, the resulting estimator is

$$|J(e)| \leq \sum_{K \in \mathbb{T}_h} \omega_K \, \rho_K \tag{5.12}$$

with ρ_K the residual of the primal equation and ω_K additional weights depending on the dual solution:

$$\rho_K := h_k^2 \, \|\beta \cdot \nabla u_h - \mu \triangle u_h - f\|_K + \frac{1}{2} \mu \, h_k^{3/2} \, \| \, n \cdot [\nabla u_h] \, \|_{\partial K}, \quad (5.13)$$

$$\omega_K := \max \left\{ h_K^{-2} \|z - i_h z\|_K \, , \, h_K^{-3/2} \|z - i_h z\|_{\partial K} \right\}. \quad (5.14)$$

The residuals ρ_K can be now computed numerically since they depend only on the discrete solution u_h. However the weights have still to be approximated. ω_K can be replaced by an approximation obtained by using local interpolation estimates (see [5])

$$\omega_k \leq C_K \, \|\nabla^2 z\|_K, \quad (5.15)$$

with an interpolation constant C_K.

Following the approach proposed in [8], in the local interpolation estimate (5.15) the exact dual solution z is replaced by an approximation z_h, discrete solution of the dual problem

$$z_h \in V_h : \qquad a(\phi, z_h) = J(\phi) \qquad \forall \phi \in V_h. \quad (5.16)$$

For simplicity, we use the same discrete space V_h for the discrete dual problem, although a finer or coarser mesh could be used.

The validity of this approximation in our application cases is justified by the results we obtain using this method in this work as well as by the results obtained in other works such as in [48]. If we substitute the second order difference quotient $\|\nabla_h^2 z_h\|_K$ for the second derivative of the dual solution in the bound in (5.15), the error can now be estimated by

$$|J(e)| \approx \eta := \sum_{K \in \mathbb{T}_h} \eta_K, \qquad \eta_K = \tilde{\omega}_K \, \rho_K, \quad (5.17)$$

with approximated weights $\tilde{\omega}_K$ numerically evaluated as

$$\tilde{\omega}_K := C_K \, \|\nabla_h^2 z_h\|_K. \quad (5.18)$$

After determining the solution u_h of the primal problem (5.3), the discrete dual problem (5.7) has to be solved. Then the residuals ρ_K and weights $\tilde{\omega}_K$ are evaluated on each cell in order to get the local error indicators η_K. The total error with respect to the error functional J is then estimated by (5.17).

5.3 Error estimation with streamline diffusion

For the stabilized discretization, the corresponding error estimate involves further terms which are needed in further developments. The modification of the bilinear form does not affect the practical computation but is relevant for the form of the a posteriori error estimate given by (5.17). The reader can find more details on this subject in [22].

We modify the bilinear form a given in (5.3) just as in Section 3.5.3 to obtain the stabilized bilinear form $a_h := a + a_\delta$, with a_δ defined by

$$a_\delta(u,\phi) := \sum_{K\in\mathbb{T}_h} \delta_K \left(\beta\cdot\nabla u - \mu\triangle u, \beta\cdot\nabla\phi\right)_K. \tag{5.19}$$

We obtain in the same way the stabilized right hand side $f_h := f + f_\delta$, with f_δ defined by

$$f_\delta(\phi) := \sum_{K\in\mathbb{T}_h} \delta_K \left(f, \beta\cdot\nabla\phi\right)_K. \tag{5.20}$$

The discrete equation is then

$$a_h(u_h,\phi) = f_h(\phi) \qquad \forall\phi\in V_h. \tag{5.21}$$

The consideration of the stabilized linear problem with least-square terms leads to the full Galerkin orthogonality

$$a_h(e,\phi) = 0 \qquad \forall\,\phi\in V_h. \tag{5.22}$$

At this point, we are at the same stage in the method as for the simple Galerkin orthogonality equation (5.5). We just have to interchange the bilinear form a with the form a_h. The dual solution z searched in V fulfills now the equation

$$a_h(\phi,z) = J(\phi) \qquad \forall\,\phi\in V. \tag{5.23}$$

The error estimate becomes then

$$\begin{aligned} J(e) &= a_h(e,z) = a(e, z-i_h z) + a_\delta(e, z-i_h z) \\ &= (f, z-i_h z) - a(u_h, z-i_h z) + f_\delta(z-i_h z) - a_\delta(u_h, z-i_h z). \end{aligned}$$

Following the same reasoning as in the case without stabilization, an *a posteriori* bound of the error with respect to the functional $J(\cdot)$ can be derived:

$$|J(e)| \le \sum_{K\in\mathbb{T}_h} \left\{ \omega_K\,\rho_K + |\delta_K\left(\beta\cdot\nabla u_h - \mu\triangle u_h - f,\ \beta\cdot\nabla(z-i_h z)\right)_K| \right\}, \tag{5.24}$$

with $\omega_K\,\rho_K$ defined as in previous section.

The estimation of $\nabla(z - i_h z)$ by the second derivative of z,

$$\|\nabla(z - i_h z)\|_K \leq C_K \, h_K \, \|\nabla^2 z\|_K, \tag{5.25}$$

leads to the following bound:

$$|J(e)| \leq \sum_{K \in \mathbb{T}_h} \omega_K \, (\rho_K + |\beta|_{\infty,K} \, \delta_K \, h_K \, \|\beta \cdot \nabla u_h - \mu \, \triangle u_h - f\|_K \,). \tag{5.26}$$

It is to be noted that the supplementary stabilization term has at least the same order in h_K as the term ρ_K, since the stabilization parameter δ_K depends on h_K (see Section 3.5.5).

5.4 Error Estimation for Non-linear Equations

We now apply the weighted error estimator, explained previously for a linear scalar equation, to non-linear problems. Let V be a Hilbert space with inner product $(.,.)$ and corresponding norm $\|.\|$, and $a(.,.)$ a semi-linear form (linear in the second argument). The variational formulation of the corresponding problem is: find $u \in V$ such that

$$a(u, \phi) = (f, \phi) \qquad \forall \phi \in V. \tag{5.27}$$

The discretization in a finite-dimensional subspace $V_h \subset V$ is: find $u_h \in V_h$ such that

$$a(u_h, \phi) = (f, \phi) \qquad \forall \phi \in V_h. \tag{5.28}$$

Let $e = u - u_h$ be the error between the continuous and the discretized solution, and $J(\cdot)$ the functional of the solution, still considered as linear, which is to be accurately known.

The aim is to find a system, named dual system in the previous section, which allows us to get an upper bound of the error on the functional. In order to have a variational formulation of this system, the form describing the problem must be linear in the test function. Moreover the linearity of the primal problem had made it possible in the previous section to write explicitely $J(e)$ independently of the continuous primal solution in equation (5.9) and following. The same argumentation cannot be used here.

Therefore, if we want to keep the same reasoning, we have to find, from the primal non-linear system, a linear system which allows us to write $J(e)$ independently of the continuous solution.

With this aim in view, we consider the derivative $a'(\cdot;\cdot,\cdot)$ of $a(\cdot,\cdot)$ with respect to its first argument, defined in a point w in the direction v by

$$a'(w;v,\phi) = \lim_{\epsilon\to 0}\left[\frac{1}{\epsilon}\,(\,a(w+\epsilon\,v,\phi) - a(w,\phi)\,)\right]. \tag{5.29}$$

We have the following orthogonality relation for the error e:

$$\int_0^1 a'(u_h + t\,e; e, \phi)\,dt = a(u,\phi) - a(u_h,\phi) = 0 \qquad \forall\,\phi \in V_h. \tag{5.30}$$

This suggests the use of the following bi-linear form for the construction of the dual problem:

$$L(u, u_h; \phi, z) := \int_0^1 a'(u_h + t\,e; \phi, z)\,dt, \tag{5.31}$$

which is linear in ϕ and z.

For representing the error $J(e)$, we then use the dual problem consisting in finding $z \in V$ such that:

$$L(u, u_h; \phi, z) = J(\phi) \qquad \forall\,\phi \in V. \tag{5.32}$$

Assuming that this problem has a unique solution $z \in V$, and using the Galerkin orthogonality (5.30), we obtain the error representation

$$J(e) = L(u, u_h; e, z - i_h z), \tag{5.33}$$

with any approximation $i_h\,z \in V_h$ of z.

The goal is to evaluate the right hand side numerically, in order to get an *a posteriori* estimate for the quantity $J(e)$ and thus a criterion for the optimal local adjustment of the discretization. Since the bilinear form $L(u, u_h; \cdot, \cdot)$ contains the unknown solution u, it has to be approximated. The simplest way is to replace u by u_h yielding a perturbed dual solution $\tilde{z} \in V$ defined by

$$L(u_h, u_h; \phi, \tilde{z}) = J(\phi) \qquad \forall\,\phi \in V. \tag{5.34}$$

This approximation affects the quality of the resulting estimator

$$J(e) \approx \tilde{J}(e) := L(u_h, u_h; e, \tilde{z} - i_h\tilde{z}). \tag{5.35}$$

Controlling the effect on the accuracy of this approximated error estimator may be a difficult task and depends strongly on the particular problem considered. Many applications which may be found for instance in [33], [48] or [5] tend to suggest that the approximated estimator supplies correct information for the local mesh refinement process.

In a similar way as for linear systems in Section 5.2, an upper bound of $\tilde{J}(e)$ can be computed by solving the perturbed discrete dual system (5.34). The application of the Cauchy-Schwarz inequality on the cell-wise representation of equation (5.33) leads to an estimation of the error in the form

$$\tilde{J}(e) \leq \sum_{K \in \mathbb{T}_h} w_K \, \rho_K \tag{5.36}$$

with residuals ρ_K and weights ω_K.

In order to describe these coefficients, we take as example a part of the momentum conservation equation, for which the form a is defined as

$$a(u, \phi) = u \cdot \nabla u + \mu \, \triangle u. \tag{5.37}$$

The residuals and weights are then given by

$$\rho_K := h_k^2 \, \| u_h \cdot \nabla u_h - \mu \triangle u_h - f \| + \frac{1}{2} \, \mu \, h_k^{3/2} \, \| \, n \cdot [\nabla u_h] \, \|_{\partial K}, \tag{5.38}$$

$$\omega_K := \max \left\{ h_K^{-2} \| z - i_h z \|_K \, , \; h_K^{-3/2} \| z - i_h z \|_{\partial K} \right\}. \tag{5.39}$$

As before, we estimate the weights ω_K by the semi-norm $|z|_{K,2}$ which is again approximated numerically by the second-order difference quotient of the solution $z_h \in V_h$ of the discrete perturbed dual problem coming from (5.34),

$$\omega_K(z) \approx \tilde{\omega}_K(z_h) = C_K \, \| \nabla_h^2 z_h \|_K. \tag{5.40}$$

The resulting weighted-residual error estimator is

$$|\tilde{J}(e)| \leq \eta = \sum_{K \in \mathbb{T}_h} \eta_K, \qquad \text{with} \quad \eta_K = \tilde{\omega}_K \, \rho_K. \tag{5.41}$$

As a final remark it is to be noted that an approximation has been made in the bilinear form defining he dual system, in order to be able to write an upper bound of $J(e)$ which may be numerically computed. To keep a control on the accuracy of the process it may be worth to compare if the weights computed with the help of dual solutions on different meshes do not differ too much. In this case the error estimates are believed to be reliable. Otherwise one could attempt to refine the mesh globally in order to improve the global approximation of u and get less perturbed dual systems. This could be the case for very coarse meshes.

5.5 Application to Reactive Flows

We apply the weighted-residual error estimation described in the previous sections to reactive flow problems. The primal system is given by equations (2.59)-(2.63).

We denote the dual solution vector by

$$z = [z_u, z_p, z_w]^T. \tag{5.42}$$

We refer to Chapter 3 for the notation concerning the primal problem. For the sake of simplicity we do not take into account the stabilizing terms in the description of the dual problem. Their effect on the dual system is straightforward. The influence of these terms on the estimator itself will be mentioned later.

The derivation of the dual problem from the primal problem and the corresponding a posteriori error estimate follows the same line of argument as in the previous section. For economical reasons, we do not use the full Jacobian of the coupled system in setting up the dual problem, but only include its dominant parts. The same simplification is used in the nonlinear iteration process. Taking the notation of Chapter 1, the resulting dual problem is the following: find $z \in V = V_- \times V_+ \times Q \times S$ such that

$$\left.\begin{aligned} -(\rho\, u \cdot \nabla z_u, \phi) + (\mu \nabla z_u, \nabla \phi) + (z_p, \phi) = J_u(\phi) \quad \forall\, \phi \in V_- \times V_+, \\ -(\nabla \cdot z_u, \chi) + (\frac{u}{T} \cdot \nabla z_T, \chi) = J_p(\chi) \quad \forall\, \chi \in Q, \\ -(\rho\, u \cdot \nabla z_w^{(i)}, \psi) + (\rho\, D_i \nabla z_w^{(i)}, \nabla \psi) - \bar{P}(\psi, z_w) = J_w(\psi) \quad \forall\, \psi \in S, \end{aligned}\right\} \tag{5.43}$$

where the bilinear form $\bar{P}$ corresponds to a linearization of the chemical production term. The linear forms J_i defined on the solution space correspond to the functional of the solution for which we want to estimate the error. This system is supplemented by appropriate boundary conditions induced by those of the primal problem.

This problem has to be solved in order to evaluate the weights in the estimators of the resulting a posteriori error estimate

$$|J(e)| \leq \eta = \sum_{K \in \mathbb{T}_h} \sum_{X \in \{u,p,w_i\}} (\rho_{K,X} + \sigma_{K,X})\, \tilde{\omega}_{K,X}, \tag{5.44}$$

$\sigma_{K,X}$ representing the terms coming from the stabilization of the system. We sum over the error estimators corresponding to each component of the functional, since we may want to control the error on a functional depending on

several variables of the primal problem. The residuals $\rho_{K,x}$ involve the cells residuals and jumps of the discrete solution across inter-elements boundaries:

$$
\begin{aligned}
\rho_{K,u} &= h_K\, r_u + \tfrac{1}{2}\, h_K^{1/2}\, \mu \,\|\, [\partial_n u_h] \,\|_{\partial K},\\
\rho_{K,p} &= h_K\, r_p,\\
\rho_{K,w_i} &= h_K\, r_w^{(i)} + \tfrac{1}{2}\, h_K^{1/2}\, D_i \,\|\, [\partial_n w_h^{(i)}] \,\|_{\partial K},\\
r_u &= \|\rho\, u_h \cdot \nabla u_h - \nabla \cdot \mu \nabla u_h + \nabla p_h\|_K,\\
r_p &= \|\nabla \cdot u_h + L(u_h, w_h)\|_K,\\
r_w^{(i)} &= \|\rho\, u_h \cdot \nabla w_h^{(i)} - \nabla \cdot (\rho D_i \nabla w_h^{(i)}) - \nabla \cdot (\rho D_i M_i^{-1} w_h^{(i)} \nabla \overline{M}) - f_i(T_h, w_h)\|_K.
\end{aligned}
$$

As already mentioned, the weights $\tilde{\omega}_{K,x}$ are evaluated by solving the dual problem numerically and replacing the exact solution z by its numerical approximation z_h:

$$
\begin{aligned}
\tilde{\omega}_{K,u} &= C_K\, h_K\, \|\nabla_h^2 z_h^{(u)}\|_K,\\
\tilde{\omega}_{K,p} &= C_K\, h_K\, \|\nabla_h^2 z_h^{(p)}\|_K,\\
\tilde{\omega}_{K,w} &= C_K\, h_K\, \|\nabla_h^2 z_h^{(w)}\|_K.
\end{aligned}
$$

The error estimator for the complete stabilized system is derived from the estimator described just above and from the result of Section 5.3. The complete estimation is

$$
\begin{aligned}
|J(e)| \leq \eta &+ |c(p_h, u_h; z - i_h z)| + |s_u(p_h, u_h; z - i_h z)|\\
&+ \sum_{i=0}^{n_s} |s_i(p_h, u_h; z - i_h z)|,
\end{aligned}
\tag{5.45}
$$

η being the estimator without stabilization. The forms c, s_u and s_i are defined in Section 3.5.5 and correspond to the pressure and streamline-diffusion stabilizations. For each equation of our system we apply the process described in Section 5.3 in order to define an upper bound of the stabilization term. An upper bound of the error on the functional is then

$$
\begin{aligned}
|J(e)| \;\leq\; \eta_{\text{total}} = \eta \;&+ \sum_{K \in \mathbb{T}_h} \tilde{\omega}_{K,u}\, r_u\, \delta_K\, (1 + |u|_{\infty,K})\\
&+ \sum_{K \in \mathbb{T}_h} \tilde{\omega}_{K,w_i}\, r_w^{(i)}\, \delta_K^{(i)}\, |u|_{\infty,K}.
\end{aligned}
\tag{5.46}
$$

The most important aspect of this a posteriori error estimate is that the local cell residuals related to the various physical effects governing the flow and transfer of temperature and chemical species are systematically weighted according to their impact on the error quantity to be controlled.

5.6 Refinement Strategies

The right hand side in the error bound (5.46) can be evaluated once the finite element solutions u_h and z_h of the primal and dual problems have been computed and can be used to estimate the size of the global error $J(e)$. Exploiting this *a posteriori* error bound it is possible to adaptively control the global error to a desired tolerance level by suitably refining the mesh.

Let an error tolerance TOL and a maximum number of mesh points N_{max} be given. The goal is to find the most economical mesh $\mathbb{T}_h$ on which

$$|J(e)| \leq \eta(u_h) = \sum_{K \in \mathbb{T}_h} \eta_K \leq TOL, \tag{5.47}$$

with the local error indicators $\eta_K = \omega_K \, \rho_K$. The usual approach to constructing a mesh which does not contain an excessively large number of elements is to proceed iteratively: we start with a coarse mesh and refine it successively based on the size of the *a posteriori* error estimate. Inequality (5.47) can be thought of as a stopping criterion in this iterative process which can be written as follows:

1. Solve the discrete problem on $\mathbb{T}_h$.
2. Evaluate the estimator $\eta = \sum_{K \in \mathbb{T}_h} \eta_K$.
3. If $\eta > TOL$: change grid $\mathbb{T}_h$ according to η_K and go to 1.
4. end.

Starting from some initial coarse mesh, the refinement criteria are chosen in terms of the local error indicators $\eta_K(u_h, z_h)$. In fact various strategies can be adopted to generate a refined mesh from a given one (point 3 of the algorithm). Here we mention three of the most popular approaches (see [8], [48] or [33]):

- *Error-per-cell strategy*: In this approach the mesh generation aims to equilibrate the local error indicators by refining or coarsening the elements K in the current mesh $\mathbb{T}_h$ in order to reach the criterion

$$\eta_K \approx \frac{TOL}{N}, \tag{5.48}$$

  with N the number of elements in the resulting mesh. Since N depends on the result of the refinement decision, this strategy is implicit and requires an iterative implementation. However it is common practice to work with a varying value of N on each refinement level, with N successively updated according to the outcome of the refinement process. This strategy will deliver a partition on which $\eta \approx TOL$, provided that N_{max} is not exceeded. This refinement criterion leads to an equidistribution of the error over the whole mesh.

- *Fixed-fraction strategy*: In each refinement step, the elements are ordered according to the size of the local error indicator $\eta_K(u_h, z_h)$, and then a fixed portion of the elements K with largest $\eta_K(u_h, z_h)$ is refined (in two dimension typically 30% since this approximately doubles the number of cells in each refinement cycle). A smaller percentage of refined grid cells per adaptive step leads to a more localized refinements of the mesh. This process is repeated until the stopping criterion $\eta \leq TOL$ is satisfied or N_{max} is exceeded.

- *Fixed-reduction strategy*: We work here with a varying tolerance TOL_{var}. Having calculated the discrete solutions u_h and z_h on a mesh $\mathbb{T}_h$, the tolerance is set to $TOL_{var} = \sigma\,\eta$, where $\sigma \in (0, 1)$ is a fixed reduction factor. In the next step one or several cycles of the error-per-cell strategy are performed with tolerance TOL_{var}, yielding a refined mesh ${\mathbb{T}_h}^{new}$ and new solutions u_h^{new}, z_h^{new} with associated error estimator $\eta(u_h^{new}, z_h^{new})$. Then the tolerance is reduced again by setting $TOL_{var} = \sigma\,\eta$ and a new refinement cycle begins. This iterative process is repeated until $TOL_{var} \leq TOL$, or N_{max} is exceeded.

In each of the three strategies we repeat mesh modification followed by solution on the new mesh until the tolerance is satisfied, or the prescribed maximum number of elements is exceeded.

For our application to reactive flows in flow reactors, we used principally the second refinement strategy, which allows to tune the localization of the refinement zones. This generally leads either to meshes containing a smaller number of cells, since in less critical zones the error is allowed to remain over the bound prescribed in the first method, or to a better accuracy in critical zones. An application of the third refinement strategy can be found in [48].

Chapter 6

Applications

In this chapter we present four reactive flow problems with different complexities in the chemical reactions. The first two problems are based on CARS (Coherent Antistokes Raman Spectroscopy) measurements for the evaluation of the deactivation rate of vibrationally excited H_2 molecules. In a first example we take into account only the wall-deactivation process, which can be considered as a set of slow chemical reactions; 4 species and 7 reactions are involved in the chemical system. In a second example we consider the wall-deactivation process as well as the exchange of the vibrational energy of H_2 molecules with D_2 molecules. Here, the chemical system involves 9 species and the 32 reactions. A third example, again based on the same CARS flow tube, is the flow simulation of a mixture where chemical reactions between H_2, NO_2 and other produced molecules take place by higher temperature (from 300K to 1700K). Since the high temperature gradient within the flow causes numerical instabilities, a time step method has to be used here to be able to converge to a quasi-stationary solution. The chemical system considered involves 7 species and 6 reactions.

The fourth example is based on a CVD (chemical vapor deposition) experiment. We are interested in the deposition of diamond on the surface of a substrate. As revelator of this deposition we look at the concentration of CH_3 near the surface of the substrate. To improve the diamond deposition, this concentration must be as high and homogeneous over the substrate as possible. The chemical model involves 39 species and 358 elementary chemical reactions. Particularly with so many species and reactions, the application of the solution method developed in this work makes it possible to reach good accuracy with reasonable memory requirement and computation time. The calculation of such reactive flows can be performed by the adaptive algorithm presented in this work on a workstation or a PC.

6.1 CARS

6.1.1 Flow Reactor – Overview

The flow tube technique has importance in modern experiments as one of the most powerful tools for the determination of elementary chemical reaction rate constants.

The basic principle of flow tubes is always the same: mixing of reactants takes place upstream in a mixing section and their consumption or the buildup of products is followed along a measurement section by some detection method for atoms, radicals, or molecules. A reaction rate constant is then deduced from measured axial concentration profiles. In order to favor diffusive processes, which minimize radial concentration gradients, a flow tube is traditionally operated at low pressure. An assumed mean flow velocity allows to convert the axial coordinate (distance between the first point of mixing and the detection point) into reaction time. The reaction rate constants of interest can then be deduced by modelling the homogeneous reaction system. However, the method is known to bear systematic errors, since it is based on the approximation of a perfect decoupling of chemical and hydrodynamic process in the flow tube. Especially in the mixing section of the reactor this assumption is not valid.

In order to carry out a reliable evaluation of rate constants from experimental data, it is desirable to take into account all relevant physical and chemical processes occurring in a reactive flow. The detailed modeling of reactive flow fields within a reactor for kinetic studies is therefore an important tool for the experimental determination of elementary rate constants.

6.1.2 Reaction Kinetic of the $H_2 - D_2$ System

The heterogeneous relaxation and the exchange of vibrational energy of the H_2 molecules has been experimentally investigated in [57] with the help of a test reactor. For this experiment, based on the assumption of non-turbulent stationary flow and chemical process, the possibility of two-dimensional numerical simulation with a finite difference scheme has been studied in [46].

With the adaptive solution method developed in this work, we are able to get an accurate determination of some physical quantities of interest (such as mass fractions or concentrations) along the axis. These computational results can then be used together with experimental measurement results to get a good approximation of reaction rates for deactivation or exchange of vibrational energy for H_2 molecules. The automatic adaptive process refines the mesh only where it is needed (essentially on the measurement points and on singularities of the solution) to get accurate values on an optimal mesh, i.e. with a minimal

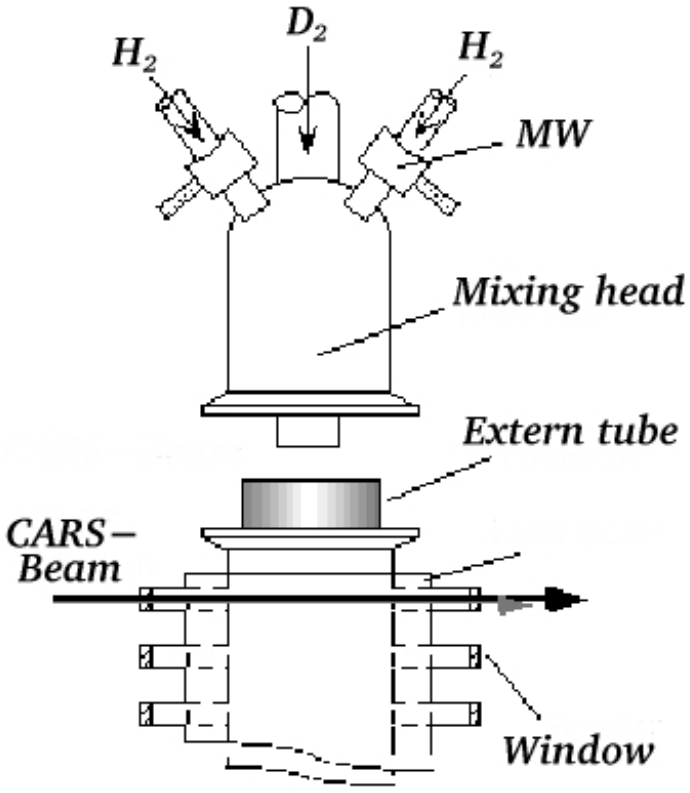

Figure 6.1: *CARS flow reactor.*

number of mesh nodes for a given precision. In this way we not only save CPU-time but we also gain in accuracy, being assured of the precision on the computed quantities.

The reactor considered here consists essentially in the concentric disposal of an external tube (radius 16 mm) in which an interior tube (internal radius 5.5 mm and wall thickness 1 mm) hands in (see Fig. 6.1).

Two gases streaming out of the outer and interior tubes get in contact at the outlet of the central tube. This central tube is long enough to guarantee fully developed laminar flow fields for both inner and outer gas flows. From this point on, the gases are mixed through convective and diffusive transport and may react with each other. The main tube (the prolongation of the outer tube) consists of a straight 32 mm diameter section equipped with an array of diametrically opposed 2mm diameter holes in the wall to allow optical CARS diagnostics with focused laser beams. In this way, it is possible to record axial profiles for species concentrations. A complete description of the experiment can be found in [57], [47] and [46].

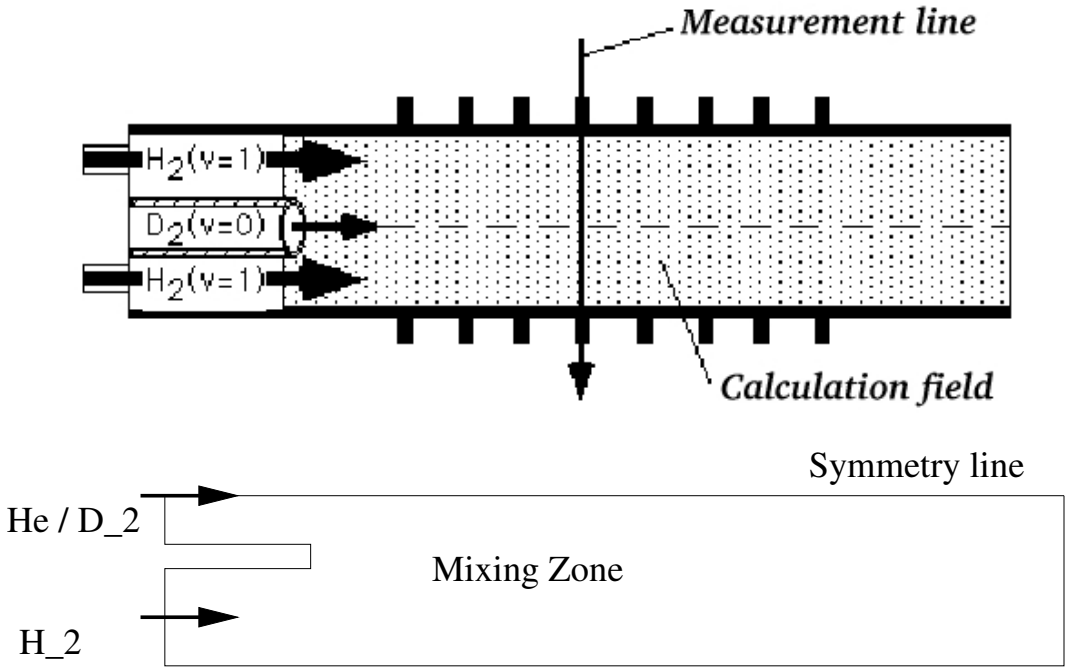

Figure 6.2: *Two-dimensional field with measurement-line positions and calculation field (half domain for symmetry reasons).*

Vibrationally excited hydrogen molecules $H_2(v'' = 1)$ are generated by microwave discharges (MW – see Fig. 6.1) in the sidearms of the mixing head bringing the gas to the outer tube. The microwave discharges create also H atoms. These atoms lead to additional reactions which the modelling of the process must take into account. In the inner tube, Helium He or in-the-ground-state Deuterium $D_2(v'' = 0)$ are injected.

The wall vibrational relaxation rate γ_{wall} for the deactivation of $H_2(v'' = 1)$ to $H_2(v'' = 0)$ and the vibrational energy transfer rate of $H_2(v'' = 1)$ in collisions with $D_2(v'' = 0)$ are the unknown reaction kinetic constant which have to be calculated.

6.1.3 First Evaluation: Wall Relaxation

We investigate the deactivation of vibrationally excited hydrogen molecules at the wall (heterogeneous relaxation). An inert gas (Helium) is used as carrier gas. It is streaming into the mixing tube from the internal tube. We consider the laminar flow for determining the reaction rate of the elementary wall-deactivation reaction (slow chemistry):

$$H_2^{(\nu=1)} \quad \xrightarrow{\text{wall}} \quad H_2^{(\nu=0)}. \tag{6.1}$$

Table 6.1: Simulation results for the $H_2(\nu = 1)$ wall-deactivation experiment on hand-adapted(top) and on automatically adapted (bottom) meshes.

Heuristic-based refinement			
Level	# Cells	H2(0)	H2(1)
1	137	0.6556	0.005294
2	481	0.7373	0.00661
3	1793	0.7962	0.007096
4	6913	0.8172	0.007434
5	7042	0.8197	0.007419
6	7494	0.8240	0.007473
7	8492	0.8269	0.007504
8	10482	0.82858	0.007521
9	15993	0.82853	0.007545

Error-estimator-based refinement			
Level	# Cells	H2(0)	H2(1)
1	137	0.6556	0.005294
2	282	0.7382	0.006063
3	619	0.7958	0.007132
4	1368	0.8149	0.007323
5	3077	0.8257	0.007457
6	6800	0.8295	0.007534
7	15100	0.8317	0.007564
8	33462	0.8328	0.007587

The complete reaction mechanism can be found in the appendix.

The unknown is the kinetic reaction constant, i.e. the wall relaxation rate γ_{wall} for the reaction described just above. A definition of γ_{wall} is given in Section 2.3.2. The quantities to be computed are the results of CARS measurements of species concentrations. The measured quantities are proportional to a weighted mean value of the mass fractions w_i along lines perpendicular to the symmetry axis of the reactor, and are used to obtain approximations of the species concentrations along the axis of the tube.

We will present the computed mean values of the mass fractions of activated and deactivated hydrogen along radial lines Γ of the two-dimensional calculation field. The error functional (see Chapter 5) used in the adaptive process is

$$J(\varphi) = \int_{\Gamma} \varphi(r, z)\, dr. \tag{6.2}$$

In order to emphasize the advantages of the method presented in this work, we also have computed the averaged mass fractions on tensor product meshes which are *a priori* refined on the basis of heuristic considerations. This process is only based on the *a priori* knowledge of the measurement lines which are considered to be the refinement lines. We begin with global mesh refinement and then go on with local refinement along the measurement lines as well as on the known singularity of the solution.

Comparison of results shows that the refinement based only on heuristic criteria is not sufficient to get reliable values from the computed solution. Table 6.1 shows the values of the average of the H_2 mass fractions along a cross section of the tube for a simulation firstly with the heuristic method and secondly with the error-estimation method.

We observe improved accuracy on the automatically adapted meshes for about the same number of grid points. In particular, monotone convergence of the quantities of interest is achieved. This is an important feature of our approach which provides high reliability of computed solutions.

Corresponding solutions and meshes are shown in Figures 6.3, 6.4 and 6.5. For the meshes refined with the use of an error estimator, the structure of the dual solution reflects the dependence of the quantity $J(X)$ (the error functional) on the local cell-residuals.

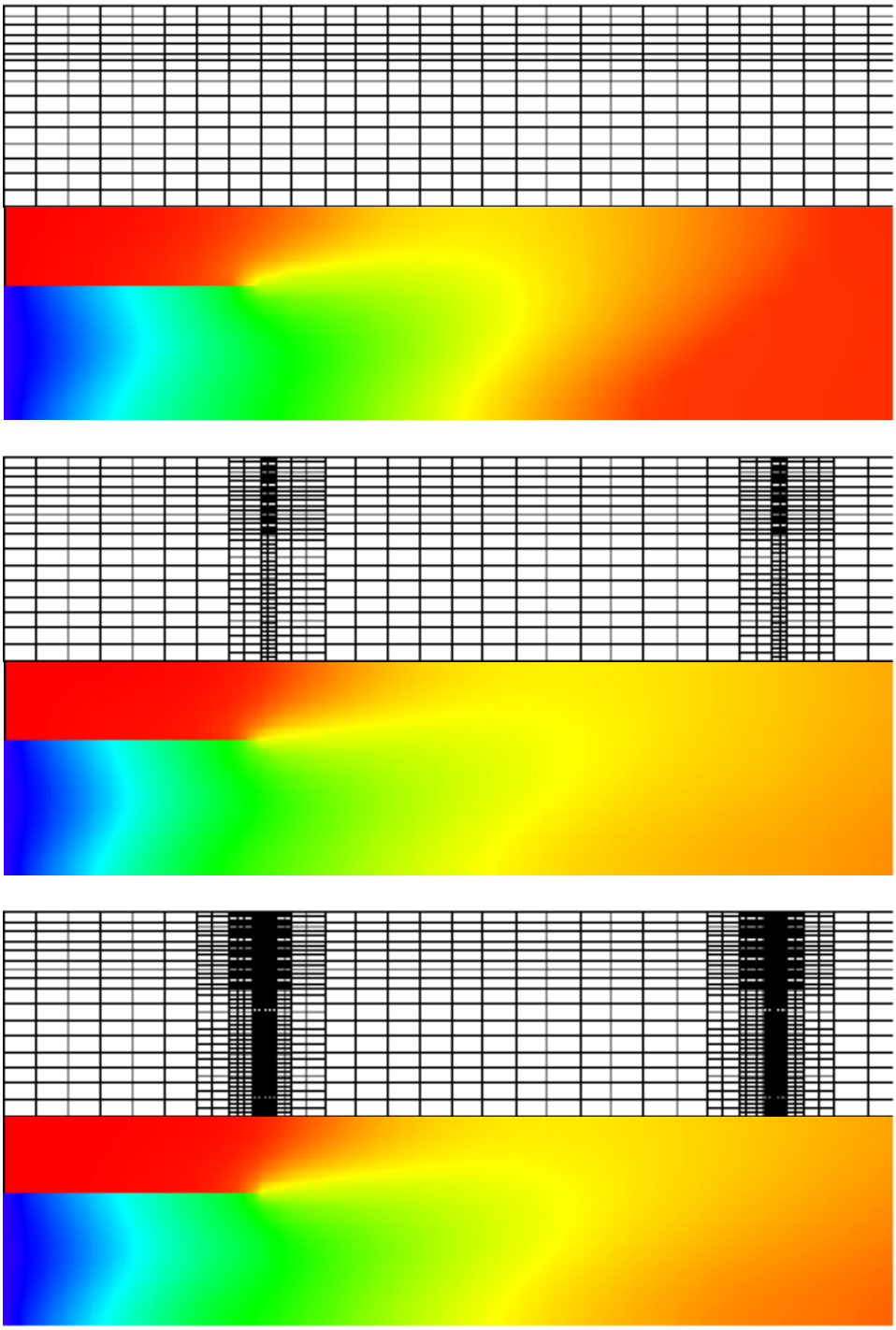

Figure 6.3: *Mass fraction of* $H_2^{(\nu=1)}$ *by the CARS simulation with heuristic refinement – Refinement levels 2, 4 and 6.*

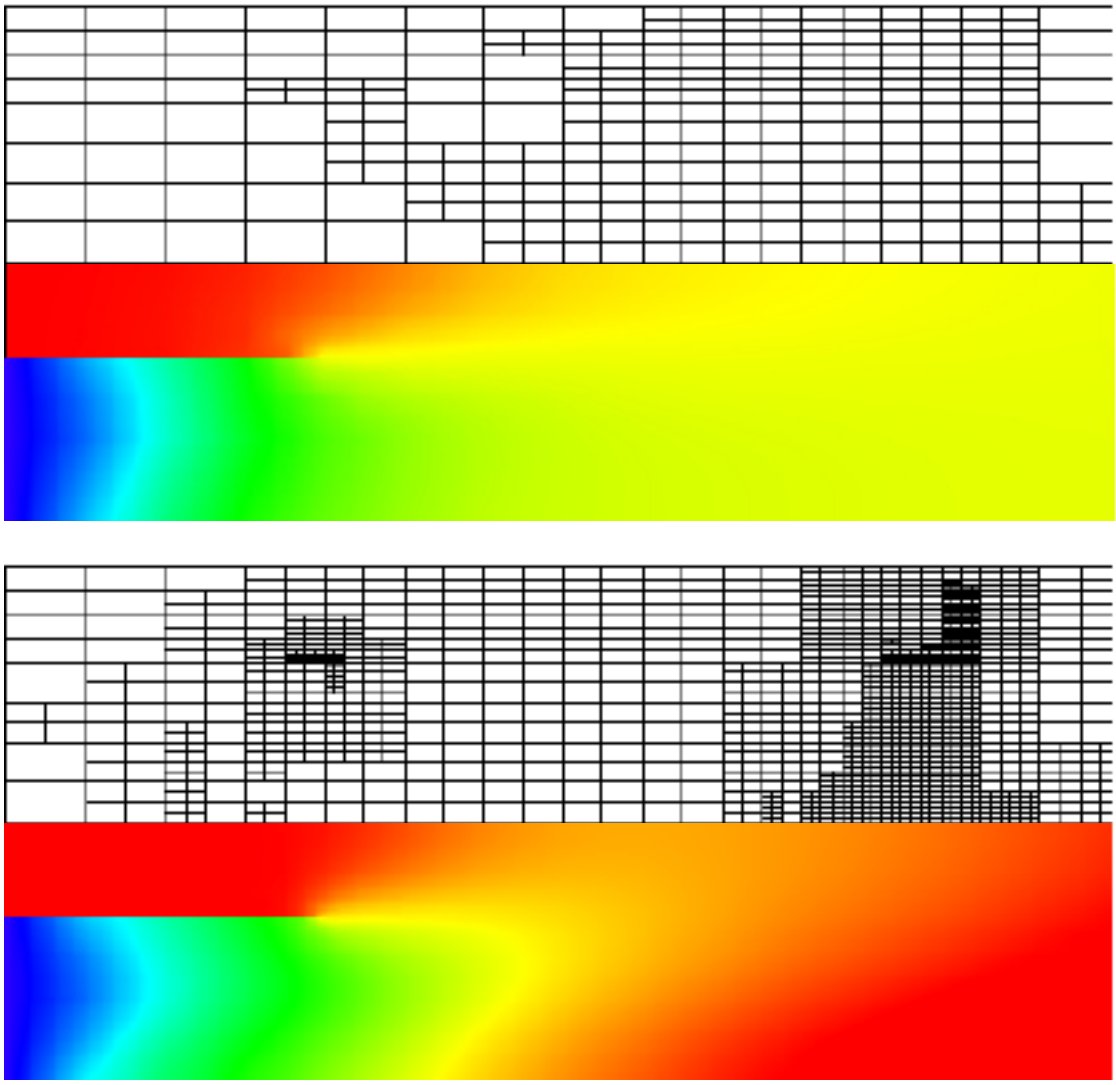

Figure 6.4: *CARS simulation with adaptive local refinement – Mass fraction of* $H_2^{(\nu=1)}$ *– Refinement levels 2 and 4.*

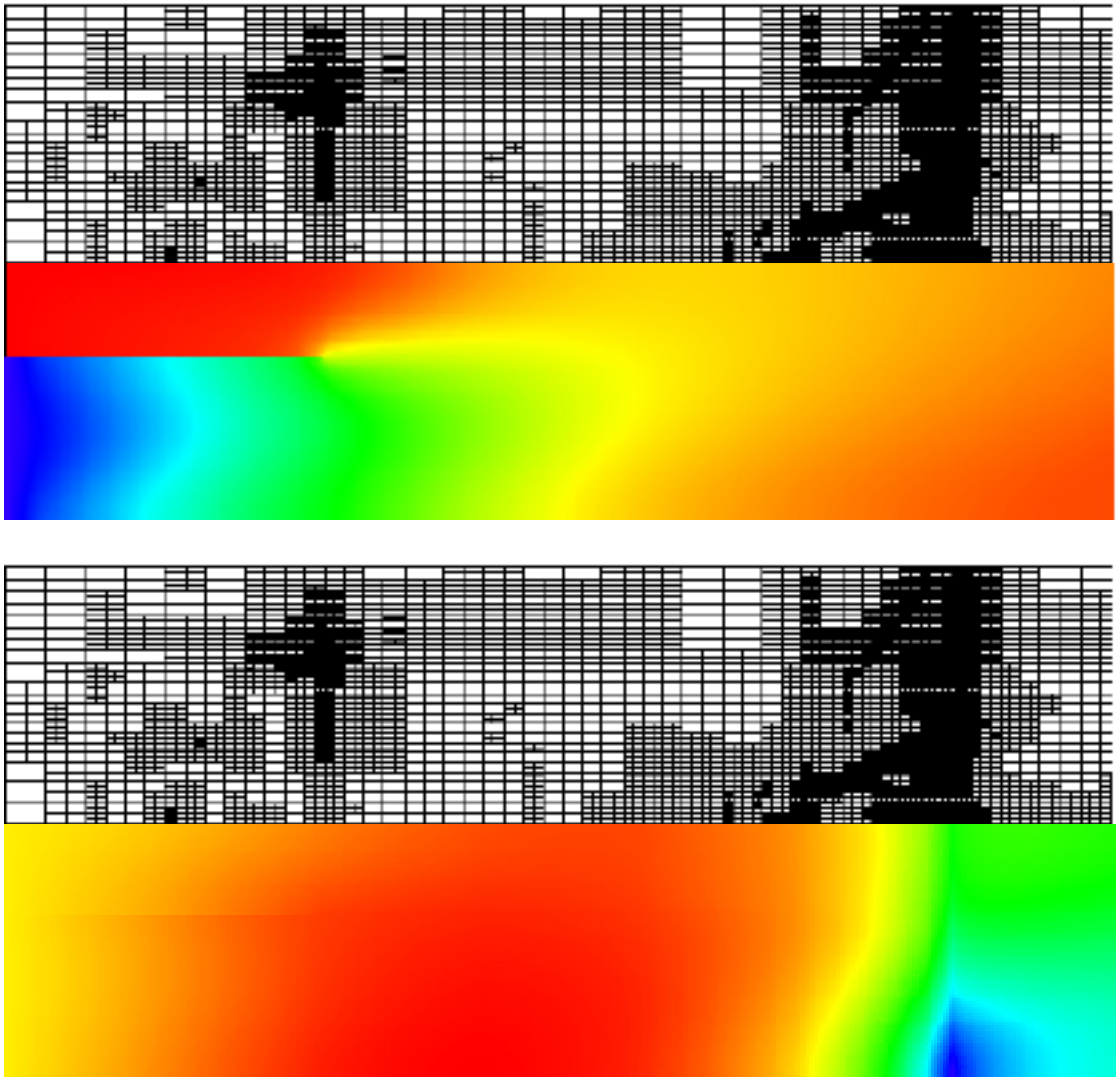

Figure 6.5: *CARS simulation with adaptive local refinement – Mass fraction of* $H_2^{(\nu=1)}$ *(top) and dual solution component corresponding to* $H_2^{(\nu=1)}$ *(bottom) – Refinement level 6.*

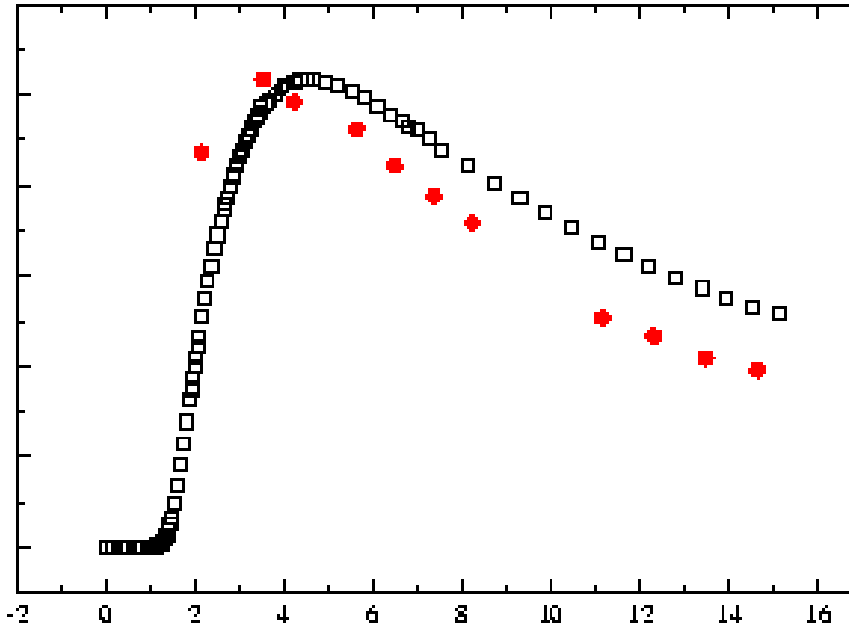

Figure 6.6: *CARS comparison between normalized simulation results (squares) and experiment measurements (points) for the evolution of the* $H_2^{(\nu=1)}$ *mass fraction along the axis.*

Our computational results have been compared to experimental measurements (see Figure 6.6). For this computation, the inflow rate for the helium which flows from the interior tube is set to 148 l/min and the inflow rate for the hydrogen which flows in the outer tube is set to 665 l/min. The thermodynamical pressure is considered to be 5.33 mbar and is constant in the whole domain. The proportion in mole of the vibrationally excited H_2 molecules at the inflow is 0.5%, the proportion of H atoms is 0.3% and the rest 99.2% is non vibrationally-excited H_2 molecules. The experimental measurements have a relative error of around 20%.

Such comparisons make it possible to approximate the deactivation rate of H_2 molecules. At the present time we have to tune manually the value of the corresponding reaction rates which we want to evaluate. A further development should be to couple the solution method with an optimization process in order to find the best approximation of the reaction rate with regard to the comparison between simulation and experiment.

As pointed out in Chapter 4, we also want here to show how different the convergence process can be when using different Jacobian matrix approximation. This shows that the convergence criterion has to be chosen carefully and a residuum drop which could seem to be sufficient to get a correct approximation of a Navier-Stokes flow may be insufficient for flows with chemical

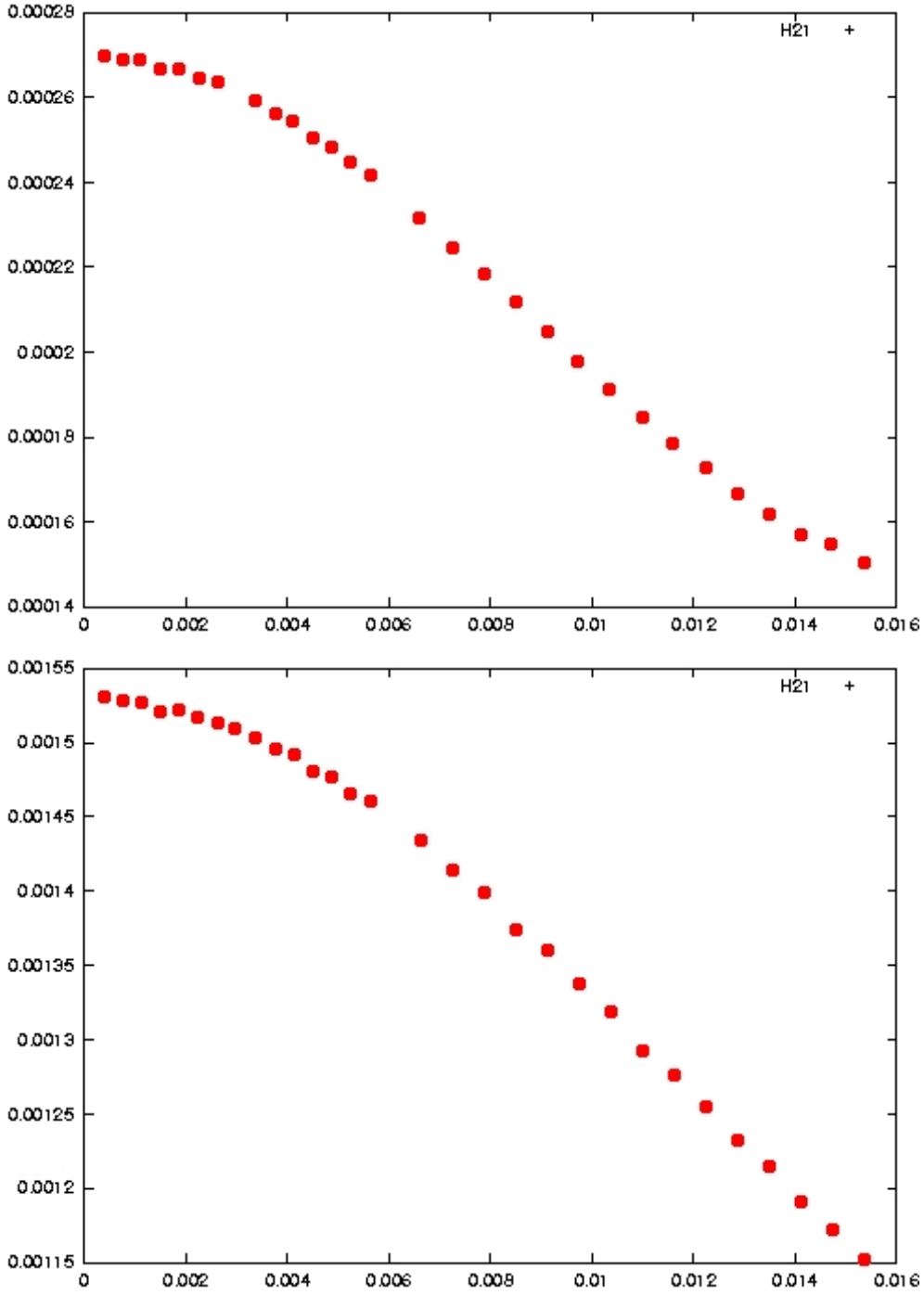

Figure 6.7: $H_2^{(\nu=1)}$ *mass fraction along a radial section at axial position 0.143 m in the CARS flow reactor. Comparison between a calculation with a Jacobian matrix taking surface reaction terms into account (above) and a calculation with an approximated Jacobian matrix (below) by a convergence with a tolerance of* 10^{-8} *on the residuum.*

reactions. We compare here two approximations of the Jacobian matrix, the first one taking into account all the chemical terms, the second one without the surface reaction terms. We want to remind the user that these terms are still taken into account in the residuum term of the defect-correction method.

For a convergence with a tolerance of 10^{-8} on the residuum, we see in Figure 6.7 that the approximated Jacobian did not allow to get a correct approximation of the solution at this point in the convergence process. The surface reactions are not yet caught by the solver and the value on the wall surface of the mass fractions for $H_2^{(\nu=1)}$ obtained with the help of the approximated Jacobian is higher than the one obtained with the Jacobian taking into account all chemical terms. While reaching a residuum of 10^{-9}, the obtained convergence leads in this case to the same results for both methods.

This means that the correct evaluation of chemical process may occur only late in the convergence process. Moreover we also have to be aware that using approximated Jacobian may in some cases lead to problems in catching all chemical processes in the solution (and thus get convergence) since we have to converge with a very small tolerance on the residuum. We actually did not experience such a problem in our applications and with the approximations of the Jacobian matrix we used (see Chapter 4).

6.1.4 Second Evaluation: Wall Deactivation and Activation Transfer

In this experiment helium is replaced by deuterium. Thus this latter gas is added through the central tube while vibrationally-excited hydrogen enters through the outer tube. We have here to take into account some more elementary reactions such as

$$H_2^{(\nu=1)} + D_2^{(\nu=0)} \quad \longrightarrow \quad H_2^{(\nu=0)} + D_2^{(\nu=1)}. \tag{6.3}$$

The complete reaction mechanism used for this computation can be found in the appendix.

Both hydrogen and deuterium are experimentally monitored in their first excited vibrational state. Therefore, in the simulation, we may be interested in the average of $H_2^{(\nu=1)}$ or of $D_2^{(\nu=1)}$ mass fractions along radial lines in the two-dimensional calculation domain. As in previous section, we could construct the corresponding functionals given by (6.2) for both species and use them for the definition of the error functional of the adaptive method. Another possibility is to take as error functional the sum of the error functionals corresponding to the mass fractions of interest (i.e. for which measurements are done).

Table 6.2: Performance comparison between the simulation code developed in this work and based on the DEAL library and a finite difference code developed by J. Segatz in [46].

code	CPU time (units $\sim$ sec.)		memory required	
	global	per vertex	global	per vertex
Waguet 13442 vertices	9360 ($\sim$ 2,5 h.)	0.70	63 Mb	4.7 Kb
Segatz 16000 vertices	85750 ($\sim$ 24 h.)	5.35	153 Mb	9.5 Kb

However, in order to demonstrate the flexibility of the adaptive method based on error estimates and duality arguments, we use here a different error functional. The CARS signal delivers the value of a weighted integration along radial lines in the tube and we had consequently taken this functional in the previous simulation. But we are actually interested in the value of concentrations along the axis. The numerical simulation allows direct access of point values of the concentrations. Therefore the functional could be chosen as

$$J(\varphi) = \varphi(r_0, z_0), \tag{6.4}$$

with $r_0 = 0$ and z_0 the coordinates of the point of interest along the axis. For the following results we took as error functional for the complete system the sum over error functionals defined as above for several species and several axial coordinates.

We see in Figure 6.8 that the automatic adaptive refinement process leads to mesh refinement on given points (r_0, z_0) but also on the zones where the reactions may strongly influence the evolution of species concentrations along the whole tube or also in the zones where the solution may have a singularity as on the top of the splitter plate.

The method described in this work requires less CPU-time and memory for the calculation of the steady state of reactive flows compared to other existing finite difference methods based on tensor product meshes. Table (6.2) shows the comparison between the simulation code developed in this work and a finite-difference code already successfully used for simulation of flow reactors developed in [46] by J. Segatz.

Considering the performance measurement for the code developed in this work, we see that the CPU time needed to attain convergence has been reduced by a factor 7 with regard to the other code, and that the memory requirement has been reduced by a factor 2. And this, without taking into account the advantages of the local refinement process. The gain in performance allows us to apply the method on more complex systems with finer (locally refined)

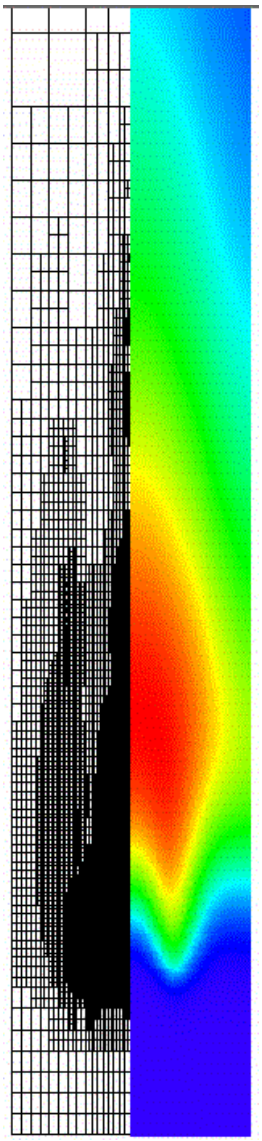

Figure 6.8: *CARS simulation with local refinement and point error functional – Mass fraction of* $HD^{(\nu=1)}$.

grids and still compute the solution on a workstation or a PC, as we also see in the following section.

6.1.5 NH-$NO2$ Chemical System

The main goal of this experiment is the direct measurement of reaction rates as well as the examination of their temperature dependency in the range of high temperature (300K - 1700K). The experimental material is the same as for CARS measurements: a flow reactor with an inner and an outer tube from which flow different gas which then react with each other in the mixing zone of the tube. One difference is that the walls are heated and thus have a given temperature. The simulation of high temperature flows is used for interpretations of experimental measurements of reaction rates as well as for investigations on their temperature dependence.

As a first step toward the computation of the complete reaction mechanism, we compute a high temperature flow reactor with a mixture consisting of H_2, NO_2 and He molecules which produces through chemical reactions OH, NO and H_2O molecules as well as H and O atoms. We use as error functional the global mean value of the NO concentration.

The solution process we used here for converging to a quasi-stationary solution is the following:

- We compute the reactive flow on a coarse grid which however is fine enough to allow to capture the principal structures of the flow and chemical reactions. Typically numerical tests showed that, for this kind of flow, a coarse grid with around 100 cells is sufficient. The quasi-convergence of the time-step process is reached as soon as the residual difference between two following time-steps is smaller than a given tolerance.
- Once a quasi-stationary solution is reached on this coarse grid we refine it locally using an error estimator.
- We compute further time steps and refine again the grid locally as soon as the quasi-convergence condition has been reach for the time step process.
- We repeat the third point until the value of the error functional reach a given tolerance.

We show in Figures 6.9 and 6.10 respectively the time evolution of the NO molecule and the O atom mass fractions within the computation domain which represents the half of an axial section of the flow tube. From the inner tube flows a mixture of NO_2 and He molecules with a mole fraction distribution respectively of 0.44 and 0.56 and with a maximal velocity of 30 m/s. From

the external tube flow H_2 molecules with a maximal velocity of 20 m/s. The pressure of the inflow is 5 mbar and the temperature 300K.

This computation is the first step toward the simulation of the complete reaction mechanism which was not available at the time of the calculation. With the help of simulation, we are able to test several mechanisms and investigate the temperature dependence of the different reaction rates which are taken into account, by comparing the simulation results, e.g. concentrations of some species, with experimental measurements of these concentrations.

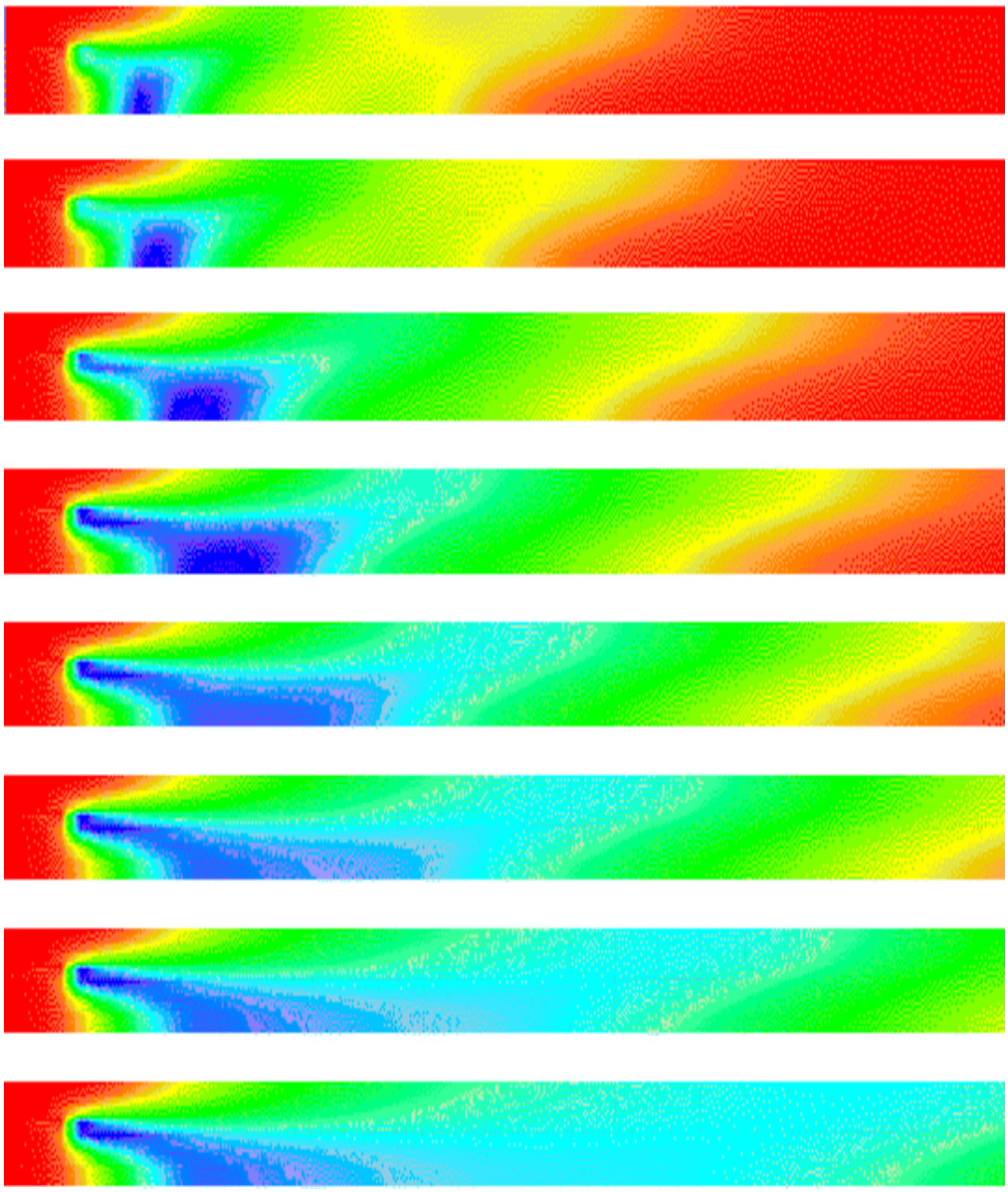

Figure 6.9: *Time evolution of NO mass fraction for an inflow of NO_2 molecules in the outer tube and of H_2 molecules in the inner tube– red represents a null mass fraction and blue represents a maximal mass fraction for this molecule*

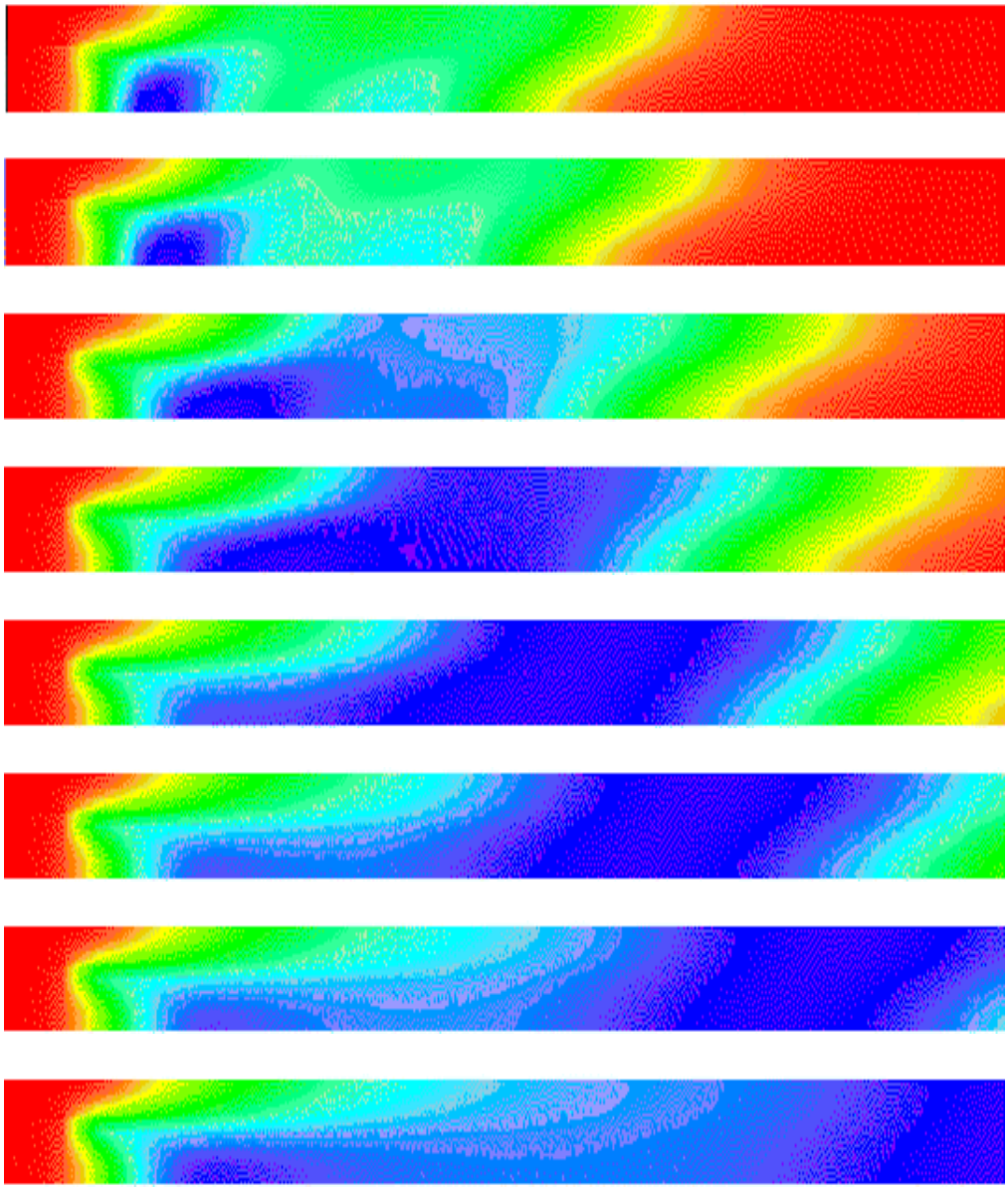

Figure 6.10: *Time evolution of O mass fraction for an inflow of NO_2 molecules in the outer tube and of H_2 molecules in the inner tube – red represents a null mass fraction and blue represents a maximal mass fraction for this atom.*

6.2 CA-CVD

In a "Chemical Vapor Deposition" (CVD) reactor, diamond can be deposited upon different materials from an hydrocarbon-hydrogen gas mixture under moderate temperature and low pressure. Improvement of the growth rate and the quality of the produced diamond layer as well as its homogeneous growth are some of the aims which are still to be reached in this field. The comprehension of the reactions on the substrate where the diamond layer settles is still incomplete. Even the species which control the deposition kinetic have not been incontestably found and the complex chemical mechanisms are not sufficiently known.

For a deeper understanding of the complex relations between gas phase and surface chemical processes and hydrodynamical processes, simulations must complement the experiments and supply a base for evaluating several models of chemical processes.

The reactor is made of a 15cm-diameter tube with a height of 20cm. The geometry of the reactor used for the experiment is axially symmetric, which makes the two-dimensional modelling possible. The reactor has three windows for the inspection of the gas composition through the detection of fluorescent light created with the help of a laser beam (see Fig. 6.11). The reader can find a comprehensive description in [23]. The pressure in the reactor is set to 50 mbar with the help of an automatically-regulated pump.

The chemical radicals which are necessary for the diamond deposition upon silicium substrates are produced injecting methane into the combustion gas of a H_2/O_2 flame. The term used for this process supported through combustion is "Combustion Assisted - Chemical Vapor Deposition", in short CA-CVD. Hydrocarbon molecules are transformed during the chemical process in reactive radicals, which depose on the substrate with the adequate crystal structure in form of diamond.

As noted above, the detailed steps of the process are not completely understood yet. However the methyl-radical (CH_3) seems to have an important role in the formation of diamond. The corresponding experimental conditions have to be set such that a suitable temperature as well as a high concentration of CH_3 molecules are found in the close proximity of the substrate surface. Methyl is created through the decomposition of methane or higher hydrocarbons. The mixing of a hot-flame exhaust gas with high concentration in hydrogen radicals with hydrocarbons leads to chemical reactions such as:

$$CH_4 + H \quad \longrightarrow \quad CH_3 + H_2. \tag{6.5}$$

The structure of the experiment is shown in Fig. 6.11. A hydrogen/oxygen flame (premixed) burns above a burner. Its exhaust gas contains beside the

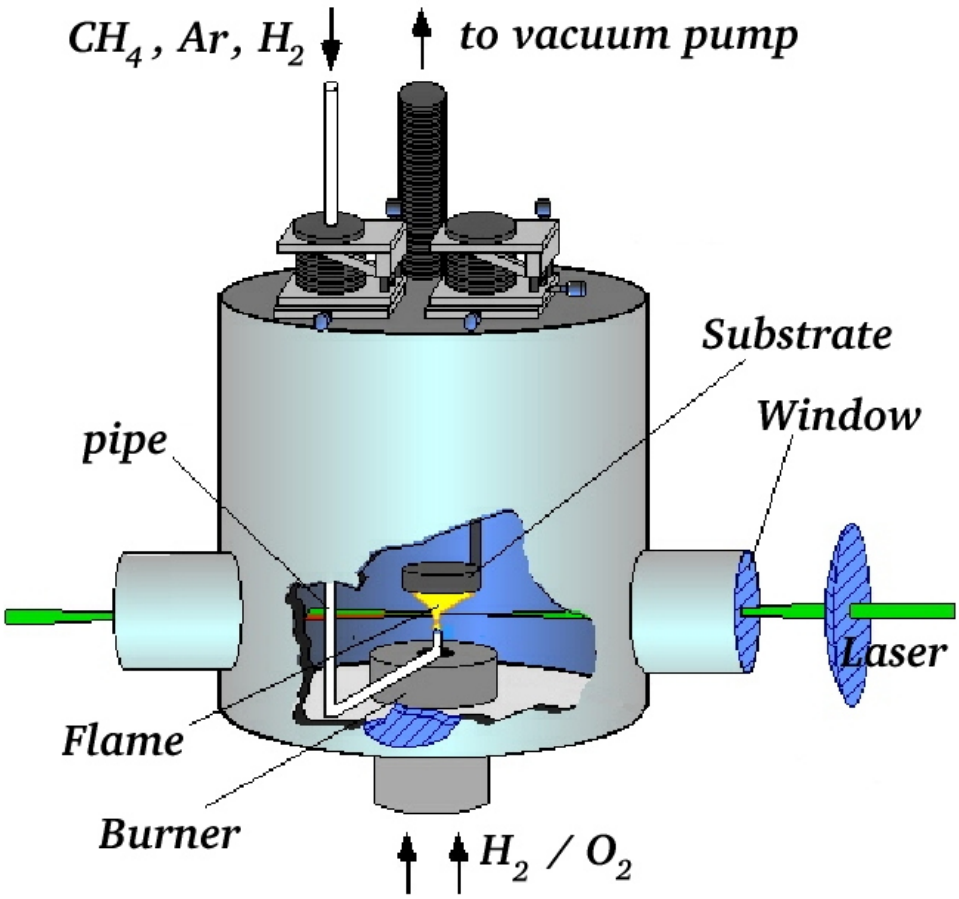

Figure 6.11: *CVD flow reactor.*

combustion product H_2O, also up to 25% H radicals (in mole) and contribute to the warming of the methane injected through the pipe. This latter gas is then transported by convection and diffusion within a "stopping-point" flow to the substrate surface. Decomposition reactions occur on the way, such that the CH_3 concentration increases at first by the consumption of H radicals, and finally decreases due to recombination and other reactions.

This later process can also be observed in the result of the simulation (see Figure 6.13). With the help of the adaptive solution method developed in this work, the concentration of CH_3 can be accurately computed. In order to optimize the CH_3 concentration on the substrate we could use for our adaptive process an error functional similar to the functional described in Section 6.1.3 and defined by relation (6.2). However as we want here to show the evolution of the CH_3 mass fraction in the reactor, we decide to use a global error functional

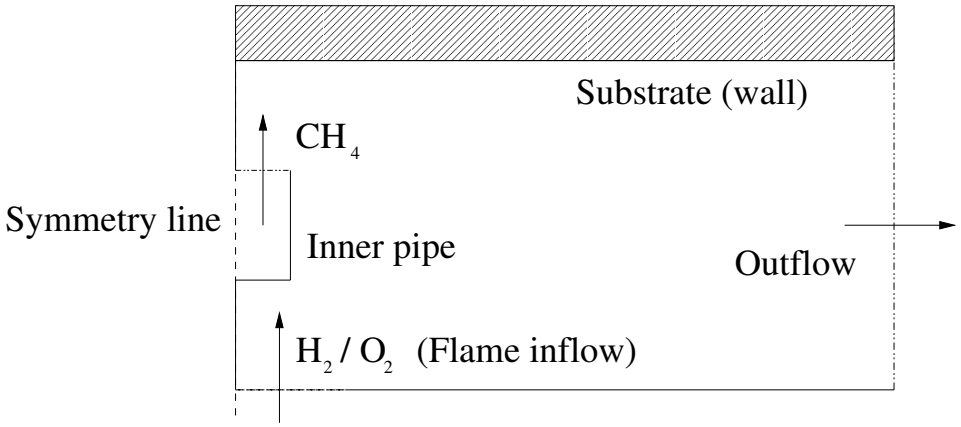

Figure 6.12: *CVD computational field – half axial section of the CVD reactor for symmetry reasons.*

on this variable defined by

$$J(\varphi) = \int_{\Omega} \varphi(r,z)\, r\, dr dz. \tag{6.6}$$

This gives us control on the mean value of the CH_3 mass fraction over the whole domain (see Chapter 5 for more details about error functionals) and makes the adaptive process refine more globally where the gradient of this variable is high or on some singularities and not on given measure points or lines (see Figure 6.13). Table 6.3 gives the convergence history of the error estimator based on the resolution of the dual system and defined in (5.46).

Table 6.3: Results for the error estimator for the CVD simulation using as error functional the global mean value of the CH_3 mass fraction.

Level	# Cells	η
1	412	4.21e-5
2	784	1.70e-5
3	1528	7.49e-6
4	2941	3.44e-6
5	5698	2.05e-6
6	11374	1.14e-6
7	23611	6.43e-7

A next step would be to optimize the CH_3 concentration on the substrate surface by controlling parameters such as the inflow velocities of the gas or the geometry. In this purpose, we would use an error functional giving control

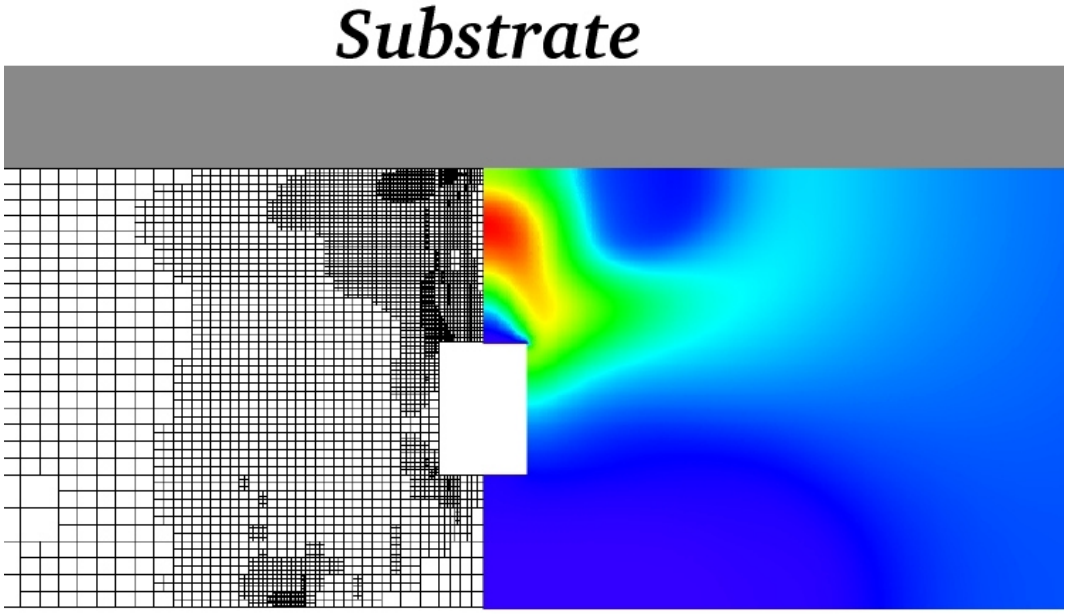

Figure 6.13: *CVD simulation with local refinement – Mass fraction of CH_3.*

to the local value of the CH_3 concentration or of the concentration of any other species involved in the diamond deposition on the substrate. Once the optimized parameters are found by simulation, they can be applied on the experiment.

The different parameters which can be used for the optimization process can be the methane flow rate or the flame exhaust gas flow rate as well as the distance between the pipe from which methane flows and the substrate. These are two different kinds of parameters: the first one involves boundary conditions, the second one the geometry of the reactor.

To simplify geometrical optimization, if we decide to optimize the distance pipe/substrate, an automatic mesh generator has been developed. It allows the user to generate a mesh for the computation domain according to geometrical parameters such as the pipe distance to the substrate and to the flame, as well as the reactor size and the pipe size (see description in Appendix C).

In a further work we could also here couple the solution process developed in this work with an optimization process for instance on the inflow boundary conditions for the inflow velocities or species concentrations. Promising results in this field can be found in [34].

Chapter 7

Conclusion and Outlook

In this work, we have developed and implemented a solution method for the low Mach-number formulation of the Navier-Stokes equations with supplementary equations describing the evolution of the temperature and chemical species (mass fractions) with source terms due to heterogeneous (surface) and homogeneous (gas-phase) chemical reactions. These equations are written in cylinder coordinates and are discretized with stabilized conforming Q1/Q1 finite elements.

The resulting nonlinear system is solved by a full-coupled defect-correction iteration based on an approximation of the Jacobian matrix of the system. We construct this approximation with regard to the consistence and solvability of the corresponding linear system.

A key element of the solver is the use of a multigrid preconditioner for the GMRES method applied for solving the linear problems arising in the defect correction iteration. We implemented three different smoothing operators for our multigrid preconditioner: a Gauss-Seidel iteration and a robust ILU factorization for the species equations, and a Vanka-type smoother for the Navier-Stokes part of our system. The multigrid method we implemented is based on the DEAL library and takes advantage of the hierarchical structure of the mesh constructed by successive refinements.

Adaptive meshes are successfully applied in the context of reactive flows. A recent approach to control the error in functionals of the solution is presented and applied to this type of problems. The reliability and efficiency of the error estimator for our applications is demonstrated through numerical results for two types of chemical models.

Comparing our method with a finite-difference code developed by J. Segatz and used in the computation of chemical flow reactors (see [46]), the calculation time has been reduced by a factor five for reaction mechanisms made of around 30 elementary reactions and involving around 10 species. We have also successfully applied our method to chemical flows involving 39 species and more than

350 chemical reactions. Even by flows with so many species and reactions, the adaptive method presented in this work allows to reach a controlled accuracy on physical quantities of the flow with acceptable computational efforts.

As promising outlook we would like to emphasize the following points:

For large chemical systems a major part of the computing time is consumed by the calculation of the Jacobian matrix of the chemical source terms and its inversion by Gauss-Seidel iterations or ILU factorization. Because these operations can be performed locally, a parallelization of the presented algorithm seems to be an adequate method.

An application to 3D problems will also increase the need of reducing memory requirements and computation times without sacrificing accuracy. Adaptive refinement methods will probably play an important role for solving 3D problems in order to reach the needed accuracy on physical quantities of interest with an optimal number of cells.

Another field of investigation is the mesh adaption for unsteady solutions. A first approach is to allow beside mesh refinement also mesh coarsening, and compute a locally-refined mesh for each time step. Research is still needed to implement a complete mesh refinement strategy for solutions depending on time. Moreover a refinement strategy for the time steps can also be defined.

Another promising perspective is the application of error control and adaptivity processes for finite element discretization to optimization problems governed by differential equations. The dual solution obtained during the adaptive mesh refinement can be used to build optimization strategies. This allows to control the value of the cost functional of the optimization problem. Some results in the field of coupling adaptivity and optimization methods can be found in [7] and [34]. As an example of possible optimization problem we want to give the diamond deposition seen in Section 6.2: by optimizing some species concentration on the substrate, the quantity and quality of the diamond layer over the substrate can be drastically increased.

7.1 Acknowledgment

I want to thank Prof. Dr. R. Rannacher for the opportunity to realize this work within his team, as well as for his help and support.

I am also deeply grateful to R. Becker, G. Kanschat and especially F.T. Suttmeier for their valuable mathematical ideas and suggestions, as well as for their help concerning the implementation of the methods presented in this work. Some numerical algorithms as well as the mesh handling are based on the DEAL library written by them.

I also thank Malte Braack for his help concerning the chemical aspects of this work, and mainly for his help concerning the implementation of the chemical model described in this work which is largely based on a C++ library written by him.

Stefan Appl, Wolfgang Bangerth and Vincent Heuveline have taken part in the correction of this manuscript. This help is gratefully acknowledged by the author.

Appendix

Appendix A

CARS-Experiment reaction model

- H_2 wall relaxation process

```
MECHANISM OF H2(V=0,1) REACT. (Yorck Schneider-Kuehnle)
****
*****************************
****                         *
****  1. H2-HE MECHANISM     *
****                         *                * reaction rates *
*********************************************
H21     +H       +        >H20     +H       * 2.36E+11   0.00      0.0
H21     +H20     +        >H20     +H20     * 6.50E+07   0.00      0.0
H21     +HE      +        >H20     +HE      * 1.56E+07   0.00      0.0
H       +H       +HE      >H20     +HE      * 5.00E+16   0.00      0.0
H       +H       +H20     >H20     +H20     * 2.90E+15   0.00      0.0
*********************************************
END
COLLISION EFFICIENCIES
END
COMPLEX REACTIONS
002  COMPLEX REACTIONS AT THE WALL
 1.00 H21     *1.0                                          1.500E-03
 1.00 H20                                                   0.0  0.00
 1.00 H       *1.0                                          1.000E-04
 0.50 H20                                                   0.0  0.00
END
```

- H_2/D_2 wall relaxation process and vibrational energy exchange

```
MECHANISM OF H2(V=0,1) REACT. (T.DREIER)
****
```

```
*****************************
****                         *
****  1. H2+D- MECHANISM     *
****    k = cm3/mol/s        *              * reaction rates *
*****************************************
H20     +D                >HD0     +H       * 1.78E+08   0.00      0.0
HD0     +H                >H20     +D       * 2.03E+07   0.00      0.0
D20     +H                >HD0     +D       * 1.27E+07   0.00      0.0
HD0     +D                >D20     +H       * 2.03E+07   0.00      0.0
H21     +H                >H20     +H       * 5.42E+10   0.00      0.0
HD1     +H                >HD0     +H       * 5.42E+10   0.00      0.0
H21     +D                >H20     +D       * 5.42E+10   0.00      0.0
HD1     +D                >HD0     +D       * 5.42E+10   0.00      0.0
H21     +D                >HD0     +H       * 2.00E+10   0.00      0.0
D21     +H                >HD0     +D       * 9.55E+09   0.00      0.0
HD1     +H                >H20     +D       * 9.55E+09   0.00      0.0
HD1     +D                >D20     +H       * 9.55E+09   0.00      0.0
H21     +D                >HD1     +H       * 1.04E+12   0.00      0.0
D21     +H                >HD1     +D       * 1.27E+09   0.00      0.0
HD1     +H                >H21     +D       * 5.21E+11   0.00      0.0
HD1     +D                >D21     +H       * 6.00E+10   0.00      0.0
H21     +HD0              >H20     +HD0     * 1.13E+11   0.00      0.0
HD1     +H20              >HD0     +H21     * 8.43E+09   0.00      0.0
H21     +D20              >H20     +D21     * 1.19E+10   0.00      0.0
D21     +H20              >D20     +H21     * 6.02E+07   0.00      0.0
HD1     +D20              >HD0     +D21     * 2.11E+09   0.00      0.0
H21     +H20              >H20     +H20     * 7.80E+07   0.00      0.0
H21     +HE               >H20     +HE      * 1.56E+07   0.00      0.0
HD1     +HE               >HD0     +HE      * 3.01E+07   0.00      0.0
H       +H       +HE      >H20     +HE      * 4.10E+08   0.00      0.0
H       +H       +H20     >H20     +H20     * 4.68E+08   0.00      0.0
D       +D       +D20     >D20     +D20     * 3.55E+08   0.00      0.0
*****************************************
END
COLLISION EFFICIENCIES
END
COMPLEX REACTIONS
005  COMPLEX REACTIONS AT THE WALL
 1.00 H21     *1.0                                          8.700E-04
 1.00 H20                                                   0.0  0.00
 1.00 D21     *1.0                                          8.700E-04
 1.00 D20                                                   0.0  0.00
 1.00 HD1     *1.0                                          8.700E-04
 1.00 HD0                                                   0.0  0.00
 1.00 H       *1.0                                          1.000E-03
 0.50 H20                                                   0.0  0.00
```

```
 1.00 D       *1.0                                              1.000E-03
 0.50 D20                                                       0.0  0.00
END
```

- NO_2 and H_2 reactive mixture

```
*****************************
****                         *
****  1. NO2-H2 MECHANISM    *
****    k = cm3/mol/s        *
*****************************************
H       +H       +M       >H2      +M       * 2.50E+09   0.00      0.0    0
H       +H       +H2      >H2      +H2      * 2.90E+03   0.00      0.0    0
H       +H       +HE      >H2      +HE      * 2.50E+09   0.00      0.0    0
H       +NO2              >OH      +NO      * 7.20E+13   0.00      0.0    0
H2      +OH               >H2O     +H       * 4.52E+11   0.00      0.0    0
OH      +OH               >H2O     +O       * 1.00E+12   0.00      0.0    0
*****************************************
END
COLLISION EFFICIENCIES
END
COMPLEX REACTIONS
001  COMPLEX REACTIONS AT THE WALL
 1.00 H       *1.0                                              1.000E-03
 0.50 H2                                                        0.0  0.00
END
```

Appendix B

CVD-Experiment reaction model

```
MECHANISM C1-C2,Methan, P = 50  MBAR, HIGH TEMP.,OHNE C2H5O BZW. C2H5OH
*****************************************
****     01.   H2-O2 React. (no HO2, H2O2)    * reaction rates *
*****************************************
O2      +H       =OH      +O                  2.000E+14  0.0      70.300
H2      +O       =OH      +H                  5.060E+04  2.670    26.300
H2      +OH      =H2O     +H                  1.000E+08  1.600    13.800
OH      +OH      =H2O     +O                  1.500E+09  1.140     0.420
*****************************************
****     02.   Recombination Reactions
*****************************************
H       +H       +M'      =H2      +M'        1.800E+18 -1.000     0.000
O       +O       +M'      =O2      +M'        2.900E+17 -1.000     0.0
H       +OH      +M'      =H2O     +M'        2.200E+22 -2.000     0.000
*****************************************
****     03.   HO2 Formation/Consumption
*****************************************
H       +O2      +M'      =HO2     +M'        2.300E+18 -0.800     0.0
HO2     +H       =OH      +OH                 1.500E+14  0.0       4.200
HO2     +H       =H2      +O2                 2.500E+13  0.0       2.900
HO2     +H       =H2O     +O                  3.000E+13  0.0       7.200
HO2     +O       =OH      +O2                 1.800E+13  0.0      -1.7
HO2     +OH      =H2O     +O2                 6.000E+13  0.0       0.0
*****************************************
****     04.   H2O2 Formation/Consumption
*****************************************
HO2     +HO2     =H2O2    +O2                 2.500E+11  0.0      -5.200
OH      +OH      +M'      =H2O2    +M'        3.250E+22 -2.000     0.0
H2O2    +H       =H2      +HO2                1.700E+12  0.0      15.700
H2O2    +H       =H2O     +OH                 1.000E+13  0.0      15.000
```

```
H2O2    +O       =OH      +HO2                2.803E+13  0.0      26.800
H2O2    +OH      =H2O     +HO2                5.400E+12  0.0       4.200
*****************************************
****     05.   CO REACTIONS
*****************************************
CO      +OH      =CO2     +H                  6.000E+06  1.500    -3.100
CO      +HO2     =CO2     +OH                 1.500E+14  0.0      98.700
CO      +O       +M'      =CO2     +M'        7.100E+13  0.0     -19.000
CO      +O2      =CO2     +O                  2.500E+12  0.0     200.000
C       +O2      =CO      +O                  2.000E+13  0.0       0.0
C       +OH      =CO      +H                  5.000E+13  0.0       0.0
*****************************************
****     10.   CH Reactions
*****************************************
CH      +O       =CO      +H                  4.000E+13  0.0       0.0
CH      +O2      =CHO     +O                  6.000E+13  0.0       0.0
CH      +CO2     =CHO     +CO                 3.400E+12  0.0       2.900
CH      +H2O     =3CH2    +OH                 5.700E+12  0.0      -3.200
CH      +OH      =C       +H2O                4.000E+07  2.0      12.300
CH      +H       =C       +H2                 1.500E+14  0.0       0.0
C       +H       +M'      =CH      +M'        3.000E+14  0.0      -1.0
*****************************************
****     11.   CHO REACTIONS
*****************************************
CHO     +M'      =CO      +H       +M'        7.100E+14  0.0      70.300
CHO     +H       =CO      +H2                 9.000E+13  0.0       0.0
CHO     +O       =CO      +OH                 3.000E+13  0.0       0.0
CHO     +O       =CO2     +H                  3.000E+13  0.0       0.0
CHO     +OH      =CO      +H2O                1.000E+14  0.0       0.0
CHO     +O2      =CO      +HO2                3.000E+12  0.0       0.0
CHO     +CHO     =CH2O    +CO                 3.000E+13  0.0       0.0
CH      +OH      =CHO     +H                  3.000E+13  0.0       0.0
*****************************************
****     12.   CH2 Reactions
*****************************************
3CH2    +H       =CH      +H2                 6.000E+12  0.0      -7.500
3CH2    +O       >CO      +H       +H         8.400E+12  0.0       0.0
3CH2    +O2      =CO      +OH      +H         1.300E+13  0.0       6.200
3CH2    +O2      =CO2     +H2                 1.200E+13  0.0       6.200
1CH2    +M'      =3CH2    +M'                 1.200E+13  0.0       0.0
1CH2    +O2      =CO      +OH      +H         3.100E+13  0.0       0.0
1CH2    +H2      =CH3     +H                  7.200E+13  0.0       0.0
3CH2    +3CH2    =C2H2    +H2                 1.200E+13  0.0       3.4
3CH2    +3CH2    =C2H2    +H       +H         1.100E+14  0.0       3.4
3CH2    +CH3     =C2H4    +H                  4.200E+13  0.0       0.0
*****************************************
```

```
****     13.   CH2O Reactions
*****************************************
CH2O    +M'      =CHO     +H       +M'       5.000E+16  0.0     320.000
CH2O    +H       =CHO     +H2               2.300E+10  1.05     13.700
CH2O    +O       =CHO     +OH               4.150E+11  0.57     11.600
CH2O    +OH      =CHO     +H2O              3.400E+09  1.2      -1.900
CH2O    +HO2     =CHO     +H2O2             3.000E+12  0.0      54.7
CH2O    +CH3     =CHO     +CH4              1.000E+11  0.0      25.500
CH2O    +O2      =CHO     +HO2              6.000E+13  0.0     170.700
3CH2    +OH      =CH2O    +H                2.500E+13  0.0       0.0
CH      +H2O     =CH2O    +H                1.170E+15 -0.75      0.0
*****************************************
****     14.   CH3 Reactions
*****************************************
CH3     +M'      =3CH2    +H       +M'       1.000E+16  0.0     379.000
CH3     +O       =CH2O    +H                8.430E+13  0.0       0.0
CH3     +H       =CH4                       1.060E+36 -7.30     36.25
CH3     +OH      >CH3O    +H                2.260E+14  0.0      64.8
CH3O    +H       >CH3     +OH               4.750E+16 -0.13     88.0
CH3     +O2      >CH2O    +OH               3.300E+11  0.0      37.400
CH3     +HO2     =CH3O    +OH               1.800E+13  0.0       0.0
CH3     +HO2     =CH4     +O2               3.600E+12  0.0       0.0
CH3     +CH3     >C2H4    +H2               1.000E+16  0.0     134.000
CH3     +CH3     =C2H6                      1.300E+58-13.8      79.30
*****************************************
****     15a.   CH3O Reactions
*****************************************
CH3O    +M'      =CH2O    +H       +M'       5.000E+13  0.0     105.0
CH3O    +H       =CH2O    +H2               1.800E+13  0.0       0.0
CH3O    +O2      =CH2O    +HO2              4.000E+10  0.0       8.9
CH2O    +CH3O    >CH3OH   +CHO              0.600E+12  0.0      13.8
CH3OH   +CHO     >CH2O    +CH3O             0.650E+10  0.0      57.2
CH3O    +O       =O2      +CH3              1.100E+13  0.0       0.0
CH3O    +O       =OH      +CH2O             1.400E+12  0.0       0.0
*****************************************
****     15b.   CH2OH Reactions
*****************************************
CH2OH   +M'      =CH2O    +H       +M'       5.000E+13  0.0     105.0
CH2OH   +H       =CH2O    +H2               3.000E+13  0.0       0.0
CH2OH   +O2      =CH2O    +HO2              1.000E+13  0.0      30.0
*****************************************
****     16.   CH3O2 Reactions
*****************************************
CH3O2   +M'      >CH3     +O2      +M'       0.724E+17  0.0     111.1
CH3     +O2      +M'      >CH3O2   +M'       0.141E+17  0.0      -4.6
CH3O2   +CH2O    >CH3O2H  +CHO              0.130E+12  0.0      37.7
```

```
CH3O2H  +CHO     >CH3O2   +CH2O              0.250E+11  0.0      42.3
CH3O2   +CH3     >CH3O    +CH3O              0.380E+13  0.0      -5.0
CH3O    +CH3O    >CH3O2   +CH3               0.200E+11  0.0       0.0
CH3O2   +HO2     >CH3O2H  +O2                0.460E+11  0.0     -10.9
CH3O2H  +O2      >CH3O2   +HO2               0.300E+13  0.0     163.3
CH3O2   +CH3O2   >CH2O    +CH3OH   +O2       0.180E+13  0.0       0.0
CH2O    +CH3OH   +O2      >CH3O2   +CH3O2     0.000E+00  0.0       0.0
CH3O2   +CH3O2   >CH3O    +CH3O    +O2       0.370E+13  0.0       9.2
CH3O    +CH3O    +O2      >CH3O2   +CH3O2     0.000E+00  0.0       0.0
*****************************************
****     17.   CH4 Reactions
*****************************************
CH4     +H       =H2      +CH3               1.300E+04  3.000    33.600
CH4     +O       =OH      +CH3               6.923E+08  1.560    35.500
CH4     +OH      =H2O     +CH3               1.600E+07  1.830    11.600
CH4     +HO2     =H2O2    +CH3               1.100E+13  0.0     103.100
CH4     +3CH2    =CH3     +CH3               1.300E+13  0.0      39.900
*****************************************
****     18.   CH3OH Reactions
*****************************************
CH3OH            =CH3     +OH                1.130E+25 -3.40    372.9
CH3OH   +H       =CH2OH   +H2                4.000E+13  0.0      25.5
CH3OH   +O       =CH2OH   +OH                1.000E+13  0.0      19.6
CH3OH   +OH      =CH2OH   +H2O               1.000E+13  0.0       7.1
CH3OH   +HO2     >CH2OH   +H2O2              0.620E+13  0.0      81.1
CH2OH   +H2O2    >HO2     +CH3OH             0.100E+08  1.7      47.9
CH3OH   +CH3     =CH4     +CH2OH             9.000E+12  0.0      41.1
CH3O    +CH3OH   >CH2OH   +CH3OH             0.200E+12  0.0      29.3
CH2OH   +CH3OH   >CH3O    +CH3OH             0.220E+05  1.7      45.4
CH3OH   +CH2O    >CH3O    +CH3O              0.153E+13  0.0     333.2
CH3O    +CH3O    >CH3OH   +CH2O              0.300E+14  0.0       0.0
*****************************************
****     19.   CH3O2H Reactions
*****************************************
CH3O2H           =CH3O    +OH                4.000E+15  0.0     180.5
OH      +CH3O2H  =H2O     +CH3O2             2.600E+12  0.0       0.0
*****************************************
*******************************
****                         *
****  4. C2 MECHANISM        *
****                         *
*******************************
*****************************************
****     19B.  C2 Reactions
*****************************************
C2      +O2      =CO      +CO                5.000E+13  0.0       0.0
```

```
C       +C       +M'      =C2      +M'        3.000E+14  0.0      -1.0
CH      +CH      =C2      +H       +H         5.000E+13  0.0      19.0
CH      +CH      =C2      +H2                 5.000E+12  0.0       0.0
C       +CH      =C2      +H                  5.000E+13  0.0       0.0
*****************************************
****    20.    C2H REACTIONS
*****************************************
C2H     +O       =CO      +CH                 1.000E+13  0.0       0.0
C2H     +O2      =HCCO    +O                  3.000E+12  0.0       0.0
C       +3CH2    =C2H     +H                  5.000E+13  0.0       0.0
C2H     +O2      =CO      +CO      +H         3.520E+13  0.0       0.0
C2H     +OH      =HCCO    +H                  2.000E+13  0.0       0.0
C2H     +OH      =C2      +H2O                4.000E+07  2.0      32.8
C2      +H2      =C2H     +H                  4.000E+05  2.4       4.1
*****************************************
****    20A.    C2O REACTIONS
*****************************************
C2O     +H       =CH      +CO                 1.000E+13  0.0       0.0
C2O     +O       =CO      +CO                 5.000E+13  0.0       0.0
C2O     +OH      =CO      +CO      +H         2.000E+13  0.0       0.0
C2O     +O2      =CO      +CO      +O         2.000E+13  0.0       0.0
C2      +OH      =C2O     +H                  5.000E+13  0.0       0.0
*****************************************
****    20B.    HCCO REACTIONS
*****************************************
HCCO    +H       =3CH2    +CO                 1.500E+14  0.0       0.0
HCCO    +O       >CO      +CO      +H         9.600E+13  0.0       0.0
HCCO    +3CH2    =C2H3    +CO                 3.000E+13  0.0       0.0
*****************************************
****    21.    C2H2 REACTIONS
*****************************************
C2H2    +M'      =C2H     +H       +M'        3.600E+16  0.0     446.0
C2H2    +O2      =HCCO    +OH                 2.000E+08  1.5     126.0
C2H2    +H       =C2H     +H2                 1.500E+14  0.0      79.6
C2H2    +O       =3CH2    +CO                 1.720E+04  2.8       2.1
C2H2    +O       =HCCO    +H                  1.720E+04  2.8       2.1
C2H2    +OH      =H2O     +C2H                6.000E+13  0.0      54.2
CH      +3CH2    =C2H2    +H                  4.000E+13  0.0       0.0
C       +CH3     =C2H2    +H                  5.000E+13  0.0       0.0
C2H2    +O       =C2H     +OH                 3.160E+15 -0.6      61.5
CH      +HCCO    =C2H2    +CO                 5.000E+13  0.0       0.0
*****************************************
****    21A.    CH2CO REACTIONS
*****************************************
CH2CO   +M'      =3CH2    +CO      +M'        1.000E+16  0.0     248.0
CH2CO   +H       =CH3     +CO                 3.600E+13  0.0      14.1
```

```
CH2CO   +O       =CHO     +CHO              2.300E+12  0.0       5.7
CH2CO   +OH      =CH2O    +CHO              1.000E+13  0.0       0.0
CH      +CH2O    =CH2CO   +H                9.460E+13  0.0      -2.11
*****************************************
****    25.    C2H3 REACTIONS
*****************************************
C2H3            =C2H2    +H                1.900E+38 -8.5     192.6
C2H3    +OH     =C2H2    +H2O              5.000E+13  0.0       0.0
C2H3    +H      =C2H2    +H2               1.200E+13  0.0       0.0
C2H3    +O      =C2H2    +OH               1.000E+13  0.0       0.0
C2H3    +O      =CH3     +CO               1.000E+13  0.0       0.0
C2H3    +O      =CHO     +3CH2             1.000E+13  0.0       0.0
C2H3    +O2     =C2H2    +HO2              5.400E+12  0.0       0.0
CH      +CH3    =C2H3    +H                3.000E+13  0.0       0.0
C2H3    +CH     =3CH2    +C2H2             5.000E+13  0.0       0.0
*****************************************
****    22A.   CH3CO REACTIONS
*****************************************
CH3CO           =CH3     +CO               7.700E+23 -4.7      68.58
CH3CO   +H      =CH2CO   +H2               2.000E+13  0.0       0.0
*****************************************
****    22B.   CH2CHO REACTIONS
*****************************************
CH2CHO  +H      =CH2CO   +H2               2.000E+13  0.0       0.0
*****************************************
****    23.    C2H4 REACTIONS
*****************************************
C2H4    +M'     =C2H2    +H2      +M'      2.500E+17  0.0     319.8
C2H4    +M'     =C2H3    +H       +M'      1.700E+18  0.0     404.0
C2H4    +H      =C2H3    +H2               1.700E+15  0.0      62.9
C2H4    +O      =CH2CHO  +H                5.200E+05  2.08      0.0
C2H4    +O      =CHO     +CH3              1.210E+06  2.08      0.0
C2H4    +OH     =C2H3    +H2O              6.500E+13  0.0      24.9
CH4     +CH     =C2H4    +H                3.000E+13  0.0      -1.7
*****************************************
****    23A.    CH3CHO REACTIONS
*****************************************
CH3CHO  +M'     =CH3     +CHO     +M'      7.000E+15  0.0     342.8
CH3CHO  +H      =CH3CO   +H2               2.100E+09  1.16     10.1
CH3CHO  +H      =CH2CHO  +H2               2.000E+09  1.16     10.1
CH3CHO  +O      =CH3CO   +OH               5.000E+12  0.0       7.6
CH3CHO  +O      =CH2CHO  +OH               8.000E+11  0.0       7.6
CH3CHO  +O2     =CH3CO   +HO2              4.000E+13  0.0     164.3
CH3CHO  +OH     =CH3CO   +H2O              2.300E+10  0.73     -4.7
CH3CHO  +HO2    =CH3CO   +H2O2             3.000E+12  0.0      50.0
CH3CHO  +3CH2   =CH3CO   +CH3              2.500E+12  0.0      15.9
```

```
CH3CHO  +CH3     =CH3CO   +CH4                2.000E-06  5.64     10.3
****************************************
****    24.   C2H5 REACTIONS
****************************************
C2H5            =C2H4    +H                  7.370E+42 -9.5     211.94
C2H5    +H      =CH3     +CH3                3.000E+13  0.0       0.0
C2H5    +O      =CH3CHO  +H                  5.000E+13  0.0       0.0
C2H5    +O      =CH2O    +CH3                1.000E+13  0.0       0.0
C2H5    +O2     =C2H4    +HO2                1.100E+10  0.0      -6.3
C2H5    +CH3    =C2H4    +CH4                1.140E+12  0.0       0.0
C2H5    +C2H5   =C2H4    +C2H6               1.400E+12  0.0       0.0
****************************************
****    25.   C2H6 REACTIONS
****************************************
C2H6    +H      =C2H5    +H2                 1.400E+09  1.5      31.1
C2H6    +O      =C2H5    +OH                 1.000E+09  1.5      24.4
C2H6    +OH     =C2H5    +H2O                7.200E+06  2.0       3.6
C2H6    +HO2    =C2H5    +H2O2               1.700E+13  0.0      85.9
C2H6    +O2     =C2H5    +HO2                6.000E+13  0.0     217.0
C2H6    +3CH2   =C2H5    +CH3                2.200E+13  0.0      36.3
C2H6    +CH3    =C2H5    +CH4                1.500E-07  6.0      25.4
****************************************
****    26.   C3 Reactions
****************************************
H       +C3     +M'      =C3H     +M'        7.000E+16 -1.000     0.00
H2      +C3     =C3H     +H                  4.000E+05  2.400     0.00
C       +C2     +M'      =C3      +M'        4.000E+16 -1.000     0.00
C       +C2H    =C3      +H                  4.000E+16 -1.000     0.00
CH      +C2     =C3      +H                  1.000E+14  0.000     0.00
****
****************************************
END
COLLISION EFFICIENCIES
M'      =H2     +H2O     +O2      +CO2     +CO      +CH4     +AR
         1.0      6.5      0.4      1.50     0.75     3.0      3.0
END
COMPLEX REACTIONS
000  COMPLEX REACTIONS
END
*****
END
```

Appendix C

A C++ Package for the Calculation of Flow Reactors with Detailed Chemistry – User Guide –

C.1 Overall Structure

This c++ package allows to calculate multicomponent gas flows taking into account convection, diffusion and chemical reactions in the gas phase as well as reactions at walls. It computes the velocity field, pressure, density and temperature distribution as well as the gas chemical composition by solving a system of PDEs describing the evolution in space and time of these variables.

The system is made of the Navier-Stokes equations supplemented with species mass conservation equations. The spatial discretization is based on a finite element approximation. The time discretization is restricted to an implicit Euler scheme. This code has been used to calculate quasi-stationary solutions and therefore accurate time-step approximations were not needed.

A defect correction scheme is used to solve the non-linear systems for each time-step. The resulting linear systems are solved with a GMRES method preconditioned by a multigrid method. The global system is split in two parts with respect to the defect-correction matrix used; the first part corresponds to the Navier-Stokes equations, which describe the average flow of the mixture, and the second part describes the chemistry.

This code is based on the **DEAL** c++ library which provides a flexible development environment for adaptive finite element methods. Be sure to have this library installed on your computer in order to be able to use the present

package. The reader can find more informations about the **DEAL** library at `http://gaia.iwr.uni-heidelberg.de`.

Our calculation code as well as the **DEAL** library have been written and tested on SUN Solaris workstations with GNU gcc 2.8. On other systems, some changes might be necessary to achieve the compilation and linking.

C.2 Getting Things Installed and Started

The package is available as a compressed tar file: `flow_reactor.tar.gz`. To uncompress and unpack the tar file use the commands:
`gzip -d flow_reactor.tar.gz`
`tar xf flow_reactor.tar`

There will be one directory created called `flow_reactor`. In this directory, a set of subdirectories are to be found:

- `Global_chemical_data` contains global chemical data about a lot of chemical species. It should not be changed.
- `SOURCE` contains the source files of the program `reactor`.
- `INSINP` contains a FORTRAN program which uses the files contained in the `Global_chemical_data` directory as well as some other parameter files (see below) in order to create a specification file defining the species that are to be found in the flow, with their chemical characteristics, as well as the reactions which are to occur in the mixture. This created file is read by the program `flow_reactor` at the start to define and initialize the chemistry for the computation.
- `USER_DATA` contains parameter files which describe the chemical species found in the mixture, the chemical reactions and the boundary conditions.
- `OUTPUT` contains the results of the computations, i.e. files in UCD (.inp) and GNUPLOT (.dat) formats.

To compile the code, go in `SOURCE`, edit the `Makefile` file and write there the absolute path of the `USER_DATA` and `SOURCE` directories in the `USER` and `SOURCE` variable declarations:
`USER = /absolute_path/USER_DATA`
`SOURCE = /absolute_path/SOURCE` Do the same for the **DEAL** library path:
`DEAL = /absolute_path/deal` Save the file and compile the code with gnu-make by typing `make`.

First a FORTRAN program called `insinp.x` from the `INSINP` directory also has to be used in order to create a file specifying all parameters and variables needed in the chemical processes as well as the boundary conditions. This executable is supplied within the package, but under certain circumstances it might be necessary to compile it again. If it is the case go in the `INSINP` directory and type `make -f Make_Inp`. This program reads chemical data and creates a new file containing the only data needed for the current calculation. Here you may also have to edit the file `Make_Inp` and write the right path declarations.

A script-file named `go`, which has to be executed in the main directory `flow_reactor`, calls the two latter programs (`insinp.x` and `reactor`), in the right order, to start the computation according to the flow chemical characteristics defined by the user. Thus to start the solution process go in the main directory and type `go`.

C.3 Input and Output Data

The files `input`, `mechanism`, `simulation.data` and `const_data`, in directory `USER_DATA`, contain all the parameters the program needs to know. A change in the file `const_data` demands that the program is compiled again (see Section C.2).

C.3.1 Chemical Mechanism

The chemical mechanism is described in the file named `mechanism`. We give an example of mechanism file. The first part describes the simple reactions which take place within the gas phase. The reaction rate is given after the definition of the corresponding reaction on the same line. Further the reactions at solid boundaries are defined with their reaction probability. Don't forget to set the number of reactions at the wall (named complex reactions).

```
MECHANISM OF D2(V=0,1) REACT.
****
*****************************
****                         *
****  1. D2-HE MECHANISM     *
****                         *
*******************************************
D21     +D       +        >D20     +D       * 2.36E+11   0.00   0.0   nist
D21     +D20     +        >D20     +D20     * 6.50E+07   0.00   0.0   n.v.
D21     +HE      +        >D20     +HE      * 1.56E+07   0.00   0.0   n.v.
```

```
D       +D       +HE      >D20     +HE      * 5.00E+16   0.00   0.0   nist
D       +D       +D20     >D20     +D20     * 2.90E+15   0.00   0.0   n.v.
*****************************************
END
COLLISION EFFICIENCIES
END
COMPLEX REACTIONS
002  COMPLEX REACTIONS AT THE WALL
 1.00 D21     *1.0                                          1.500E-03
 1.00 D20                                                   0.0  0.00
 1.00 D       *1.0                                          1.000E-04
 0.50 D20                                                   0.0  0.00
END
```

C.3.2 Inflow Data

The inflow data are given in the file `input`. In this file one can set the mole fractions of each species, the temperature, and velocity of the mixture at the inflow boundary. This boundary contains two different area, the inner and outer tubes. The file structure is the following:

```
OPTIONS.................(FORMAT  7(A4,6X), END WITH -END -)
REGRID   /PCON     /PROFIL   /TSO      /         /         /         /
STORE    /EXTRA   2/OUTPUT  1/ENERG   2/         /         /         /
END      /         /         /         /         /         /         /
SPECIES.......................(Format  7(2A4,1X,A1), end with -END -)
HE      ,H20     ,H21     ,H       ,HD0     ,HD1     ,D20     ,
D21     ,D       ,        ,        ,        ,        ,        ,
END
*************************************************************************
INFLOW COMP.    INNEN    AUSSEN ...(FORMAT A10,2F10.3, END WITH -END -)
HE      :      0.792     0.000      (SAME ORDER AS ABOVE !!!!!)
H20     :      0.000     0.992      (MOLE-FRACTION)
H21     :      0.000     0.005      ****
H       :      0.000     0.003      ****
HD0     :      0.000     0.000      ****
HD1     :      0.000     0.000      ****
D20     :      0.115     0.000      ****
D21     :      0.002     0.000      ****
D       :      0.091     0.000      ****
P       :    5.33E-3   5.33E-3      BAR
T       :       292.      292.      K
U       :      0.000     0.000      M/S
```

```
V       :       64.00      34.00       M/S
```

There are some more lines in this file but they are outdated and not taken into account. It is important to write the name of the species in the list on the top of the file in the right format (8 characters between 2 commas). After the species list, the specification of the inflow data is to be found in two columns for the inner (INNEN) and the outer (AUSSEN) tube; first the species mole fraction, then the pressure, the temperature and finally the radial and axial velocities. It is to be noted that the species MOLE fractions are to be given in this file, although the outputs of the program give mass fractions.

In directory `GLOBAL_CHEM_DATA`, the files `mol.dat` and `thermo.dat` contain species specific databases and should not be changed or even edited.

The script `go` in the main directory calls the preprocessor `insinp.x`, which itself reads the input files and species data bases to create a data set called fort.3 also written in the main directory. This data set is read by the actual simulation code to define the flow chemical characteristics.

C.3.3 Simulation Process

The file `const_data.h` in directory `USER_DATA` contains data concerning the solvers, the adaptive process and the outputs. This file is made of several well defined parts:

- Time step - Solver tolerance:

  ```
  #define TIME_STEP_SIZE 2.
  #define TIME_STEP_NUMBER 50
  #define MAX_SIMPLE_IT  30
  #define SOLVER_TOL  1.E-7
  ```

  The time step size is normed by the density of the mixture and therefore is actually around a factor 10^{-4} smaller as the time step given by `TIME_STEP_SIZE`.
  The total number of time steps is given by `TIME_STEP_NUMBER`, and the number of time steps without refinement of the mesh is set by `MAX_SIMPLE_IT`. A quasi-stationary state can in this way be reached before the local refinement process begins. After `MAX_SIMPLE_IT` number of iterations the adaptive refinement process begins.
  `SOLVER_TOL` is the tolerance of the defect-correction process on the residual.

- Number of species:

```
#define SPECIES_COMP 10
```

It should be set to the number of species + 1 for the temperature.

- Neutral species:

```
#define NEUTRAL_SPECIE 1
```

It is used to define the species which is found in the tube at the start of the calculation. It should be a neutral species which does not react (or only weakly) with other species of the mixture. This allow to avoid too stiff source terms at the beginning of the computation.

- Wall-reaction flag:

```
static int WALL_CHEMISTRY = 1;
```

if `WALL_CHEMISTRY` is equal to 1, the wall reactions are taken into account. If it is equal to 0 they are not.

C.3.4 Refinement process

The refinement process is based on the accurate calculation of some average or point values of mass fractions for selected species. The following variables allow the user to indicate which values for which species has to be known with accuracy.

- Observation flag (solve-dual-problem flag):

```
#define OBSERVATION 1
```

This flag is set to 1 if some physical values have to be known with accuracy. In this case the dual problem is solved for each refinement steps and the dual solution is used to calculate the corresponding error estimator that is used to refine the mesh.
If this flag is set to 0, the dual problem is not solved and the error estimator does not contain any weights.

The following variables make sense only if the latter flag is set to 1, i.e. average or point values of some species mass fractions are to be known with accuracy.

- Observed species:

```
#define OBSERVATION_SIZE  2
static int OBSERVATION_SPECIES[OBSERVATION_SIZE] = {1,2};
```

The first variable defines the number of species for which the mass fraction has to be known with accuracy. This number must be between 1 and SPECIES_COMP-1. The second variable is an array and contains the numbers of the corresponding species. The species are ordered in the same way as in the file input.

- Observation direction:

```
#define OBSERVATION_XLINE 1
#define OBSERVATION_YLINE 0
#define OBSERVATION_AXE_POINTS 0
```

X corresponds to the radial direction and Y corresponds to the axial direction. Here we define which value has to be known with accuracy. For each of these 3 variables the value one means that this value is to be calculated with precision.
OBSERVATION_XLINE corresponds to average values of the mass fraction of the species defined above along radial lines which are defined later.
OBSERVATION_YLINE corresponds to average values of the mass fraction of the species defined above along axial lines which are defined later.
OBSERVATION_AXE_POINTS corresponds to the point values of the mass fraction of the species defined above along the axis of the tube. The positions of these points along the axis are defined later.
There must be one and only one of these three variables with the value set to 1. The two others must have the value 0.

- Position of the observation lines/points (in meter):

```
#define OBSERVATION_NUMBER 4
static double OBSERVATION_RADIUS[OBSERVATION_NUMBER] = {0.};
static double OBSERVATION_HEIGHTS[OBSERVATION_NUMBER] = {1,2,3,4};
```

The variable OBSERVATION_NUMBER defines the number of lines or points where average or point values of the mass fractions have to be known with precision.
The variable OBSERVATION_RADIUS is relevant only if OBSERVATION_YLINE is set to 1, since it defines the radius for each line (parallel to the tube axis) where the averaged mass fraction has to be calculated with precision.
The variable OBSERVATION_HEIGHTS is relevant only if OBSERVATION_YLINE is set to 1, since it defines the position on the tube axis for each radial line or point of the axis where the mass fraction has to be calculated with precision.

- Number of maximal refinement level

  ```
  #define MAX_REFINEMENT_LEVEL 20
  ```

  This variable defines the maximal number of refinement level for the adaptive mesh refinement process. It is set to as default to 20 and can be left to this value.

C.3.5 Output Data

The output that can be customized here are done in Gnuplot format and corresponds to the variable evolutions along radial lines. The output files are stored in the directory `OUTPUT` which is in the main directory.

- Number of output lines:

  ```
  #define OUTPUT_NUMBER 3
  ```

  With this variable, one defines the number of lines for which there must be an output file. In this file the evolution of the flow and chemical variables are written in Gnuplot format.

- Axial position of the output lines

  ```
  static double OUTPUT_HEIGHTS[OUTPUT_NUMBER] = {1,2,3};
  ```

  This array contains the axial position of the output lines expressed in meter from the tube start.

These files in Gnuplot-format have the following structure:

```
#file : OUTPUT/output_15_0.dat
#line output for y = 0.238 of variables:
#radial position, u, v, p*, T, HE, H20, H21, H, HD0, HD1,
D20, D21, D, rho, P/rho
0.0045 -0.115513 26.7997 0.549813 292 1 1.01773e-13 1e-13
1.35934e-13 1e-13 1e-13 1e-13 1e-13 1e-13 0.000878807 606504
```

The first line is the name of the file. The second line contains a description of the section for which we get the variable evolution. The third line is a description of the order in which the variables are stored in the file. And the following lines contain the data. The units for these data are SI (m/s, Pa, K, $cdots$) and mass fraction is stored for the species.

Additionally to these Gnuplot output file, output files containing the complete solution on the whole domain are created at the end of each time step or refinement step. They are also stored in the directory `OUTPUT`. These files are in UCD format, which can be read by AVS, dealvision or DeViSoR, which all three are visualization programs.

C.3.6 Mesh data

The name of the mesh file is given in the file `simulation.data` with absolute or relative path from the main directory where the script `go` is called. The domain dimensions are also to be found in this file.

```
#***************** Data about the computational field ******************
#***********************************************************************

#**** Mesh file name
#   ******************
cars_split.inp

#**** Domain dimensions (in meter) : ***
#       tube height     |       tube radius     |       splitter radius
        0.15                    0.016                   0.006
```

The tube height is the length of the tube. The tube radius is the radius of the outer tube. And the splitter radius is the radius of the intern tube. The values are needed by the program to calculate the inflow values.

C.4 Automatic mesh generation for CVD

In order to simplify the geometrical optimization process for CVD experiment we developed a mesh generator for the CVD geometry. This is only one file: `CVD_mesh_generator.cc` which can be simply compiled and linked by any c++ compiler.

The parameters which need to be set in the file are the following:

```
  name = "mesh.inp";

 /**
   *                        |-- substrat
   *                        V
```

```
 * |-----------------------------------------------------------------
 * |  <------------------------------ symmetry line                  |
 * |                                                                 |
 * | d_substrat                                                      |
 * |                                                                 |
 * |  CH4 inflow                                                     |
 * |----|                                          free outflow  --> |
 *      |                                                           |
 *      |     <- pipe                                               |
 *      |                                                           |
 * |----|                                                          |
 * |                                                               |
 * |                                                               |
 * |  d_pipe                                                       |
 * |                           H2/O2  inflow                       |
 * |                                                               |
 * -----------------------------------------------------------------
 *
 *
 */



/**
 * Enter here the numbers of columns and lines of the mesh to be generate
 *   ^ Lines
 *   |
 *   |
 *   |
 *   ----------> Columns
 */

//                                  # = number of

int nb_under_pipe_cols = 2;       // # columns under the pipe
int nb_above_pipe_cols = 2;       // # columns over the pipe
int nb_under_substrat_cols = 15;  // # columns on the right of the pipe
int nb_under_lines = 3;           // # lines under the pipe
int nb_pipe_lines = 3;            // # lines beside the pipe
int nb_between_lines = 4;         // # lines over the pipe

/**
  * enter here the widths of the domain (in m)
```

```
    */
    double substrat_width = 0.025;
    double pipe_width = 0.001;

   /**
    * enter here the heights of and distances in the domain (in m)
    */
    double substrat_height = 0.005;
    double pipe_height = 0.003;

    // distance between the inflow of the flame exhaust gas
    double d_pipe = 0.003;

    // distance between the pipe and the substrat
    double d_substrat = 0.005;


    /**
    * enter the numbers of the different boundary lines
    */
    int symmetry = 2;
    int outflow = 0;
    int wall = 3;
    int substrat_wall = 7;
    int CH4_inflow = 4;
    int H2O2_inflow = 5;

/*****************************************************************/
```

Bibliography

[1] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation, 1984.

[2] Mejdi Azaiez, Christine Bernardi, Monique Dauge, and Yvon Maday. Spectral methods for axisymmetric domains. *Institut de Recherche Mathématiques de Rennes, Prépublication*, 96–37, 1996.

[3] M. Baum, T. Poinsot, and D. Thévenin. Accurate boundary conditions for multicomponent reactive flows. *Jour. Comp. Phy.*, 116:247–261, 1994.

[4] R. Becker. *An Adaptive Finite Element Method for Incompressible Navier-Stokes Equations on Time-dependent Domains*. Phd thesis, University of Heidelberg, 1995.

[5] R. Becker, M. Braack, R. Rannacher, and C. Waguet. Fast and reliable solution of the Navier-Stokes equations including chemistry. *Preprint IWR (SFB 359), University of Heidelberg*, (99–03), 1999.

[6] R. Becker, G. Kanschat, and F.-T. Suttmeier. Differential equation analysis library. http://gaia.iwr.uni-heidelberg.de/DEAL/index.html, 1997.

[7] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite elements for optimal control of partial differential equations: Basic concept. *Preprint IWR, University of Heidelberg*, (98–55), 1998. To appear in SIAM J. Control and Optimization.

[8] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[9] C. Bernardi, M. Dauge, and Y. Maday. Numerical analysis and spectral methods in axisymmetric domains. part i: Functional prerequisite. *Publications du Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie*, 1994.

[10] C. Bernardi, F. Laval, B. Métivet, and B. Pernaud-Thomas. Finite element approximation of viscous flows with varying density. *SIAM J. Numer. Anal.*, 29(5):1203–1243, 1992.

[11] C. Bernardi and Y. Maday. Properties of some weighted sobolev spaces ans applications to spectral approximations. *SIAM J. Numer. Anal.*, 26(4):769–829, 1989.

[12] R. B. Bird, W. E. Stewart, and E. N. LightFoot. *Transport Phenomena.* John Wiley and Sons, Inc., New York, 1960.

[13] M. Braack. *An Adaptive Finite element Method for Reactive Flow Problems.* PhD thesis, University of Heidelberg, 1998.

[14] D. Braess. *Finite Elemente.* Springer Verlag, 1991.

[15] S. C. Brenner and R. L. Scott. *The Mathematical Theory of Finite Element Methods.* Springer, Berlin-Heidelberg-New York, 1994.

[16] T. P. Coffee and J. M. Heimerl. Transport algorithms for premixed laminar steady-state flames. *Comb. Flame*, 43:273, 1981.

[17] M. E. Coltrin, R. J. Kee, and F.M. Rupley. Surface chemkin (version 3.6): A fortran package for analyzing heterogeneous chemical kinetics at a solid-surface - gas-phase interface. Technical report, Sandia National Laboratories, 1990.

[18] M. E. Coltrin, H. K. Moffat, R. J. Kee, and F. M. Rupley. *CRESLAF: a FORTRAN Program for modeling laminar, chemically reacting, boundary-layer flow in cylindrical or planar channels.* Sandia National Laboratories, 1991.

[19] Y. D'Angelo and B. Larrouturou. Comparison and analysis of some numerical schemes for stiff complex chemistry problems. *Mathematical Modelling and Numerical Analysis*, 29(3):259–301, 1995.

[20] G. Dixon-Lewis. Flame structure and flame reaction kinetics, ii. transport phenomena in multicomponent systems. In *Proc. Roy. Soc.*, number A 307, pages 111–135, 1968.

[21] J. Dongarra, A. Lumsdaine, R. Poso, and K. A. Remington. *IML++ v. 1.2 - Iterative Methods Library - Reference Guide*, April 1996.

[22] P. Houston E. Süli, R. Rannacher. Modified dual problem for stabilized FEM. to appear.

[23] P. Farmanara. Spektroskopische Untersuchungen zur Verbrennungsunterstützten Abscheidung von Diamantschichten. Master's thesis, University of Heidelberg, 1996.

[24] C. A. J. Fletcher. *Computational Techniques for fluid dynamics.* Springer, 1997.

[25] V. Girault and P.-A. Raviart. *Finite Elements for Navier-Stokes Equations*. Springer-Verlag, Berlin, 1986.

[26] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1: Fundamentals of Numerical Discretization. John Wiley and Sons, Inc., New York, 1988.

[27] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird. *Molecular Theory of Gases and Liquids*. John Wiley and Sons, Inc., New York, 1954.

[28] T. J. R. Hughes. Recent progress in the development and understanding of supg methods with special reference to the compressible Euler and Navier-Stokes equations. *Inter. Jour. Num. Meth. Fluid Dyn.*, 7:1261–1275, 1987.

[29] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumvent the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation for the Stokes problem accomodating equal order interpolation. *Comp. Meth. Appl. Mech. Eng.*, 59:89–99, 1986.

[30] C. Johnson. *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge-Lund, 1987.

[31] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Math. Comput.*, 47(175):1–18, 1986.

[32] W. Juchmann. Laserspektroskopische Untersuchungen der CVD-Synthese von Diamant. Master's thesis, Fakultät für Physik und Astronomie, Universität Heidelberg, 1995.

[33] G. Kanschat. *Parallel and Adaptive Galerkin Methods for Radiative Transfer Problems*. Phd thesis, University of Heidelberg, 1996.

[34] H. Kapp. *Adaptive Finite Element Method for Optimization in Partial Differential Equations*. Phd thesis, to appear, University of Heidelberg, 2000.

[35] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller. A fortran computer code package for the evaluation of gas-phase, multicomponent transport properties. Technical report, Sandia National Laboratories, 1986.

[36] R. J. Kee, F. M. Rupley, and J. A. Miller. The chemkin thermodynamic data base. Report SAND87-8215.UC4, Sandia National Laboratories, 1987.

[37] P. Leyland, F. Benkhaldoun, N. Maman, and B. Larrouturou. Dynamical mesh adaption criteria for accurate capturing of stiff phenomena in combustion. Technical Report 1876, INRIA, 1993.

[38] G. Lube and A. Auge. Stabilized Galerkin finite element methods for simulating thermally coupled flows. In *Finite Element in Fluids*, volume 1, pages 232–241. CIMNE / Pineridge Press, 1993.

[39] A. Majda. *Compressible Fluid Flow and Systems of Conservation Laws in Several Space Variables*. Springer-Verlag, 1984.

[40] A. Majda and J. Sethian. The derivation and numerical solution of the equations for zero Mach number combustion. *Comb. Sci. and Tech.*, 42:185–205, 1985.

[41] B. Mercier and G. Raugel. Résolution d'un probème aux limites dans un ouvert axisymétrique par éléments finis en r, z et séries de fourier en θ. *R.A.I.R.O. Analyse numérique / Numerical Analysis*, 16(4):405–461, 1982.

[42] C.-D. Munz, S. Roler, R. Klein, and K. J. Geratz. The extension of incompressible flow solvers to the weakly compressible regime. *Theoretical and Computational Fluid Dynamics*, 1994.

[43] R. Poso. *MV++ v. 1.5a - Matrix/Vector Class - Reference Guide*. National Institute of Standards and Technology, March 1997.

[44] I. L. Ryhming. *Dynamique des Fluides*. Tec & Doc, 1990.

[45] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, Publishing Company, 1996.

[46] J. Segatz. *Simulation und Optimierung von laminaren Strömungsreaktoren*. PhD thesis, University of Heidelberg, 1995.

[47] J. Segatz, R. Rannacher, J. Wichmann, C. Orlemann, T. Dreier, and J. Wolfrum. Detailed numerical simulations in flow reactors: A new approach in measuring absolute rate constants. *J. Phys. Chem.*, 100:9323–9333, 1996.

[48] F. T. Suttmeier. *Adaptive Finite Element Approximation of Problems in Elasto-Plasticity Theory*. Phd thesis, University of Heidelberg, 1996.

[49] R. Temam. *Navier-Stokes Equations*. North-Holland, 1984.

[50] T. E. Tezduyar and Y. J. Park. Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Comp. Meth. App. Mech. and Eng.*, 59:307–325, 1986.

[51] L. Tobiska and R. Verfürth. Analysis of a streamline diffusion finite element method for the Stokes and Navier-Stokes equations. *SIAM J. Numer. Anal.*, 33(1):107–127, 1996.

[52] S. P. Vanka. Block-implicit multigrid solution of navier-stokes equations in primitive variables. *J. Comp. Phys.*, 65:138–158, 1986.

[53] R. Verfürth. *A Review of A Posteriori Estimation and Adaptive Mesh-Refinement Techniques, Advances in Numerical Mathematics.* Wiley/Teubner, New York-Stuttgart, 1996.

[54] L. Waldmann and E. Trübenbacher. Formale kinetische Theorie von Gasgemischen aus anregbaren Molekülen. *Zeitschr. Naturforschg.*, 17a:363–376, 1962.

[55] J. Warnatz. Modelling and simulation of reacting flows including detailed chemical reaction. In M. Hafez and K. Oshima, editors, *Computational Fluid Dynamics Review 1995.* John Wiley and Sons, 1995.

[56] J. Warnatz and U. Maas. *Technische Verbrennung.* Springer Verlag, Berlin, 1992.

[57] J. Wichmann. Untersuchung homogener und heterogener Schwingungsrelaxationsprozesse von Wasserstoff im Strömungsreaktor mit der Kohärenten Antistokes Raman Spektroskopie,. Master's thesis, University of Heidelberg, 1993.

[58] F. A. Williams. *Combustion Theory.* Addison-Wesley Publishing Compagny, 1985.

[59] G. Wittum. On the robustness of ILU smoothing. *SIAM J. Sci. Stat. Comp.*, 10, 1989.

[60] G. Zhou. How accurate is the streamline diffusion finite element method? *Preprint IWR, University of Heidelberg*, 92–22, 1995.