Karlsruhe Institute of Technology
Department of Chemical and Process Engineering
Engler-Bunte-Institute
Division of Combustion Technology
Prof. Dr.-Ing. Henning Bockhorn

Bachelor Thesis

Implementation and Validation of a Solver for Direct Numerical Simulations of Turbulent Reacting Flows in OpenFOAM

by

cand. chem. ing. Henning Bonart

Evaluator: Prof. Dr.-Ing. Henning Bockhorn
Supervisor: Dipl.-Ing. Feichi Zhang

October 2012

Karlsruher Institut für Technologie
Fakultät für Chemieingenieurwesen und Verfahrenstechnik
Engler-Bunte-Institut
Bereich Verbrennunsgtechnologie

Aufgabenstellung zur Bachelorarbeit

cand. chem. ing. Henning Bonart

Implementierung und Validierung eines Lösers für Direkte Numerische Simulationen von turbulenten, reagierenden Strömungen in OpenFOAM

Ziel dieser Arbeit ist die Implementierung eines Lösers für die dreidimensionale Direkte Numerische Simulation (DNS) von turbulenten, reagierenden Strömungen in der Entwicklungsumgebung OpenFOAM. Dabei soll die Berechnung der molekularen Flüsse auf der kinetischen Gastheorie basieren. Darüber hinaus sollen verschiedene komplexe chemische Reaktionsmechanismen eingebunden werden können.

Mittels der Simulation einer eindimensionalen, laminaren Vormischflamme soll der Löser mit Ergebnissen von CHEMKIN/PREMIX im Hinblick auf Diffusions- und Reaktionsvorgänge validiert werden. Danach soll gezeigt werden, das dreidimensionale DNS von turbulenten, reagierenden Strömungen mit dem Löser möglich sind. Zur Durchführung der Simulationen soll der hauseigene Rechencluster sowie der CRAY XE6 vom Hochleistungsrechenzentrum Stuttgart genutzt werden.

Die Arbeit gliedert sich in folgende Arbeitsschritte:

1. Einarbeitung in die Theorie der turbulenten Strömung und Verbrennung
2. Aussuchen eines geeigneten Lösers in OpenFOAM als Basisströmungslöser
3. Erweiterung der Transportgleichungen des Basislösers auf multikomponenten Reaktionsmischungen
4. Einbindung von Cantera zur Berechnung der Eigenschaften von multikomponenten Reaktionsmischungen wie Dichte, spezifische Wärmekapazitäten, Wärmeleitkoeffizient, Diffusionskoeffizienten, chemische Quellterme etc.
5. Validierung des neuen Lösers anhand einer eindimensionalen laminaren Vormischflamme
6. Durchführung einer dreidimensionalen DNS einer turbulenten Vormischflamme
7. Beschreibung von Theorie und Implementierung sowie Darstellung und Diskussion der Ergebnisse in einer schriftlichen Ausarbeitung
8. Vorstellung der Bachelorarbeit in einem öffentlichen Seminarvortrag

Betreuer: Dipl.-Ing. Feichi Zhang
Aufgabensteller: Prof. Dr.-Ing. Henning Bockhorn

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, October 2012 ....
(**Henning Bonart**)

Contents

Figures, Tables and Listings III

Nomenclature V

1. Introduction 1

1.1. Motivation 1

1.2. Objectives 3

1.3. Structure of the Thesis 3

2. Basic Aspects of Turbulent Combustion 5

2.1. Structure of Laminar Premixed Flames 5

2.2. Characteristics of Turbulent Flow 7

2.2.1. Energy Cascade and Kolmogorov Scales 7

2.3. Interaction between Premixed Flames and Turbulent Flow 8

2.3.1. Turbulent Flame Speed 9

2.3.2. Flame Stretch Rate 10

2.3.3. Combustion Regimes 11

3. Mathematical Description of Chemically Reacting Flows 13

3.1. Statistical Thermodynamics and the Rigorous Kinetic Theory 13

3.1.1. The Boltzmann Equation 14

3.1.2. Enskog's General Transport Equation 15

3.1.3. Chapman-Enskog Theory 15

3.2. Transport Equations for Chemically Reacting Flows 16

3.2.1. Equations of Mass 16

3.2.2. Equation of Momentum 17

3.2.3. Equation of Energy 17

3.3. Multicomponent Molecular Transport 18

3.3.1. Species Mass Flux 18

3.3.2. Momentum Flux 20

3.3.3. Heat Flux 21

3.4. Reaction Kinetics 21

3.5. Chemical Time Scales 22

4. Numerical Solution of Partial Differential Equations 23

4.1. The Finite Volume Method 23

4.1.1. Approximation of Integrals 24

4.1.2. Interpolation and Differentiation Procedures 25
4.2. Methods for Unsteady Problems 26
4.3. Solution of Linear Equation Systems 27

5. Implemented Solver in OpenFOAM and the Cantera Interface 29
5.1. Software Packages 29
5.2. Connection of OpenFOAM with Cantera 30
5.2.1. Structure of the Interface 31
5.2.2. Integration of the Interface into OpenFOAM 31
5.2.3. Exchange of Data between OpenFOAM and Cantera 32
5.3. Implementation of the Solver 34
5.3.1. OpenFOAM's Approach for Transport Equations 34
5.3.2. Implementation of Species Mass Equations 35
5.3.3. Implementation of Sensible Enthalpy Equation 36

6. Validation of the Solver 37
6.1. Case Description 37
6.2. Numerical conditions 38
6.3. Comparison with CHEMKIN/PREMIX 40
6.3.1. Temperature and Species Mass Fractions 40
6.4. Influence of the Grid Resolution 42
6.5. Species Production Rates 44
6.6. Computation of Chemical Time Scales 44

7. Direct Numerical Simulation of a Turbulent Premixed Flame in 3D 46
7.1. Numerical Setup 46
7.2. Initial conditions 48
7.3. Topological Results 49
7.4. Parallel Performance of the Cray XE6 (HERMIT) 52

8. Summary and Perspective 54

A. Code Fragments for OpenFOAM in C++3 55

B. Reaction Mechanism 57

Figures

1.1. World energy-related CO_2 emissions by scenario 1

2.1. Species and temperature profiles for a laminar, premixed flat methane-oxygen flame. 6
2.2. Schematic representation of the turbulent kinetic energy spectrum E as a function of the wavenumber k. 8
2.3. Kinematic interaction between a turbulent eddy and a propagating flame front. 9
2.4. Schematic representation of the turbulent flame velocity. 10
2.5. Schematic classification of turbulent combustion regimes. 11

3.1. Graphical expression of the Boltzmann equation. 14

4.1. Parameters used in the Finite Volume Method. 24
4.2. Example of a 2D, structured, non-orthogonal grid to simulate the flow through a duct. 25
4.3. Approximation of the time integral over an interval. 27

5.1. Inheritance and dependency diagram for the coupling library. 32
5.2. Initialization sequence of the coupling library during runtime. 33
5.3. Data exchange between OpenFOAM and Cantera through the interface *canteraFoamModel*. 34

6.1. Physical domain of the laminar, premixed flame. 38
6.2. Numerical domain of the laminar, premixed flame. 39
6.3. Temperature and species mass fraction profiles obtained from OpenFOAM and from CHEMKIN/PREMIX. 41
6.4. Profiles obtained from OpenFOAM with 600 Cells and 3000 Cells. 43
6.5. Reaction rates obtained from OpenFOAM. 44
6.6. Calculated chemical time scales τ for T, O_2 and OH. 45

7.1. Numerical domain of the turbulent, premixed flame. 47
7.2. Initial velocity and temperature fields of the premixed turbulent flame. . 48
7.3. From the initial conditions calculated Kolmogorov scales. 48
7.4. Isosurface of Y_{CH_4}. Bottom with eddy dissipation rate and backside with vorticity. 49
7.5. Two dimensional slices of the temperature field and the vorticity field with heat release rate of the turbulent premixed flame. 50

7.6. Different mass fractions and reaction rates. 51
7.7. Scale-up on Cray XE6 with a grid of 2×10^6 cells. 53

Tables

6.1. Physical conditions of the simulation with OpenFOAM. 37
6.2. Numerical setups. 38
6.3. In OpenFOAM used discretization schemes for the mathematical terms. . 39
6.4. Comparison of distinctive results obtained with OpenFOAM and CHEMKIN/PREMIX. 40
6.5. Comparison of distinctive results obtained with 600 Cells and 3000 Cells. 42

7.1. Physical and numerical conditions of the turbulent, premixed flame in 3D. 46

Listings

A.1. Implementation of the species equation 5.8 in OpenFOAM. 55
A.2. Implementation of the energy equation 5.12 in OpenFOAM. 56
A.3. Initialization during run time and performing a downcast to obtain access to derived class functions in OpenFOAM. 56

B.1. The used reaction mechanism in CHEMKIN format 57

Nomenclature

Latin symbols

Δh_f^0	Specific standard-state heat of formation	[J/kg]
$\mathcal{D}$	Binary diffusion coefficient	[$\mathrm{m^2/s}$]
$\mathcal{M}$	Name	[-]
$\overline{M}$	Molecular weight	[kg/kmol]
A	Surface	[$\mathrm{m^2}$]
$A^\star$	Pre-exponential constant	[varies]
c_p	Specific heat at constant pressure	[J/(kg·K)]
Co	Courant number	[-]
D	Ordinary multicomponent diffusion coefficient	[$\mathrm{m^2/s}$]
d	Distance	[m]
d	Molecular driving force	[1/m]
D_k^T	Thermal diffusion coefficient	[kg/(m·s)]
Da	Damköhler number	[-]
E	Energy	[J]
E_a	Activation energy	[J/mol]
F	External force	[$\mathrm{kg\cdot m/s^2}$]
f	Cell face	[-]
f	Velocity distribution function	[$\mathrm{s^3/m^6}$]
g	Gravitational force	[$\mathrm{kg\cdot m/s^2}$]
H	Terms of higher order	[-]
h	Specific enthalpy	[J/kg]
h_c	Specific chemical enthaly	[J/kg]
h_s	Specific sensible enthaly	[J/kg]
I	Total number of chemical reactions	[-]
j	Diffusive mass flux	[$\mathrm{kg/(m^3\cdot s)}$]
K	Total number of chemical species	[-]
k	Rate constant	[varies]
k	Wavenumber	[1/m]
Ka	Karlovitz number	[-]
L	Matrix for the computation of transport quantities	[-]
l	Length	[m]
Le	Lewis number	[-]
m	Mass	[kg]
N	Number of molecules	[-]
n	Number of moles	[-]
p	Pressure	[Pa]

Q	Source	[-]
q	Heat flux	[J/(m^2·s)]
r	Chemical reaction rate	[kg/m^3]
R_s	Specific gas constant	[J/(kg·K)]
Re	Reynolds number	[-]
S	Surface	[m^2]
s	Flame speed	[m/s]
Sc	Schmidt number	[-]
T	Temperature	[K]
t	Time	[s]
T_O	Inner reaction layer temperature	[K]
V	Mass diffusion velocity	[m/s]
V	Volume	[m^3]
v	Velocity	[m/s]
w	Convective or diffusive flux	[-]
X	Mole fraction	[-]
x, y, z	Direction	[m]
Y	Mass fraction	[-]

Greek symbols

β	Temperature exponent	[-]
Γ	Molecular collision	[-]
δ	Thickness of inner reaction layer	[m]
ϵ	Rate of dissipation of turbulent kinetic energy	[m^2/s^3]
θ	Perturbation function	[-]
Λ	Transport coefficient	[-]
λ	Air equivalence number	[-]
λ_0	Thermal conductivity	[W/(m·K)]
μ	Dynamic viscosity	[kg/(m·s)]
ν	Kinematic viscosity	[m^2/s]
ν	Stoichiometric coefficient	[-]
ξ	Ordering parameter	[-]
ρ	Mass density	[kg/m^3]
σ	Collision diameter	[m]
τ	Stress tensor	[kg/(m·s^2)]
τ	Time scale	[s]
ϕ	Physical quantity	[-]
ψ	Quantity	[-]
Ω	Reduced collision integral	[-]
ω	Rate of progress	[kg/(m^3·s)]

Indices and Accents

0	Unburnt

$\bar{}$	Mean value
$\dagger$	Transposed
$\dot{}$	Time derivation/rate
∞	Surrounding
$\vec{}$	Vector
c	Characteristic
i	Species i
k	Species k
L	Laminar
T	Turbulent

Constants .....

k_B	Boltzmann constant ($1.3806488 \cdot 10^{-23}$)	[J/K]
N_A	Avogadro constant ($6.02214129 \cdot 10^{23}$)	[mol^{-1}]
R	Universal gas constant (8.3144621)	[J/(mol·K)]

Abbreviations

1D	One dimension
3D	Three dimensions
fvc	finite volume calculus
fvm	finite volume method
CFD	Computational Fluid Dynamics
BLAS	Basic linear algebra subprograms
CFL	Courant–Friedrichs–Lewy condition
CG	Conjugate gradients
CHEMKIN	Software for chemical kinetics
DNS	Direct numerical simulation
FDM	Finite Difference Method
FVM	Finite Volume Method
GPL	General Public License
LAPACK	Linear algebra package
LES	Large eddy simulation
LU	Lower upper
MATLAB	Matrix laboratory
MPI	Message Passing Interface
MPI	Message passing interface, e.g. MPICH2/OpenMPI
NRBC	Non-reflecting boundary conditions
ODE	Ordinary differential equation
OpenFOAM	Open Source Field Operation and Manipulation, registered trademark of OpenCFD Limited
PBiCG	Preconditioned biconjugated gradient
PCG	Preconditioned conjugate gradient
PDE	Partial differential equation

PISO	Pressure implicit with split operator
RANS	Reynolds-averaged Navier–Stokes
SIMPLE	Semi-implicit method for pressure-linked equations
SUNDIALS	Suite of nonlinear and differential/algebraic equation solvers

1. Introduction

1.1. Motivation

Chemically reacting flows impact many aspects of human life via, for example, medicinal chemistry, chemical synthesis, material processing and combustion processes [1]. Combustion of hydrocarbon fossil fuels (or fuels stemming from renewable sources) is still an important part of almost every energy conversion [2].

Technical applications based on combustion processes are important for transportation, power generation and chemical engineering. At the same time pollution caused by combustion processes leads to environmental problems. For instance, the contribution of CO_2 emissions to global warming is now considered to be proven and may be a main threat to our standard of living. Figure 1.1 shows the world-energy related CO_2 emissions by different scenarios. To reach the 450 Scenario[1] an improvement of direct and indirect energy conversion efficiency is without alternative [2].

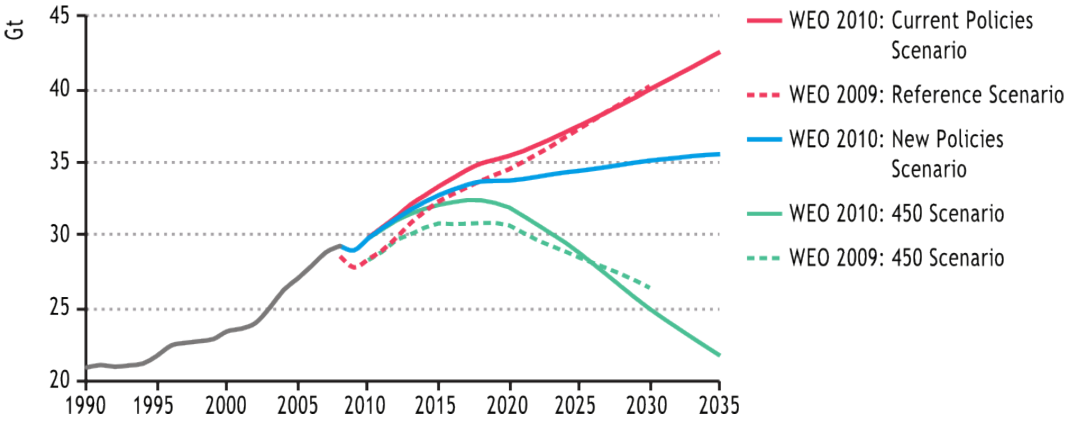

Fig. 1.1.: World energy-related CO_2 emissions by scenario. Figure taken from [2].

Other pollutants, such as unburnt hydrocarbons, soot, nitrogen oxides and sulfur oxides, can contribute to health hazards and cause smogs or acid rain [3]. However, due to stricter government regulations for, e.g., the automotive industry and the energy sector, pollutants emission has been reduced. In order to be conform to such legal regulations

[1] The 450 Scenario describes pathways to a long-range CO_2 concentration in the atmosphere of 450 parts per million, see [2].

and to meet the public expectation about cleaner and more efficiently industrial processes as well as to reach the 450 Scenario target of the Internal Energy Agency, it is still of crucial importance to improve combustion technology.

Since most of the industrial applications of combustion involve turbulent flows, a deeper understanding of the interplay between turbulence and combustion is needed. In many cases turbulence increases combustion. On the other hand the heat release leads to gas expansion and density variations and therefore influences the turbulent flow [4]. This interaction of turbulence and chemical reaction is still not fully understood. Due to the small length and time scales, details of the mixing process, as well as the structures in turbulent flows, have not been fully investigated, yet. Therefore, a deeper understanding of the interplay between turbulence and chemical reaction processes is of large interest in order to improve efficiency in combustion.

The mechanisms involved in such turbulent combustion flows have been the object of numerous theoretical, experimental and numerical scientific research in the last century [4–6]. Thereby, experimental studies have greatly advanced our knowledge on turbulent combustion. Recently, experiments that simultaneously measured the turbulent velocity fields and the reacting scalar concentrations, together with the local temperature, have contributed to a more realistic understanding of such turbulent burning processes. However, in order to gain detailed insight into the fundamental physics of the turbulence-chemistry interaction on small space and time scales, numerical simulations are to date without alternative.

In numerical simulations of turbulent combustion, additionally to the standard complexities of turbulent non-reacting simulation, other difficulties, such as the strong heat release and the complex chemistry behind the combustion process, have to be taken into account [7]. The computational grid has to be sufficient small in order to resolve the smallest eddies in the turbulent flow. Due to numerical stability reasons very small time steps are necessary. Hence, direct numerical simulations of turbulent reacting flows demand a prohibitive amount of computational time. However, the huge progress in computer technology in the past two decades now offers the opportunity for direct numerical simulations of three-dimensional (3D) turbulent reacting flows for certain cases. This allows for studying the chemistry-turbulence interaction in more detail (see e.g. [8, 9]).

For a direct numerical simulation the underlying physical and chemical models which describe chemical kinetics or molecular transport are required to be as precise as possible. Thanks to extensive research in chemical kinetics a lot of chemical mechanisms can now be modeled with great precision. Consequently, highly realistic chemical kinetic models are implemented in many fluid dynamics codes. However, the use of multicomponent transport equations based on the Maxwell-Stefan diffusion model or the rigorous kinetic theory of ideal gases is still rare in computational fluid dynamics. The reason is probably the additional complexity of the implementation and the higher computational cost. Numerous studies have shown that, when omitting multicomponent diffusion equations, significant errors occur in simulations of laminar flames [1, 10–13]. As to turbulent flow, scientific publications comparing different diffusion models are very rare but it has been argued by [9, 14] that accuracy should be improved when using the full multicomponent transport equations.

1.2. Objectives

In the present thesis the main objective is to implement and validate a DNS solver which is able to simulate turbulent, chemically reacting flows with a detailed diffusion modeling and complex chemical kinetics. The equations are implemented with as little simplifications as possible and mostly derived from the rigorous kinetic theory of gases. The C++ toolbox OpenFOAM is used as the underlying structure for the DNS solver. The main advantages of OpenFOAM are its high numerical accuracy and its free availability [15].

Moreover, multicomponent transport coefficients, which are obtained from the extended Chapman-Enskog theory [16] and based on the molecular properties of the gases, are used. These multicomponent diffusion coefficients are calculated with Cantera, an open source chemical kinetics software. Cantera features a high calculation speed when calculating chemical source terms based on complex chemical mechanisms as well as efficient calculation routines for multicomponent coefficients [17]. The interconnection between OpenFOAM 2.0.1 and Cantera is newly programmed based on an library by [18] for OpenFOAM 1.5.

To validate the solver laminar premixed flames are simulated and compared to calculations obtained with CHEMKIN/PREMIX with respect to numerical accuracy and stability. Furthermore, the simulation of a turbulent premixed flame in 3D is carried out. For that purpose, the solver, as well as the OpenFOAM framework and the Cantera routines, are adapted for the hardware structure of the high performance cluster Cray XE6 in Stuttgart, Germany, where the simulations have been carried out.

1.3. Structure of the Thesis

The thesis is organized in the following way: After the current introduction a general discussion on the physics of the combustion and flow dynamics in chapter 2 is presented. In the following chapter 3 the mathematical description of chemically reacting flows is presented with an introduction to statistical thermodynamics and the kinetic theory of gases, from which the transport equations, as well as the equations for the molecular fluxes are derived. A brief discussion of chemical kinetics follows, to close the mathematical description of the partial differential equations. The organization of the thesis from sections 3.1 to 3.4 has been designed in a way to gradually guide the reader from the most basic physical model as a starting point towards the later implemented equations. The numerical methods used in the simulation of reacting flows are presented in chapter 4. The finite volume method is discussed, as well as methods for unsteady problems and solution strategies for linear equation systems. Chapter 5 gives at first an overview over the used software. In the following, the coupling library between OpenFOAM and

Cantera is presented and the new solver for DNS of chemically reacting flows with multicomponent diffusion modeling is introduced. In chapter 6 the developed solver is applied to numerical simulations of flat premixed flames for validation purpose. The results are compared with calculations obtained from CHEMKIN/PREMIX. Finally the solver is used for DNS of 3D turbulent premixed flames and the results are briefly discussed in chapter 7. The thesis finishes with a conclusion and a perspective.

2. Basic Aspects of Turbulent Combustion

Combustion is a highly exothermic, chemical reaction between a fuel and an oxidizer. There are two general types of combustion processes which can be distinguished by the nature of their mixing state. In a premixed combustion the composition of fuel and oxidizer is spatially homogeneous while in a non-premixed combustion the unburnt fuel composition varies in space such that mixing and chemical reaction occur simultaneously. Between these two extremes one often encounters so called partially-premixed flames. Secondly, combustions may be classified with respect to the underlying flow regimes which can be either laminar or turbulent. Latter is encountered in most practical combustion systems such as combustion engines, aircraft engines, industrial burners and furnaces.

In the following sections a brief description of the different flame types with respect to their mixing state and fluid motion is presented and some fundamental definitions are introduced.

2.1. Structure of Laminar Premixed Flames

A flow is defined laminar if the fluid layers move smoothly in such a way that they remain ordered and do not mix in the plane perpendicular to the direction of the flow [19]. Whether a flow is laminar or not depends on the ratio of the inertial and viscous forces which is used to define the Reynolds number

$$\mathrm{Re} = \frac{l_c v \rho}{\mu} \,, \tag{2.1}$$

where l_c is a characteristic length, v is the flow velocity, ρ is the mass density and μ is the dynamic viscosity. If a critical value of Re is reached the laminar flow starts to become unstable and changes to turbulent flow. For example in an internal channel flow, this happens at $Re \approx 2000$.

Laminar premixed flames propagate towards a premixed mixture of fuel and air. They occur in gas ranges, heating appliances and Bunsen burners. The understanding of laminar premixed flames is a prerequisite to study turbulent premixed flames [10]. Due its simple configuration laminar premixed flat flames have been an important object in numerical combustion science. Therefore a lot of specialized numerical codes, like CHEMKIN/PREMIX [20] and Cantera [17], exist.

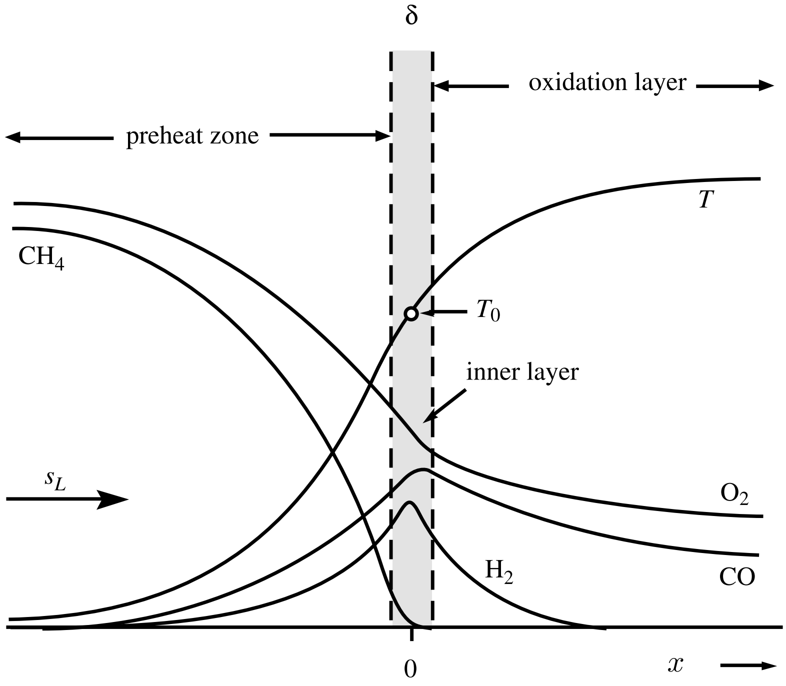

Fig. 2.1.: Species and temperature profiles for a laminar, premixed flat methane-oxygen flame. Figure taken from [4].

In order to distinguish different mixtures of fuel and oxidizer the air number is defined as [10]

$$\lambda_{air} = \frac{X_{air}/X_{fuel}}{X_{air,stoich.}/X_{fuel,stoich.}} \ , \tag{2.2}$$

where X_{air} (X_{fuel}) are the mole fractions of the air (fuel) in the mixture, and $X_{air,stoich.}$ ($X_{fuel,stoich.}$) are the mole fractions of the air (fuel) in the stoichiometric mixture. Based on the value of λ_{air} premixed combustions can be divided into different classes: rich combustion with $\lambda_{air} < 1$, stoichiometric combustion with $\lambda_{air} = 1$ and lean combustion with $\lambda_{air} > 1$.

The schematic structure of a premixed, flat, methane-air flame is shown in Figure 2.1. It consists of three main regions. In the chemically inert preheating zone, heat released from the reaction is transported by conduction. The reaction layer is typically thin and called fuel consumption layer or inner layer with thickness δ and temperature T_O. This layer is responsible for keeping the reaction process alive. Here the fuel is consumed and the radicals are depleted by chain-breaking reactions. The inner layer temperature T_O corresponds to the crossover temperature between chain-branching and chain-breaking reactions. In the oxidation layer the final oxidation to the products is accomplished and the temperature reaches its maximum. The thickness of the reaction zone δ can be calculated from the temperature profile with [21]

$$\delta = \frac{T_2 - T_1}{\max\left(\left|\frac{\partial T}{\partial x}\right|\right)} \ . \tag{2.3}$$

In a steady flow of premixed gas, the flame propagates upstream until the flow velocity normal to the flame front is equal to the laminar flame speed s_L. The laminar flame speed can be calculated from the integral of the burning rate across the flame brush with [21]

$$s_L = -\frac{1}{\rho_0 Y_{fuel,0}} \int_{-\infty}^{+\infty} \dot{r}_{fuel} \, \mathrm{d}x \; . \tag{2.4}$$

The laminar burning velocity dependence on the fuel type, the air number, the pressure, the unburnt flow temperature, etc. and is a very important characterization of combustions [4].

2.2. Characteristics of Turbulent Flow

As mentioned in the section before the occurrence of turbulent flow essentially depends on the ratio of inertial and viscous forces in the fluid. In a turbulent flow the destabilizing inertial forces exceed the stabilizing viscous forces such, that the generation rate of vortices is higher then the viscous dissipation rate [19]. Therefore, turbulence arises from instabilities associated with large Re numbers.

To further characterize the flow, a turbulent Reynolds number Re_T is introduced [22]:

$$Re_T = \frac{v' l_T}{\nu} \tag{2.5}$$

Here, v' is the turbulence intensity or velocity fluctuation, ν is the kinematic viscosity and l_T is the turbulent length scale. The turbulent Reynolds number represents the ratio of turbulent transport to molecular transport of momentum.

Compared to laminar flows, turbulent flows have no well-defined values for velocities and physical scalars. The flow regime is characterized by transient chaotic and stochastic property changes. Variations of velocity lead to fluctuations in physical scalars such as density, temperature or transport quantities. Since the physics of turbulence are not fully understood the integration of chemical reactions is even more difficult.

2.2.1. Energy Cascade and Kolmogorov Scales

Turbulence can be considered to be composed of eddies of different sizes. Large eddies are unstable and break up, transferring their energy E to somewhat smaller eddies which break up again. This energy cascades continuous until the Reynolds number of the eddy is sufficiently small that the eddy motion is stable and molecular viscosity is effective in dissipating the kinetic energy [22].

Figure 2.2 shows the energy spectrum measured for all wavelengths k. For small wavenumbers corresponding to large scale eddies the energy per unit wavenumber increases with a power law between k^2 and k^4 . This range is not universal and is determined by large scale instabilities, which depend on the boundary conditions of the flow. The spectrum attains a maximum at a wavenumber that corresponds to the integral or

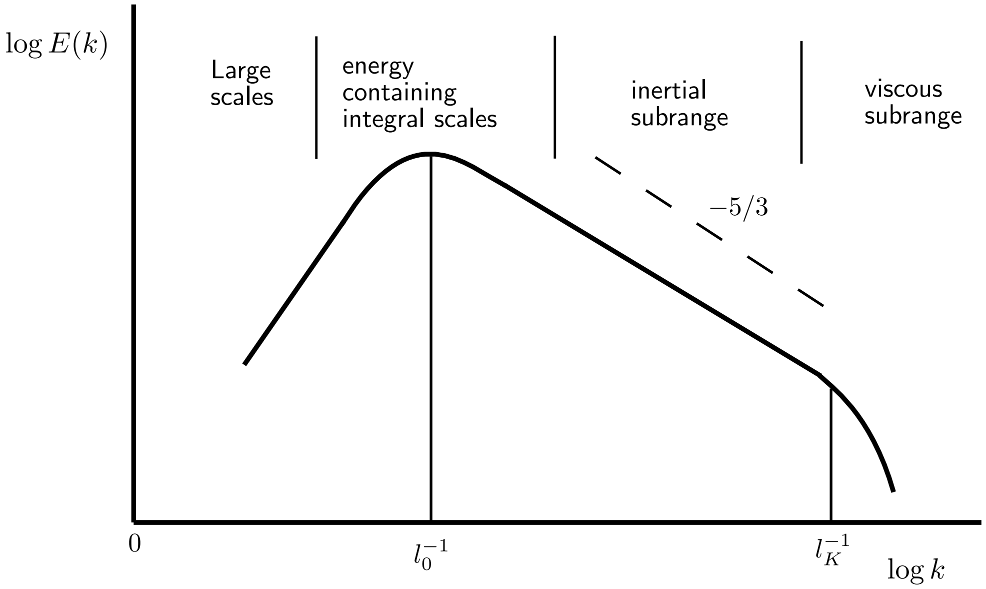

Fig. 2.2.: Schematic representation of the turbulent kinetic energy spectrum E as a function of the wavenumber k.

turbulence macroscale l_0, since eddies of that scale contain most of the kinetic energy. For larger wavenumbers corresponding to the inertial sub-range the energy spectrum decreases following the $k^{-5/3}$ law. Finally, there is a cutoff at the Kolmogorov microscale l_K. In the viscous sub-range, the energy per unit wavenumber decreases exponentially owing to viscous effects. Therefore, the Kolmogorov microscale represents the smallest length scales associated with a turbulent flow [4].

To determine the necessary grid resolution and time step in numerical simulations of turbulent flows one has to know about the value of the smallest scales. The Kolmogorov microscale for length l_K and time τ_K are defined in [22] to

$$l_K = (\nu^3/\epsilon)^{1/4} \quad \text{and} \quad \tau_K = (\nu/\epsilon)^{1/2} \;, \tag{2.6}$$

in which ϵ is the rate of dissipation of turbulent kinetic energy defined to

$$\epsilon = 2\nu \left\langle s_{ij} s_{ij} \right\rangle = \nu \left\{ \left\langle \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right\rangle \left\langle \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right\rangle \right\} \;, \tag{2.7}$$

and ν is the kinematic viscosity. Here, s_{ij} is the fluctuating strain rate and $\langle\rangle$ is the expectation.

2.3. Interaction between Premixed Flames and Turbulent Flow

A highly exothermic chemically reaction and turbulence can have large influences on each other. The flow is strongly accelerated by passing the flame front due to the thermal expansion. In addition, turbulent eddies wrinkle the flame front (as illustrated in figure 2.3) and enhance the chemical reaction. In some cases on the other hand, the flow can completely inhibit the chemical reaction and lead to flame quenching [4].

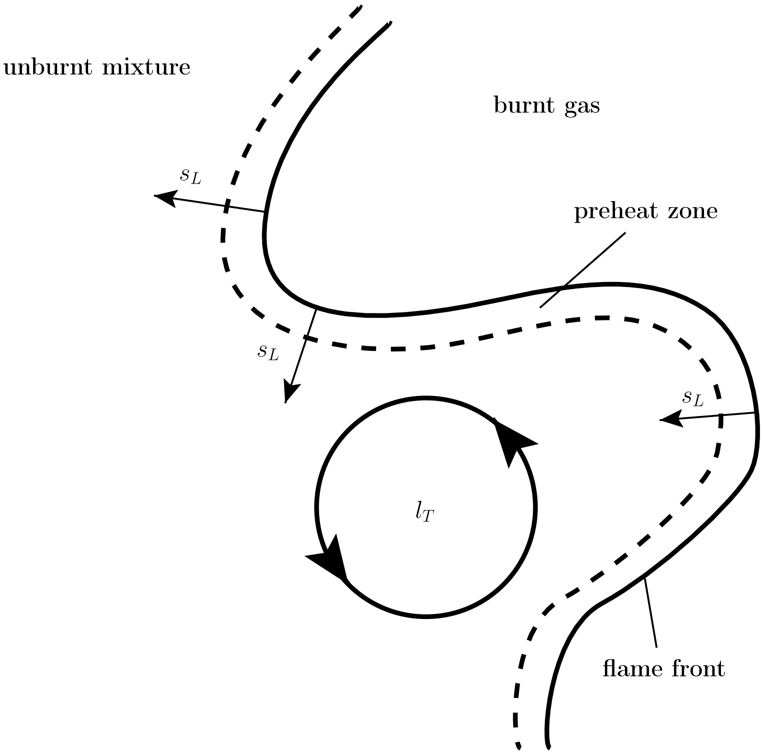

Fig. 2.3.: Kinematic interaction between a turbulent eddy and a propagating flame front.

2.3.1. Turbulent Flame Speed

Unlike the laminar flame speed, which depends mostly on the thermal and chemical properties of the unburnt mixture, the turbulent flame speed s_T also depends on the transient interaction between turbulence and chemical reaction. As shown in figure 2.4 the instantaneous flame front represents a large area A_T propagating with the locally laminar flame speed s_L. One can see, that the turbulent flame front is much greater than the mean flame area $A_T \gg \bar{A}$. For an observer traveling with the flame, the turbulent flame speed can be defined as the velocity at which reactants enter the flame zone in a direction normal to the flame. Thus, the turbulent flame speed can be expressed as

$$\dot{m} = \rho_0 A_T s_L = \rho_0 \bar{A} s_T \ , \tag{2.8}$$

with $\dot{m}$ equal the mass flow rate of the unburnt gas mixture and ρ_0 equal the unburnt mixture mass density. Consequently the ratio of turbulent flame speed to local laminar flame speed is equal to the ratio of wrinkled flame area to the mean flame area from which s_T can be calculated to

$$\frac{s_T}{s_L} = \frac{A_T}{\bar{A}} \rightarrow s_T = \frac{A_T}{\bar{A}} \ s_L \ . \tag{2.9}$$

In addition, the turbulent burning velocity can be calculated with

$$s_T = -\frac{1}{\bar{A}\rho_0 Y_{fuel,0}} \int_{-\infty}^{+\infty} \dot{r}_{Fl} \, \mathrm{d}V \ , \tag{2.10}$$

which is similar to equation (2.4) for the laminar burning velocity.

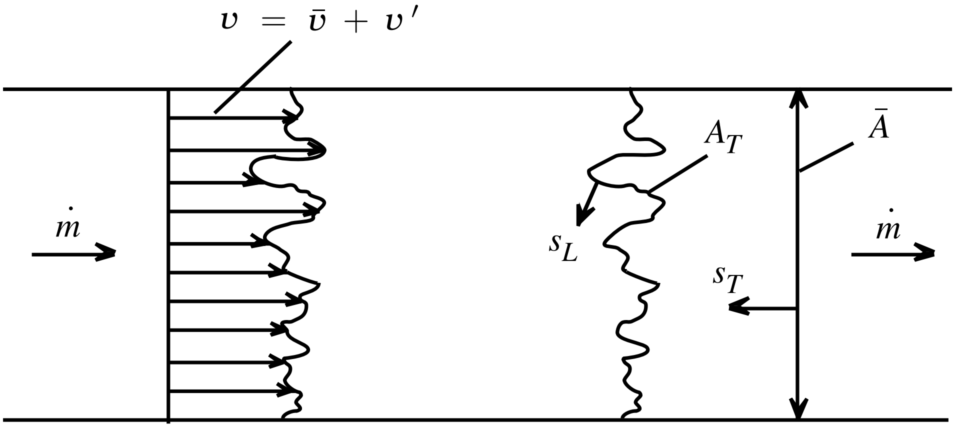

Fig. 2.4.: Schematic representation of the turbulent flame velocity. Figure similar to [4].

2.3.2. Flame Stretch Rate

The flame stretch rate is an important quantity in the understanding of flame phenomena such as extinction and the local structure of turbulent flames. In general, the flame stretch rate reads [21]

$$\kappa = \frac{1}{A_T}\frac{\mathrm{d}A_T}{\mathrm{d}t} \ , \tag{2.11}$$

which describes the fractional rate of change of a flame surface element A_T. Note, that this equation uses the substantial derivation of the surface to include the convective change. Because of the highly unsteady flame surface, A_T can not be easily calculated. Therefore, different approaches for equation (2.11) have been developed (see e.g. [23]). Here, an expression for the flame stretch rate of stoichiometric premixed flames is derived based on chemical reaction rates. Assuming, that the total inlet mass of reactants is consumed by chemical reactions with rate $\dot{r}$ in the flame volume $V_{Fl.}$ and taking equation (2.8) into account, one can write

$$\dot{m}_{fuel} = \int_{V_{Fl.}} \dot{r}_{fuel}(V) \ \mathrm{d}V = \rho_0 Y_{fuel,0} A_T s_L = \rho_0 Y_{fuel,0} \bar{A} s_T \ . \tag{2.12}$$

With $\bar{A} = \text{const.}$, $s_L = \text{const.}$ and $\rho_0 = \text{const.}$ it must apply, that

$$s_T \sim A_T \sim \int_{V_{Fl.}} \dot{r}_{fuel}(V) \ \mathrm{d}V \approx \sum_i \dot{r}_{fuel,i} V_i = V_{CV} \sum_i \dot{r}_{fuel,i} \ , \tag{2.13}$$

with V_{CV} as the volume of a single cell which is equal for all cells in an equidistant mesh. Consequently, it must apply locally in cell i, that

$$(s_T)_i \sim (A_T)_i \sim V_{CV}(\dot{r}_{fuel})_i \ . \tag{2.14}$$

Finally, by inserting the expressions in equation (2.11) the flame stretch rate κ can be expressed as

$$\kappa = \frac{1}{A_T}\frac{\mathrm{d}A_T}{\mathrm{d}t} = \frac{1}{\dot{r}_{fuel}}\frac{\mathrm{d}\dot{r}_{fuel}}{\mathrm{d}t}\ . \tag{2.15}$$

This equation describes the fractional rate of change of the fuel consumption rate $\dot{r}_{fuel}$. Note, that this equation is valid for every reactant.

2.3.3. Combustion Regimes

Diagrams defining regimes of premixed turbulent combustion in terms of velocity and length scale ratios have been proposed by many authors (see e.g. [4] for details). Here, a regime diagram, figure 2.5, is discussed following [21]. Therefore, two additional reduced numbers are used, the turbulent Damköhler number Da_T and the Karlowitz number Ka, defined as:

$$Da_T = \frac{\tau_T}{\tau_c} = \frac{l_T/v'}{\delta/s_L} \tag{2.16}$$

$$Ka = \frac{\tau_c}{\tau_K} = \frac{\delta^2}{l_K^2} \tag{2.17}$$

Da_T describes the ratio between the turbulent time scale τ_T and the characteristic chemical time scale τ_c. The interaction of the chemical reaction with the dissipative turbulent structures of the flow field is described with Ka, which is the ratio of the characteristic chemical time scale and the characteristic time scale of the smallest Kolmogorov eddy τ_K. Additionally, the turbulent Reynolds number introduced before is used in the diagram 2.5.

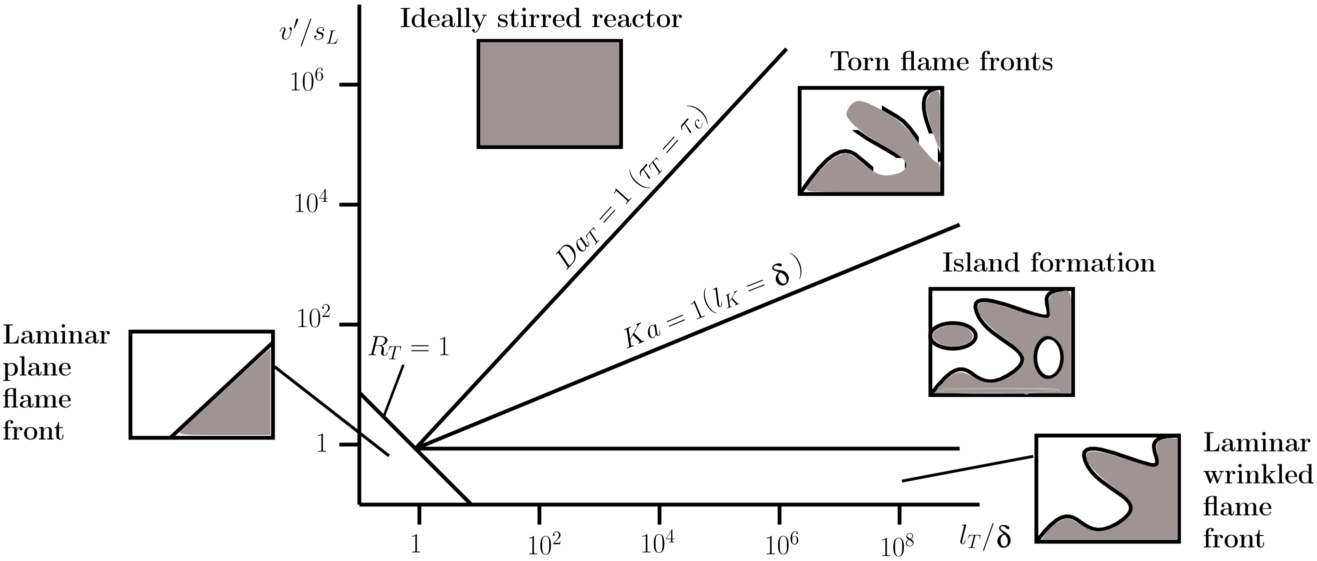

Fig. 2.5.: Schematic classification of turbulent combustion regimes.

By using the reduced numbers Re_T, Ka and Da, turbulent premixed combustions can be classified into three major categories. The thin reaction zone regime or flamelet regime exists for $Ka < 1$, which means, that the flame thickness is smaller than the Kolmogorov length scale. It assumes, that the flame structure is not effected by turbulence. The flame sheet is wrinkled by vortices not small enough to enter it. At $Da > 1$ and $Ka > 1$, the smallest eddies can enter into the flame front and enhance the diffusion inside of the flame, which leads to a thickened flame. Here, the turbulent flow is intense enough to generate eddies able to effect the structure of the reaction zone. Since it is expected, that the lifetime of such eddies is very short, their impact on the reaction zone is thus limited. If the reaction time is greater than the time needed for turbulent fluctuation, a well stirred reactor regime occurs. It is characterized by $Da < 1$ and $Ka > 1$. Consequently, most of the eddies can enter into the reactive-diffusive layer during their lifetime and enhance the diffusivity within the reaction zone. It is even possible, that nearly all of the eddies are embedded into the reaction zone. Hence, the flame front is very thick and behaves similar to an ideally stirred reactor. Since in a turbulent flow a wide range of length scales occur, a turbulent premixed flame is not represented by a single point in figure 2.5, but by a zone that may cross the different regimes.

3. Mathematical Description of Chemically Reacting Flows

In the following an introduction to statistical thermodynamics and the rigorous kinetic theory of gases (see [1, 24, 25]) is presented. The focus in this section is on the Chapman-Enskog formulation and will not go through detailed derivations of all results. After, the transport equations, as well as the equations for the molecular fluxes are derived from the Chapman-Enskog formulation. A brief discussion of chemical kinetics follows, to close the mathematical description of the partial differential equations.

3.1. Statistical Thermodynamics and the Rigorous Kinetic Theory

Statistical thermodynamics apply probability theory to a large number of particles in order to add a molecular-level interpretation of the macroscopic thermodynamic quantities described by classical thermodynamics. It therefore provides a mathematical description of thermodynamic, chemical kinetic and transport quantities which are needed in the numerical simulation of chemically reacting flows [24].

The rigorous kinetic theory of gases is used to relate the motion or kinetic energy of a large number of molecules to macroscopic thermodynamic quantities [1]. The theory makes the assumptions that

(i) only binary collisions occur and that therefore the pressure of the gas mixture is moderate

(ii) quantum-mechanical effects are negligible and that therefore temperature has to be moderate

(iii) the dimensions of the containing vessel and any obstacle therein are large compared to the molecule mean free path which means that the average distance separating the gas particles is large compared to their size.

Strictly speaking, the rigorous kinetic theory of gases applies only for elastic collisions but it can be extended to take account for the effect of inelastic collisions.

Since the assumptions made by the rigorous kinetic theory include the properties for an ideal gas the equation of state is given by

$$pV = Nk_BT = nN_Ak_BT = nRT \tag{3.1}$$

where p is the pressure, V is the volume, N is the number of molecules, k_B the Boltzmann constant, T the temperature, n the number of moles, N_A is the Avogadro constant and R is the universal gas constant.

3.1.1. The Boltzmann Equation

The Boltzmann equation given by [26] or [25][1] is used as a starting point:

$$\frac{\partial f_k}{\partial t} + \vec{v}_k \cdot \vec{\nabla} f_k + \frac{1}{m_k}\left(\vec{F}_k \cdot \frac{\partial f_k}{\partial \vec{v}_k}\right) = \sum_j \left[\Gamma_{kj}^{(+)} - \Gamma_{kj}^{(-)}\right] \tag{3.2}$$

As shown in figure 3.1, equation (3.2) describes the time evolution of the velocity distribution function $f_k(\vec{x}, \vec{v}_k, t)$. The quantity $f_k(\vec{x}, \vec{v}_k, t)\mathrm{d}\vec{x}\mathrm{d}\vec{v}_k$ is the mean number of molecules of species k at time t, which are located in the volume element $\mathrm{d}\vec{x}$ around $\vec{x}$, and which have velocities within the range $\mathrm{d}\vec{v}_k$ around $\vec{v}_k$. The population of molecules propagating to a different position in phase space after a time $\mathrm{d}t$ is increased by some collisions $\Gamma_{kj}^{(+)}$ and decreased by others $\Gamma_{kj}^{(+)}$. The Γ_{kj} involve the intermolecular potential energy function, e.g. the Lennard-Jones-Potential, and all details of the collision trajectories [19]. The quantity $\vec{F}_k$ is an external force acting on a molecule of species k.

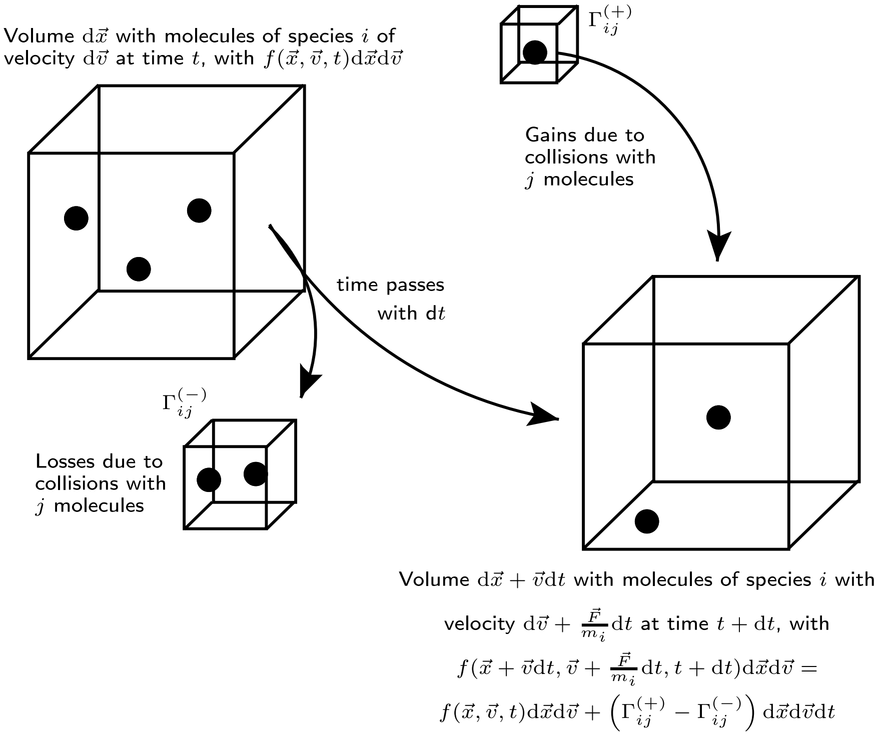
Volume $\mathrm{d}\vec{x}$ with molecules of species i of velocity $\mathrm{d}\vec{v}$ at time t, with $f(\vec{x}, \vec{v}, t)\mathrm{d}\vec{x}\mathrm{d}\vec{v}$
$\Gamma_{ij}^{(+)}$
Gains due to collisions with j molecules
time passes with $\mathrm{d}t$
$\Gamma_{ij}^{(-)}$
Losses due to collisions with j molecules
Volume $\mathrm{d}\vec{x} + \vec{v}\mathrm{d}t$ with molecules of species i with velocity $\mathrm{d}\vec{v} + \frac{\vec{F}}{m_i}\mathrm{d}t$ at time $t + \mathrm{d}t$, with
$f(\vec{x} + \vec{v}\mathrm{d}t, \vec{v} + \frac{\vec{F}}{m_i}\mathrm{d}t, t + \mathrm{d}t)\mathrm{d}\vec{x}\mathrm{d}\vec{v} =$
$f(\vec{x}, \vec{v}, t)\mathrm{d}\vec{x}\mathrm{d}\vec{v} + \left(\Gamma_{ij}^{(+)} - \Gamma_{ij}^{(-)}\right)\mathrm{d}\vec{x}\mathrm{d}\vec{v}\mathrm{d}t$

Fig. 3.1.: Graphical expression of the Boltzmann equation.

The Boltzmann equation can be viewed as a transport equation in the six-dimensional position-velocity space, where the right hand side of equation (3.2) serves as a source term. In cases where the function f can be calculated all properties of a gas mixture, like its local temperature of the chemical properties are completely known [19].

[1] Different from [25] from now vector-tensor notation is used, e.g. [19]

In the following subsections, the Boltzmann equation is used to derive the transport equations for mass, momentum and energy, as well as formulations for the molecular fluxes and the transport properties.

3.1.2. Enskog's General Transport Equation

A general transport equation for a physical quantity can be derived from the Boltzmann equation without actually determining the form of the distribution functions f_k. By multiplying equation (3.2) with the quantity ψ_k of species k and by integrating over $\vec{v}_k$ one obtains Enskog's general transport equation for the physical quantity ψ_k associated with the i-th kind of molecule [25]

$$\begin{aligned}\frac{\partial(n_k\overline{\psi}_k)}{\partial t} + \vec{\nabla}\cdot(n_k\overline{\psi_k\vec{v}_k}) - n_k\left\{\overline{\frac{\partial\psi_k}{\partial t}} + \left(\overline{\vec{v}_k\cdot\vec{\nabla}\psi_k}\right) + \left(\frac{\vec{F}_k}{m_k}\cdot\overline{\frac{\partial\psi_k}{\partial\vec{v}_k}}\right)\right\} = \\ \int\psi_k\sum_j\left[\Gamma^{(+)}_{kj} - \Gamma^{(-)}_{kj}\right]\mathrm{d}\vec{v}_k\ ,\end{aligned} \tag{3.3}$$

where n_k is the number of moles of species k and where the overbars indicate averaged quantities. A summation of equation (3.3) over all species k gives the transport equation for quantity ψ of the entire gas mixture.

For conserved quantities, such as momentum ($\psi = \sum_k m_k\vec{v}_k$), kinetic plus internal energy ($\psi = \sum_k \frac{1}{2}m_k v_k^2 + e_k^{(int)}$) or the global mass ($\psi = \sum_k m_k$) it is explicitly shown in [25], that the right hand side of equation (3.3) vanishes as it should. Note that, however, that the mass of one species k is not conserved in case of a chemical reaction: By choosing $\psi_k = m_k$ the sum of the right hand side of equation (3.3) represents the conversion rate of molecules of species k which is in general not zero.

In section 3.2 Enskog's general transport equation (3.3) is used to derive explicit formulations for the transport equations of mass, momentum and energy.

3.1.3. Chapman-Enskog Theory

To obtain rigorous expressions for the molecular fluxes and the corresponding transport coefficients a solution to the Boltzmann equation (3.2) has to be found. Here, a practical solution given by Enskog is presented [26]. By expanding the velocity distribution function f_k with different orders of approximation $f_k^{[r]}$ one obtains

$$f_k = f_k^{[0]} + \xi f_k^{[1]} + \xi^2 f_k^{[2]} + \cdots \xi^r f_k^{[r]};\ , \tag{3.4}$$

with ξ as an ordering parameter. One can find an analytical solution for the 0-th approximation of f_k [25]. Now the first-order approximation to f_k can be written in terms of a perturbation function θ_k as

$$f_k^{[1]} = f_k^{[0]}\theta_k\ , \tag{3.5}$$

and substitute $f_k^{[1]}$ in equation (3.2) to obtain an integro-differential equation for θ. As shown by [25], θ has a specific form with two unknown scalar functions [1]. By expanding the unknown scalar functions in finite series of Sonine polynomials, one can use the results to obtain first-order approximations of the molecular fluxes and the transport properties [25]. Approximations of higher order are not worthwhile since the calculation is very time consuming and the additionally gained accuracy is small [16].

3.2. Transport Equations for Chemically Reacting Flows

For numerical simulation of chemically reacting flows the transport equations constitute the mathematical and physical core of the simulation. Here, the transport equations are presented in a different way from the traditional continuum mechanics approach, e.g. [19]. Following [25], it is shown, that, based on the Boltzmann equation respectively Enskog's general transport equation presented before, the equations of mass, momentum and energy are direct consequences of the conservation laws for mass, momentum and energy.

Generally, a transport equation can be written as

$$\underbrace{\frac{\partial}{\partial t}(\rho\phi)}_{\text{Transient term}} = - \underbrace{\vec{\nabla}\cdot(\rho\vec{v}\phi)}_{\text{Convection term}} - \underbrace{\vec{\nabla}\cdot(\Lambda_\phi\vec{\nabla}\phi)}_{\text{Diffusion term}} + \underbrace{Q_\phi}_{\text{Source term}} \quad , \tag{3.6}$$

with the conserved quantity ϕ and transport coefficient Λ_ϕ. The transient term accounts for the accumulation of ϕ and the source term represents any sources or sinks in the concerned control volume. The fluxes over the control volume faces are described by either the convective term due to the flow velocity field $\vec{v}$ or the diffusion term due to its gradients.

3.2.1. Equations of Mass

By choosing $\psi_k = m_k$ in equation (3.3) one obtains, after some intermediate transformations, the transport equation of the mass fraction for each species k in a mixture of K species [25]

$$\frac{\partial}{\partial t}(\rho Y_k) = -\vec{\nabla}\cdot(\rho Y_k\vec{v}) - \vec{\nabla}\cdot\vec{j}_k + \dot{r}_k \qquad k = 1,2,3,\cdots,K \ , \tag{3.7}$$

where $Y_k = m_k/m$ is the mass fraction of k-th species, ρ is the mixture density, $\vec{v}$ is the velocity, $\dot{r}_k$ is the reaction rate of species k (see section 3.4) and $\vec{j}_k$ is the molecular diffusion flux (see section 3.3).

Summing equation (3.7) over all Y_k and using the relations

$$\sum_{k=1}^{N} Y_k = 1, \quad \sum_{k=1}^{N} \vec{j}_k = 0, \quad \sum_{k=1}^{N} \dot{r}_k = 0 \tag{3.8}$$

one obtains the transport equation for the total mass of a multicomponent reacting mixture [19, 25]

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{v}) \ . \tag{3.9}$$

3.2.2. Equation of Momentum

The momentum of a system of two colliding molecules is always conserved in any collision, even if a chemical reaction occurs. By choosing $\psi_k = m_k \vec{v}_k$ in equation (3.3) one obtains the equation of momentum with gravitation $\vec{g}$ as the only external force as [25]

$$\frac{\partial}{\partial t}(\rho \vec{v}) = -\vec{\nabla} \cdot (\rho \vec{v} \vec{v}) - \vec{\nabla} \boldsymbol{\pi} + \rho \vec{g} \ , \tag{3.10}$$

where

$$\vec{\nabla} \boldsymbol{\pi} = \vec{\nabla} p + \vec{\nabla} \cdot \vec{\tau} \tag{3.11}$$

is the rate of momentum addition by molecular transport with $\vec{\tau}$ as the molecular momentum flux vector (compare section 3.3).

3.2.3. Equation of Energy

Applying $\psi_k = \frac{1}{2} m_k v_k^2 + u_k^{(int)}$ into equation (3.3) and summing over all species k one obtains the transport equation for energy. By using the thermodynamic definition of enthalpy the transport equation for specific enthalpy is given as [1, 25]

$$\frac{\partial}{\partial t}(\rho h) = -\vec{\nabla} \cdot (\rho h \vec{v}) - \vec{\nabla} \cdot \vec{q} - \tau : \vec{\nabla} \vec{v} + \frac{\mathrm{D}p}{\mathrm{D}t} + \dot{Q}_{source} \ , \tag{3.12}$$

where $\vec{q}$ is the molecular heat flux (compare section 3.3), $\dot{Q}_{source}$ is a combination of all heat source terms like the radiative flux, $\frac{\mathrm{D}p}{\mathrm{D}t}$ is the reversible rate of enthalpy due to compression and $(\tau : \vec{\nabla} \vec{v})$ is the irreversible rate of enthalpy due to viscous dissipation.

Further with the sensible enthalpy h_s and the chemical enthalpy h_c the total enthalpy h can be expressed as [21]

$$h = h_s + h_c = \underbrace{\int_{T_0}^{T} c_p(T)\, \mathrm{d}T}_{\text{sensible enthalpy}} \quad + \quad \underbrace{\sum_{k=1}^{N} \Delta h_{f,k}^0 Y_k}_{\text{chemical enthalpy}} \ , \tag{3.13}$$

with the heat capacity for an ideal gas at constant pressure [24]

$$c_p(T) = \left(\frac{\partial h}{\partial T} \right)_p \ . \tag{3.14}$$

Neglecting viscous dissipation and heat source terms like the radiative flux, one obtains the equation of energy in terms of sensible enthalpy to

$$\frac{\partial}{\partial t}(\rho h_s) = -\vec{\nabla} \cdot (\rho h_s \vec{v}) - \vec{\nabla} \cdot \vec{q} + \frac{\mathrm{D}p}{\mathrm{D}t} + \vec{q}_{reaction} \ . \tag{3.15}$$

The separation of sensible enthalpy from the heat of formations gives this equation the advantage over equation (3.12), that the heat of formations become the only source term

$$\vec{q}_{reaction} = -\sum_{k=1}^{N} \Delta h^0_{f,k} \dot{r}_k \ . \tag{3.16}$$

Additionally, this equation can be used as a definition for the heat release rate.

To obtain the temperature from the sensible enthalpy, the definition of the sensible enthalpy (3.13) is solved for T. In general, the mixture averaged specific heat $c_P(T) = \sum_{k=1}^{K} c_{p,k}(T) Y_k$ is a function of T and therefore, the integral in (3.13) has to be solved iteratively [21]. The temperature dependencies of the pure species specific heat capacities $c_{p,k}$ are fitted by 4-th order NASA polynomials:

$$\frac{c_{p,k}(T)\bar{M}_k}{R} = a_{1,k} + a_{2,k}T + a_{3,k}T^2 + a_{4,k}T^3 + a_{5,k}T^4 \tag{3.17}$$

Here, $a_{n,k}$ are coefficients of the k-th species which has to be given as input.

3.3. Multicomponent Molecular Transport

Gradients exist in a gas under non-equilibrium conditions in one or more physical quantities: composition, mass averaged velocity and temperature. These gradients cause the molecular or diffusive transport of mass, momentum and energy through the gas.

In this section, expressions for the diffusive flux vectors in the transport equations (3.7), (3.10) and (3.15) will be derived based on the Chapman-Enskog formulation in subsection 3.1.3. In addition, a formulation for multicomponent transport coefficients laid out by [16] for computational purpose will be given.

3.3.1. Species Mass Flux

Inserting the first approximation of the distribution function in terms of the perturbation function mentioned in subsection 3.1.3 and making use of the polynomial expansions and orthogonality relations [25] one obtains the mass diffusion velocity [1, 16]

$$\vec{V}_k = \frac{1}{X_k \overline{M}} \sum_{j=1}^{N} \overline{M}_j D_{kj} \vec{d}_j - \frac{D_k^T}{\rho Y_k} \frac{1}{T} \vec{\nabla} T \ , \tag{3.18}$$

where $X_k = n_k/n$ is the molar fraction of species k, $\overline{M}$ is the mean molecular weight, $\overline{M}_j$ is the molecular weight of species j, D_{kj} is the ordinary diffusion coefficient of species k through species j, D_k^T is the thermal diffusion coefficient of species k and the driving force $\vec{d}_j$ is defined as

$$\vec{d}_j = \vec{\nabla} X_j + (X_j - Y_j) \frac{1}{p} \vec{\nabla} p \ . \tag{3.19}$$

Finally, the multicomponent mass flux in equation (3.7) is given by

$$\vec{j}_k = \rho Y_k \vec{V}_k = \frac{\rho Y_k}{X_k \overline{M}} \sum_{j=1}^{N} \overline{M}_j D_{kj} \vec{d}_j - D_k^T \frac{1}{T} \vec{\nabla} T \; . \tag{3.20}$$

Equation (3.20) shows that the mass flux can be caused by three different phenomena. The ordinary diffusion flux due to a gradient in concentration, the flux due to a pressure gradient and the thermal diffusion flux due to a temperature gradient. The flux due to a pressure gradient is very small compared to other effects and can be neglected [25]. Additionally, a flux due to external forces can occur for example in charged mixtures.

Mass Diffusion Coefficients

Based on the theory provided by [25] and laid out for computational purpose by [16], the multicomponent diffusion coefficients D_{kj} and the multicomponent thermal diffusion coefficients D_k^T in equation (3.20) are computed from a system of equations defined as (L)-matrix which consists of nine sub-matrices:

$$\begin{pmatrix} L^{00,00} & L^{00,10} & 0 \\ L^{10,00} & L^{10,10} & L^{10,01} \\ 0 & L^{01,10} & L^{01,01} \end{pmatrix} \begin{pmatrix} a_{00}^1 \\ a_{10}^1 \\ a_{01}^1 \end{pmatrix} = \begin{pmatrix} 0 \\ X \\ X \end{pmatrix} \tag{3.21}$$

with the right hand side vector composed by the mole fraction vectors X_k.

Every component of the L matrix is a $K \times K$, with total number of species K, matrix. For example the elements of the matrix $L^{00,00}$ are calculated with

$$L_{jk}^{00,00} = \frac{16T}{25p} \sum_{l=1}^{N} \frac{X_l}{\overline{M}_j \mathcal{D}_{jl}} [\overline{M}_k X_k (1 - \delta_{jl} - \overline{M}_j X_j (\delta_{jk} - \delta_{kl})] \; , \tag{3.22}$$

where δ is a small value due to prohibit species concentration values of exactly zero and $\mathcal{D}_{jk}$ is the binary diffusion coefficients calculated from Chapman-Enskog's theory with

$$\mathcal{D}_{jk} = \frac{3}{16} \frac{\sqrt{2\pi k_B^3 T^3 / m_{jk}}}{p \pi \sigma_{jk}^2 \Omega_{jk}^{(1,1)\star}} \; . \tag{3.23}$$

Here k_B is the Boltzmann constant, $m_{jk} = \frac{m_k m_j}{m_k + m_j}$ with m_k the mass of a molecule k, is the reduced molecular mass, σ_{jk} is the collision diameter and $\Omega_{jk}^{(1,1)\star}$ is the collision integral. The collision diameter and the collision integral can be found in transport data files, e.g GRI-Mech 1.2 [27].

With the inverse (P) of the $L^{00,00}$-block the first-order approximation of the multicomponent diffusion coefficients is given by

$$D_{jk} = X_j \frac{16T}{25p\overline{M}_k} \overline{M} (P_{jk} - P_{jj}) \; , \tag{3.24}$$

and the thermal diffusion coefficients are given by

$$D_k^T = \frac{8\overline{M}_k X_k}{5R} a_{k00}^1 \ . \tag{3.25}$$

Here, R is the universal gas constant and a_{k00}^1 is calculated from the (L)-matrix. Equation (3.24) shows two characteristics of the multicomponent diffusion matrix predicted by [25],

$$D_{jj} = 0 \quad \text{and} \quad D_{jk} \neq D_{kj} \ . \tag{3.26}$$

For further details on the (L)-matrix one should consult [1, 16].

3.3.2. Momentum Flux

The laminar momentum flux $\vec{\tau}$ for a Newtonian fluid is given by [19]:

$$\vec{\tau} = -\mu[\vec{\nabla}\vec{v} + (\vec{\nabla}\vec{v})^\dagger] + \frac{2}{3}\mu(\vec{\nabla}\cdot\vec{v})\delta \tag{3.27}$$

Here, μ is the mixture viscosity and δ is the identity matrix. Note, that the momentum flux is not derived from expressions obtained from the rigorous kinetic theory of gases, but from hydrodynamic equations [25].

The viscosity of a mixture μ is calculated from Wilke's mixture rule [28] modified by [19]

$$\mu = \sum_{k=1}^{K} \frac{X_k \mu_k}{\sum_{j=1}^{K} X_j \Phi_{kj}} \ , \tag{3.28}$$

with

$$\Phi_{kj} = \frac{1}{\sqrt{8}}\left(1 + \frac{\overline{M}_k}{\overline{M}_j}\right)^{-1/2}\left(1 + \left(\frac{\mu_k}{\mu_j}\right)^{1/2}\left(\frac{\overline{M}_j}{\overline{M}_k}\right)^{1/4}\right)^2 \ , \tag{3.29}$$

and the pure species viscosity derived from the rigorous kinetic theory of gases

$$\mu_k = \frac{5}{16}\frac{\sqrt{\pi m_k k_B T}}{\pi\sigma_k^2 \Omega_{kk}^{(2,2)\star}} \ . \tag{3.30}$$

k_B is the Boltzmann constant, σ_k is the collision diameter for the k-k interaction potential, m_k is the mass of molecule k and $\Omega_{kk}^{(2,2)\star}$ the reduced collision integral.

It has been shown by [25] that Wilke's mixture rule is a very good approximation to the equation derived from the rigorous kinetic theory. One should note, that the pure species viscosities calculated with equation (3.30) are still based on the rigorous kinetic theory.

3.3.3. Heat Flux

By using the rigorous kinetic theory of gases for the energy flux $\vec{q}$ in a multicomponent gas mixture, the heat flux vector is given by [16] to

$$\vec{q} = -\lambda_0 \vec{\nabla} T + \sum_{k=1}^{N} \vec{j}_k h_k - \sum_{k=1}^{N} \frac{RT}{\overline{M_k} X_k} D_k^T \vec{d}_k \ , \tag{3.31}$$

where λ_0 is the special mixture thermal conductivity, $\vec{j}_k$ is the diffusive mass flux discussed before, h_k is the specific enthalpy of species k and d_k is the diffusive driving force see equation (3.19)

As one can see from equation (3.31) heat can be transported through a multicomponent gas caused by a temperature gradient known as Fourier's law, a diffusive flux by species, and the reciprocal process to thermal diffusion called Dufour effect [25]. Like the ordinary and thermal diffusion coefficients, the thermal conductivity λ_0 is calculated with a^1_{i10} and a^1_{i01} from the L-matrix (3.21) with

$$\lambda_0 = \lambda_{0,trans.} + \lambda_{0,inter.} = -4 \sum_k X_k (a^1_{i10} + a^1_{i01}) \tag{3.32}$$

It consists of a translational and a internal part due to the molecular vibration.

3.4. Reaction Kinetics

Dependent on physical conditions and used species a chemical reaction is proceed by a series of elementary reactions, called the reaction mechanism. Generally, each elementary reaction can be described as

$$\sum_{k=1}^{K} \nu_{ki}{}' \mathcal{M}_k \rightleftharpoons \sum_{k=1}^{K} \nu_{ki}{}'' \mathcal{M}_k \quad (i = 1, \cdots, I) \ , \tag{3.33}$$

where K is the total number of chemical species, I is the total number of chemical reactions, $\mathcal{M}_k$ is the name of species k and ν_{ki} is the stoichiometric coefficient of species k in the forward direction respectively reverse direction in the i-th reaction.

To obtain the rate of progress ω_i of a two-body reaction i the reaction rates of for- and backward reaction have to be combined to [1]

$$\omega_i = k_{f,i} \prod_{k=1}^{K} [\mathcal{M}_k]^{\nu_{ki}{}'} - k_{r,i} \prod_{k=1}^{K} [\mathcal{M}_k]^{\nu_{ki}{}''} \ , \tag{3.34}$$

where $k_{f,i}$ and $k_{r,i}$ are the rate constants for the forward and reverse direction of reaction i and $[\mathcal{M}_k]$ is the concentration of species k.

The rate constant k is computed by a modified three-parameter Arrhenius form

$$k = A^\star \, T^\beta \exp(-E_a/RT) \ , \tag{3.35}$$

where $A^{\star}$ is the pre-exponential constant, β is the temperature exponent and E_a is the activation energy. Obviously, the rate constant depends strongly on the temperature.

Equation (3.34) together with the reaction equations (3.33) and the rate constant equations (3.35) build the chemical kinetic model. This system of ordinary differential equations have to be solved numerically [29]. The parameters needed for chemical kinetic models are listed in data files like in the appendix B.

Finally, the chemical source for species $\dot{r}_k$ is given by

$$\dot{r}_k = \sum_{i}^{I} (\nu_{ki}'' - \nu_{ki}')\omega_i \ . \tag{3.36}$$

3.5. Chemical Time Scales

The needed time step in a numerical simulation is determined by the smallest time scale on which the regarded physical and chemical phenomena occur [3, 21]. Therefore, knowledge about the magnitudes of chemical time scales is important and offers the opportunity to adjust the time step.

Because of the many approaches a definition by [30] will be used to calculate chemical time scales:

$$\tau_\phi(x) = \phi(x) \left(\frac{\partial \phi(x)}{\partial t} \right)^{-1} \ . \tag{3.37}$$

Here, the time scale $\tau(x)$ is determined with the variation of a characteristic quantity ϕ on a specific location with time t.

For highly unsteady chemical reactions like a moving flame front and negative mass fraction gradients, this equation can be further converted with

$$\frac{\partial \phi}{\partial t} = \frac{\partial \phi}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial \phi}{\partial x} \ v \ , \tag{3.38}$$

into a more suitable equation,

$$\tau_\phi(x) = \phi(x) \left(\left| \frac{\partial \phi(x)}{\partial x} \right| \cdot v \right)^{-1} \ . \tag{3.39}$$

This equation describes the variation of a quantity ϕ with location x multiplied by the velocity v. One should note, that equation (3.39) is mathematically not clearly defined for mass fraction gradients equal zero.

For turbulent premix flames it is assumed, that the chemical time scale is not affected by turbulence as long as the smallest eddies can not penetrate into the thin reaction zone [4].

4. Numerical Solution of Partial Differential Equations

The transport equations in section 3.2, together with the expressions for the fluxes and transport properties in section 3.3 and the expressions for the chemical sources in section 3.4, discussed in the chapter before, form a complicated system of partial differential equations. Due to the demand on accuracy one can not include simplifications to obtain an analytic solution but has to approximate the solution numerical.

The components of a numerical solution method include a physio-mathematical model, which was introduced in the sections before, a discretization method, a numerical grid and a solution method [7].

A spatial discretization method has to be used to approximate the position dependent parts of the partial differential equations by a system of algebraic equations at discrete locations in space. There are many approaches like finite difference, finite element or finite volume methods. Due to its importance in computational fluid dynamics and its implementation in OpenFOAM® only the finite volume method (FVM) is presented following [7, 31]. Additionally, in unsteady problems a discretization of the time dependent parts of the partial differential equations has to be used.

4.1. The Finite Volume Method

In the FVM the solution domain is first subdivided into a finite number of contiguous control volumes. Then the conservation equations are applied in integral form to each control volume and volume integrals are transformed into surface integrals by using Gauss's Theorem. After, the integrals are approximated using suitable quadrature formulas. Since the variable values are calculated only at the centroid of each control volume, interpolation is used to express variable values at the control volume surfaces. As a result, one obtains an algebraic equation for each control volume. Finally, the resulting matrix is solved directly or iteratively.

As mentioned above, by integrating the stationary part of the generic transport equation (3.6) over a volume and using Gauss's Theorem to transform volume integrals containing a divergence term to surface integrals, a stationary generic conservation equation for a quantity ϕ in integral form is obtained to

$$\int_S \rho\phi\vec{v}\cdot\, \mathrm{d}\vec{S} = \int_S (\Lambda\vec{\nabla}\phi)\cdot\, \mathrm{d}\vec{S} + \int_V q_\phi \,\mathrm{d}V \ , \tag{4.1}$$

where S is a surface, V is a volume, Λ is a transport coefficient, q_ϕ is a source term and ϕ is some physical quantity. The used parameters in FVM are described in figure 4.1.

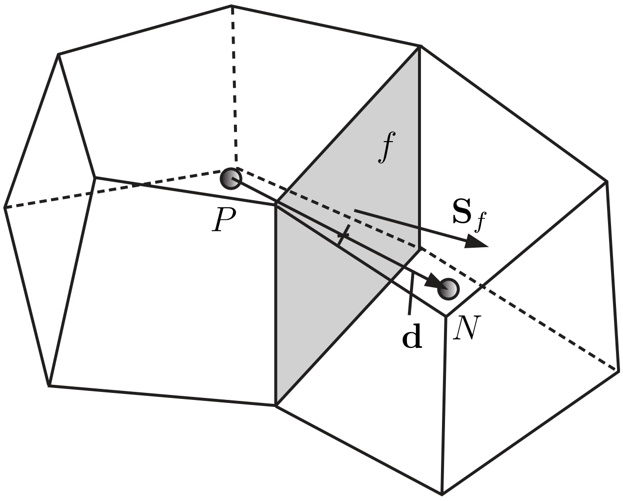

Fig. 4.1.: Two control volumes with centers P and N are connected through face f with face normal $\vec{S}_f$. Figure taken from [31].

Here, P and N are the control volume centers, f is the connecting cell face, $\vec{d}$ is the distance between the cell centers and $\vec{S}_f$ is the face area vector.

Figure 4.2 shows how a structured grid divides the solution domain of a duct into a finite number of small control volumes. Equation (4.1) applies to each single control volume. A sum over all control volumes results in the global conservation equation, since surface integrals over inner control volumes cancel out. This provides a principal advantage of FVM thus global conservation is build in.

4.1.1. Approximation of Integrals

To obtain an algebraic equation for a control volume from equation (4.1), the integrals need to be approximated.

Surface Integrals

By adding the integrals over the six control volume faces together one obtains the net flux through the control volume boundary:

$$\int_S \vec{w} \cdot \, \mathrm{d}\vec{S} = \sum_k \int_{S_k} \vec{w} \, \mathrm{d}\vec{S}_k \; , \tag{4.2}$$

where w is the convective or diffusive flux vector from (4.1). Usually, values for velocity and physical quantities are taken from the last time step. To preserve conservation, control volumes are not allowed to overlap so that each control volume face is unique to the two control volumes which lie on either site. Each cell face has only one owner and

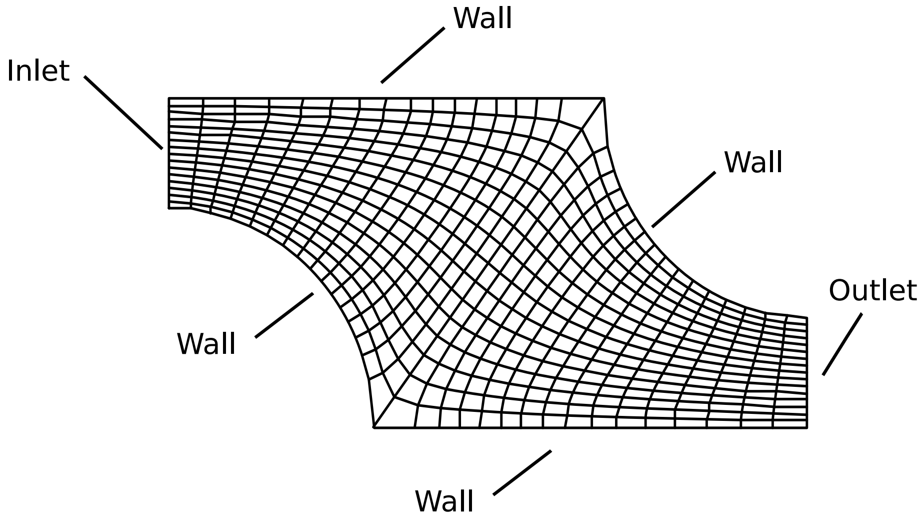

Fig. 4.2.: Example of a 2D, structured, non-orthogonal grid to simulate the flow through a duct. Figure similar to [7].

one neighbor [31].

The cell face integral $\int_{S_k}$ in equation (4.2) can not be calculated exactly, since only discrete values of f at the control volume center exist. Therefore w has to be approximated, e.g. by the midpoint rule:

$$\int_{S_f} w_f \, \mathrm{d}S_f = \bar{w}_f S_f \approx w_f S_f \tag{4.3}$$

Here, the integral is approximated as a product of the mean value over the surface $\bar{w}_f$, which has to be approximated itself, and the cell-face area S_f. To obtain higher approximations, the flux has to be calculated at more than two locations.

Volume Integrals

The last term in equation (4.1) requires integration over the volume. The simplest second-order approximation can be obtained by replacing the volume integral by the product of the mean value of the integrand, which is approximated as the value at the center of the control volume:

$$\int_V q_\phi \, \mathrm{d}V = \bar{q}\Delta V \approx q_P \Delta V \ , \tag{4.4}$$

where q_P is the value of q at the center of the control volume. To obtain approximation of higher order, one has to interpolate between values of q besides the value at the center.

4.1.2. Interpolation and Differentiation Procedures

In order to obtain the approximations to the integrals, values of variables at locations other than control volume centers are needed. Since numerous possibilities are available, only a few while be further discussed.

Linear Interpolation Scheme (CDS)

The simplest second-order scheme to obtain ϕ_e at the control volume face center is the linear interpolation between the two nearest nodes,

$$\phi_f = \phi_N \left(\frac{x_f - x_P}{x_N - x_P} \right) + \phi_P \left(1 - \frac{x_f - x_P}{x_N - x_P} \right) \ , \tag{4.5}$$

A Taylor series expansion of ϕ_N about the point x_P gives

$$\phi_f = \phi_N \frac{x_f - x_P}{x_N - x_P} + \phi_P \left(1 - \frac{x_f - x_P}{x_N - x_P} \right) - \frac{(x_f - x_P)(x_N - x_f)}{2} \left(\frac{\partial^2 \phi}{\partial x^2} \right)_P + H \ , \tag{4.6}$$

where H denotes higher-order terms. One can see, that the leading truncation error is proportional to the square of the gird spacing. Therefore the scheme is second order accurate.

Upwind Interpolation Scheme (UDS)

In upwind interpolation, ϕ_f is approximated as

$$\phi_f = \begin{cases} \phi_P & \text{if} (\vec{v} \cdot \vec{S_f})_f > 0 \\ \phi_N & \text{if} (\vec{v} \cdot \vec{S_f})_f < 0 \end{cases} \tag{4.7}$$

A Taylor series expansion about P gives for a Cartesian grid and $(\vec{v} \cdot \vec{S_f})_f > 0$

$$\phi_f = \phi_P + (x_f - x_P) \left(\frac{\partial \phi}{\partial x} \right)_P + \frac{(x_f - x_P)^2}{2} \left(\frac{\partial^2 \phi}{\partial x^2} \right)_P + H \ , \tag{4.8}$$

One can see, that the UDS is of first order. Its leading truncation error term is diffusive and will therefore never yield oscillatory solutions but at the expense of accuracy. Peaks or rapid variations in the variables will be smeared out.

4.2. Methods for Unsteady Problems

Since many physical phenomenal, like turbulence, are highly unsteady, the time derivations in the transport equations are not zero. Therefore, just as the spatial derivations, the time must be discretized. The main difference between spatial and time coordinates is the direction of influence. Unsteady flows are elliptic in space but parabolic-like in time. That means, that a force at any space may influence the flow anywhere else but a force at a given instant will only affect the flow in the future.

As an example, an first order ordinary differential equation with an initial condition is considered:

$$\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = f(t, \phi(t)); \ \ \phi(t_0) = \phi^0 \tag{4.9}$$

If one finds a solution ϕ a short time Δt after the initial point, the solution at $t_1 = t_0 + \Delta t$ can be regarded as a new initial value for the next time step. Therefore the solution

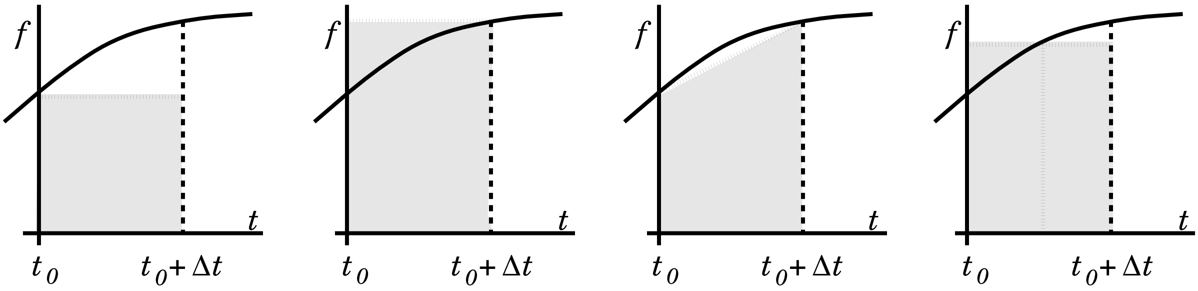

Fig. 4.3.: Approximation of the time integral $f(t)$ over an interval Δt. From left to right: explicit Euler, implicit Euler, trapezoidal rule, midpoint rule [7].

methods advance in a step-by-step or "marching" manner.

Integrating equation (4.9) from t_n to $t_{n+1} = t_n + \Delta t$ one obtains

$$\int_{t_n}^{t_{n+1}} \frac{\mathrm{d}\phi}{\mathrm{d}t}\mathrm{d}t = \phi^{n+1} - \phi^n = \int_{t_n}^{t_{n+1}} f(t,\,\phi(t))\;\mathrm{d}t\;, \tag{4.10}$$

with $\phi^{n+1} = \phi(t_{n+1})$.

To calculate ϕ at discrete values in time, an approximation of the integral on the right hand side has to be found. Figure 4.3 shows four relatively simple methods, the gray area is the approximation of the integral.

Using the value of integrand at the initial point one obtains the explicit or forward Euler method. Instead, by using the final point in the estimation one obtains the implicit or backward Euler method. By using the midpoint one obtains the midpoint rule which is the basis to the leapfrog method. Finally, by using a straight line interpolation one obtains the trapezoid rule, which is the basis to the Crank-Nicolson method. Implicit methods, which means including future times steps into the approximation, are more stable than explicit.

A necessary condition for convergence is the Courant–Friedrichs–Lewy (CFL) condition. It is defined with the Courant number Co

$$Co = \frac{\vec{v}\Delta t}{\Delta x} \leq Co_{max}\;, \tag{4.11}$$

which represents the fraction of the cell that the flow advances during a time step. Normally, the maximum Courant number is taken to $Co_{max} = 1$ [7].

4.3. Solution of Linear Equation Systems

Summing all discretized equations obtained from equation (4.1) for each control volume leads to system of algebraic equations

$$A\vec{\phi} = \vec{Q}\;, \tag{4.12}$$

where A is the matrix with the coefficients, $\vec{\phi}$ is the solution vector and $\vec{Q}$ contains the source terms.

To solve these equations, direct methods and iterative methods can be used. With direct methods, the solution is determined in one single step. The basic method is the Gaussian elimination, which divides large systems of equations systematically into smaller ones. For full matrices, it is one of the fastest and most accurate methods, but usually matrices in fluid mechanics are sparse. In addition, the rounding errors can grow rapidly and therefore the accuracy may be insufficient.

In iterative methods the solution vector $\vec{\phi}$ is corrected until a convergence criterion is satisfied. After n iterations an approximate solution $\vec{\phi}^n$ is obtained which does not satisfy the equations (4.12) exactly. Instead, there is a non-zero residual ζ^n:

$$A\vec{\phi}^n = \vec{Q} - \vec{\zeta}^n \tag{4.13}$$

To iteratively drive the residual to zero one can write

$$M\vec{\phi}^{n+1} = N\vec{\phi}^n + \vec{B}. \tag{4.14}$$

Since, at convergence, $\vec{\phi}^{n+1} = \vec{\phi}^n = \vec{\phi}$ it must be

$$P\,A = M - N \quad \text{and} \quad \vec{B} = P\,\vec{Q}\;, \tag{4.15}$$

where P is a non-singular pre-conditioning matrix. Suitable for iterative methods are the Jacobi-Iteration, the Gauss-Seidel-Iteration and relaxation methods. There are also methods which work partly directly and partly iteratively, for example, the block iteration, LU (Lower-Upper) iteration and the CG (conjugate gradient) method.

5. Implemented Solver in OpenFOAM and the Cantera Interface

This chapter starts with a short discussion of the used software packages OpenFOAM 2.0.1 and Cantera 1.8. Thereafter, the developed interconnection library between OpenFOAM and Cantera is presented. Finally, the implementation of the species and sensible enthalpy equations as well as the integration of the coupling library into existent solvers is showed.

5.1. Software Packages

OpenFOAM Version 2.0.1

OpenFOAM® [1] is an open source object-oriented software package written in C++03. It provides a framework to develop numerical solvers in continuum mechanics. Main advantages of the design are the overloaded operators, allowing expressive and versatile syntax for implementations of complex physical models, the extensive pre- and postprocessing tools including complex geometry handling and data in-/output and a wide range of implemented discretization schemes and boundary conditions. Additionally, OpenFOAM provides a good parallelization, a high accuracy and is available free of charge. It has been validated many times and is widely used. Fundamental developments are done by the OpenFOAM Foundation with contributions from a large community. The complete source code is licensed under the GNU General Public License (GPL) for free use and customization [15].

In OpenFOAM the discretization of the solution domain is accomplished by the finite volume method (FVM) presented before. For the discretization of mathematical equations OpenFOAM provides two basic classes. Implicit discretization is handled by the class *fvm*. Using the different overloaded operators, e.g. laplacian or grad, of the class *fvm*, adds directly to the solving matrix. Explicit discretization is accomplished by using the class *fvc*. Operators from *fvc* add a source term to the solving matrix. The user has the possibility to chose during runtime from a wide range of discretization schemes which differ in accuracy, stability and error order (e.g. linear Gauss schemes in second or fourth order for gradients, upwind or cubic divergence schemes, Crank-Nicholson for the time and many more). OpenFOAM offers many solvers for specific cases like compressible, multiphase or reacting flows. Additionally, one can chose during runtime which turbulence, combustion or thermodynamic model the solver should use [32].

[1] Open Source Field Operation and Manipulation, registered trademark of OpenCFD Limited

OpenFOAM provides a framework for parallelization via domain decomposition using different schemes or third-party applications, e.g. Scotch, and the possibility to include different implementations of the Message Passing Interface (MPI), e.g. Open MPI or MPICH2. The solving matrices can be solved using different strategies, e.g. the linear preconditioned conjugate gradient (PCG) solver for symmetric matrices or the linear preconditioned biconjugated gradient (PBiCG) solver for asymmetric matrices where the pressure-velocity coupling can be solved using a semi-implicit method for pressure-linked equations (SIMPLE) algorithm [7, 32].

Cantera Version 1.8

Cantera from the California Institute of Technology is an object oriented software toolkit for problems involving chemical kinetics, thermodynamics and transport processes [17]. It consists of a kernel written in C++03 which is accessible through a wide range of environments like MATLAB and C++. Cantera provides fast, efficient algorithms and is highly customizable [33]. The complete source code is licensed under the Berkeley Software Distribution 3-Clause License (BSD-3) which has been verified as a GPL-compatible free software license [34].

In simulation codes for turbulent, reacting flows, like OpenFOAM, Cantera can be used to evaluate thermodynamic properties, chemical sources or transport coefficients that appear in the transport equations. Cantera includes highly accurate equations derived from the rigorous kinetic theory of gases presented in chapter 3. The implementation is accomplished following [16]. Due to the mathematically expensive operations on very stiff ordinary differential equation systems and matrices inversion, Cantera makes use of highly specialized software packages like SUNDIALS[2], BLAS[3] and LAPACK[4]. To simulate complex systems, a number of ideal reactors is available [33]. Cantera can read thermophysical, transport and chemical data from data files in the CHEMKIN format, see [35]. Additionally, it is possible to perform sensitivity analysis.

5.2. Connection of OpenFOAM with Cantera

Since the numerical computation of multicomponent transport coefficients is not trivial, see [1, 16], a new implementation into OpenFOAM is not worthwhile. As mentioned in the section before, Cantera is already capable of calculating accurate multicomponent transport coefficients needed in the transport equations, see chapter 3. Additionally, Cantera can read transport data files in the CHEMKIN format and the calculation of chemical reaction rates is faster as in OpenFOAM [18], which is an important factor while performing time-consuming direct numerical simulations. Both OpenFOAM and Cantera use C++ as the main language. Hence, it is possible to include source code from Cantera directly into OpenFOAM-classes respectively use Cantera functions in OpenFOAM. Based on the *Alternate Chemistry Library* for OpenFOAM 1.5 (see [18])

[2] Suite of nonlinear and differential/algebraic equation Solvers (SUNDIALS)
[3] Basic Linear Algebra Subprograms (BLAS)
[4] Linear algebra package (LAPACK)

the coupling interface between OpenFOAM and Cantera was rewritten to meet the requirements of OpenFOAM in Version 2.0.1.

The interface makes use of templates to work with existent solvers through the run-time selection mechanism. The extensive use of OpenFOAM respectively C++ programming principles enables easy development of further extensions. Additionally, the coupling offers access to sensible enthalpies, multicomponent or mixture-averaged transport properties or chemical reaction rates. The input files containing molecular properties and chemical mechanism data are in the classical CHEMKIN compatible format.

5.2.1. Structure of the Interface

As shown in figure 5.1, the new coupling interface consists of several classes. The structure is similar to OpenFOAM's with numerous classes derived from original source files. The classes *canteraChemistryModel* and *canteraHsPsiMixtureThermo* offer access to cell dependent values like the chemical source terms, the enthalpy source term or the transport properties. Therefore they are derived from the class *canteraMixture*, which sets up the physical conditions in Cantera, like temperature, pressure and concentration. It therefore contains an object of the class *canteraThermo*, which calculates cell dependent thermophysical properties through Cantera while making use of an object of the class *canteraGasMixWrapper*. The class *canteraGasMixWrapper* deploys the coupling with original classes from Cantera. It is derived from Cantera's class *IdealGasMix* and contains an object of Cantera's class *Multitransport*.

To guarantee full compatibility to existent reacting flow solvers, the class *canteraHsPsiMixtureThermo* is derived from the original class *hsCombustionThermo* and the class *canteraMixture* is derived from the original class *basicMulticomponentMixture*. Consequently one can perform downcasts[5] in existent solvers to obtain access to functions which are newly introduced through classes of the interface, e.g. mass diffusion coefficients.

5.2.2. Integration of the Interface into OpenFOAM

Since the interface offers access to nearly all functions known from the original OpenFOAM classes, one can select during runtime which library calculates physical properties like sensible enthalpy or viscosity. This is realized with dictionary files in the case directory through OpenFOAM's runtime-selection mechanism. Note, that in general no changes on solver's source code is needed. However, like already mentioned before, OpenFOAM's original classes do not offer mass diffusion coefficients. Hence, to implement transport equations like in the section before, the source code of the solver has to be changed more deeply.

[5] Type casting is converting an expression of a given type into another type. Downcasting is converting a bass-class reference or pointer to a derived-class. The opposite process is called upcasting [36, 37].

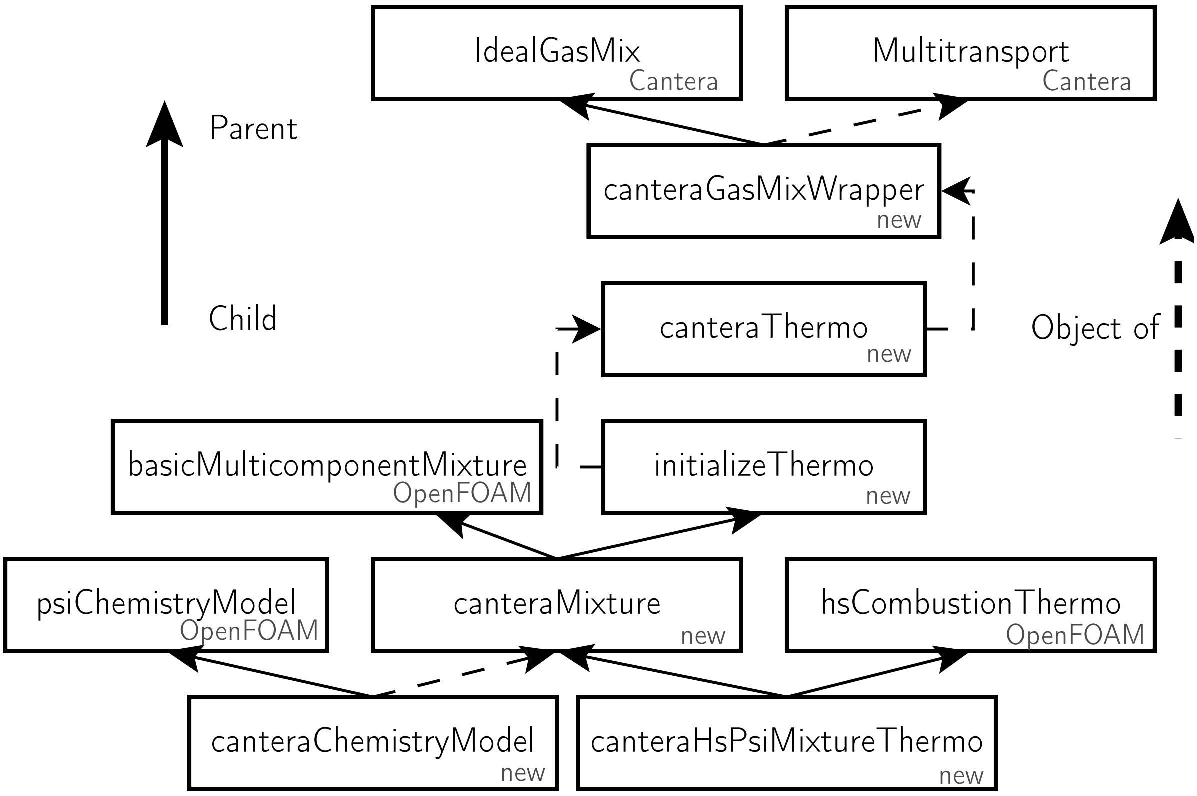

Fig. 5.1.: Inheritance and dependency diagram for the coupling library. Dotted arrow "contains an object of", normal arrow "derived from".

Figure 5.2 shows the initialization sequence of the coupling library in any solver during runtime. The order in which the class constructors are called indicates the high level of interconnection between original classes and classes from the interface. It is important that the start of the initialization is performed by the original function *psiChemistryModelNew*. This function is already implemented in existent solvers and guarantees the connection to OpenFOAM's runtime selection mechanism. It reads input files and chooses the chemistry class, here the *canteraChemistryModel*. The function *hsCombustionThermoNewType* chooses the thermo class from input files. Here, the class *canteraHsPsiMixtureThermo* is used, which calls the subclass constructors until finally the class *canteraGasMixWrapper* deploys the connection to Cantera's original classes. Listing A.3 in the appendix shows the implementation of the initialization.

5.2.3. Exchange of Data between OpenFOAM and Cantera

To obtain access to derived class properties, like the mass diffusion coefficients, a downcast shown in listing A.3 in the appendix, has to be performed. Figure 5.3 shows the exchange of data between OpenFOAM and Cantera through the interface *canteraFoamModel*. OpenFOAM solves the implemented transport equations for energy, species, velocity and pressure, and passes the thermodynamic data in vectors of dimension number of control volumes N to the interface. The interface then constructs for each single control volume an independent reactor. For every reactor Cantera calculates cell depen-

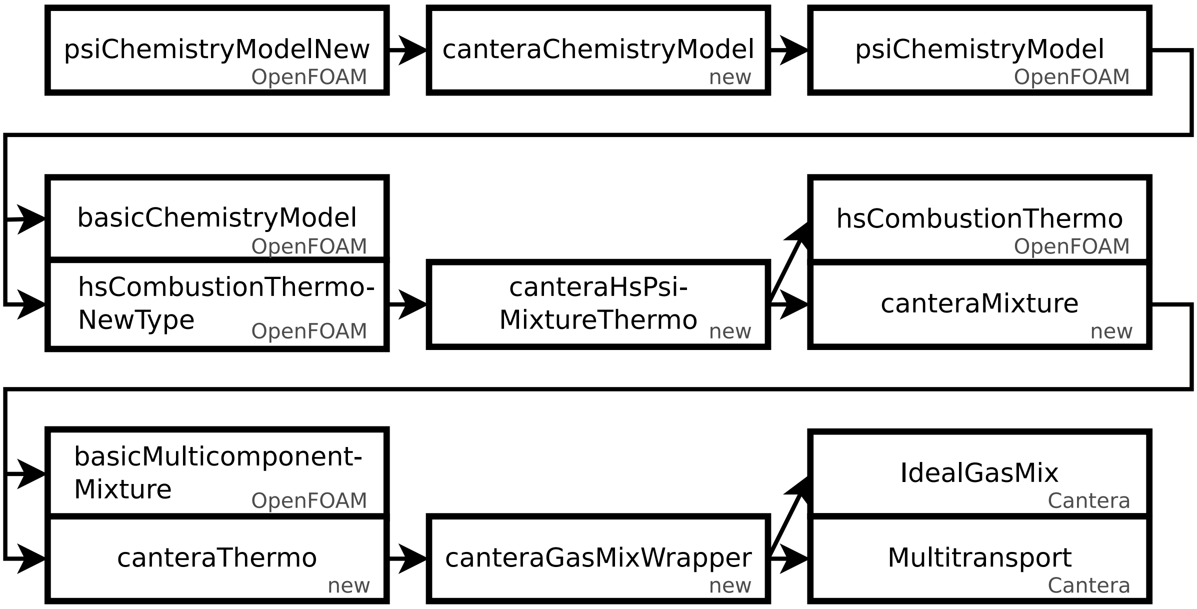

Fig. 5.2.: Initialization sequence of the coupling library in a solver during runtime (order: top left first, bottom right last).

dent data like the chemical reaction rates $\dot{r}_k$ or transport properties like $\mu, \lambda, D_{kj}, D_k^T$. Cantera then returns per reactor vectors of dimension number of species K for $\dot{r}_k$ or D_k^T, simple scalars for μ or λ, and matrices of dimension $(K \times K)$ for D_{kj}. The interface collects the values of every cell and constructs vectors or matrices with cell values for every control volume and returns matrices of dimension $(N \times K)$ for the reaction rates or of dimension $(N \times K \times K)$ for the ordinary diffusion coefficients D_{kj} and vectors of dimension (N) for μ or κ.

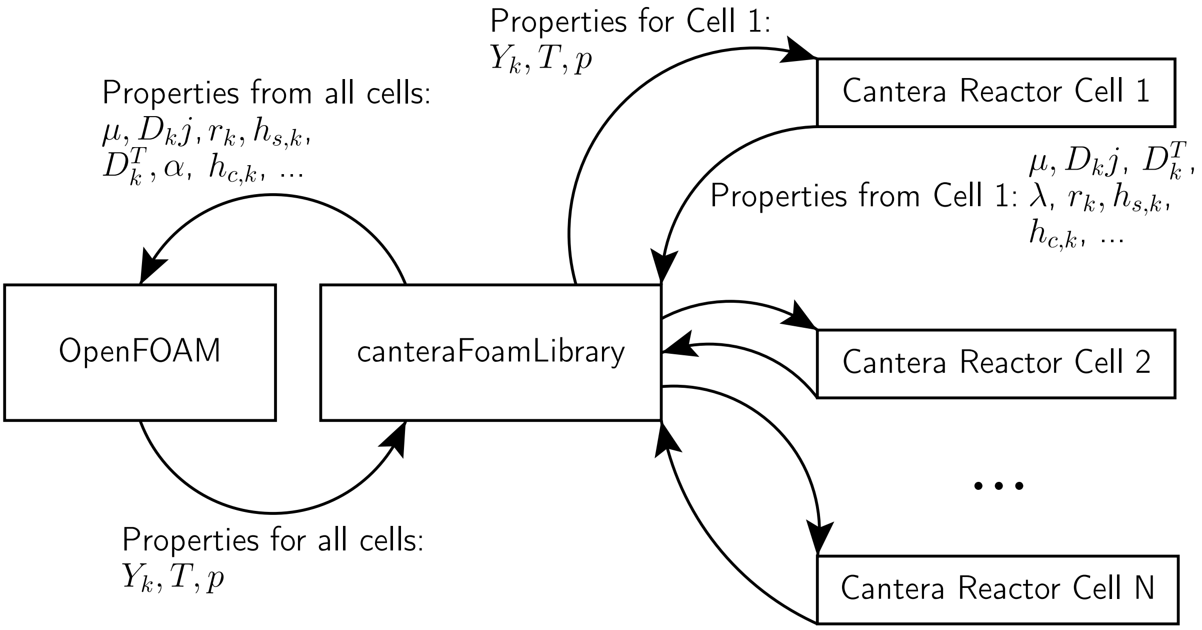

Fig. 5.3.: Data exchange between OpenFOAM and Cantera through the interface *canteraFoamModel*.

5.3. Implementation of the Solver

The new solver is based on the OpenFOAM solver *reactingFoam*. Transport equations for species mass and energy based on equation (3.7) and (3.15) replace the approach in the original solver discussed in the next subsection. To use functions from Cantera like multicomponent mass diffusion coefficients or reaction rates, the interface presented above has to be loaded dynamically during runtime or linked while compiling. Note, that accurate expressions for the continuity equation 3.9 as well as for the momentum equation 3.10 are already implemented in the original solver and therefore not altered.

5.3.1. OpenFOAM's Approach for Transport Equations

The original transport equations for species and sensible enthalpy are

$$\frac{\partial}{\partial t}(\rho Y_k) = -\vec{\nabla}\cdot(\rho Y_k\vec{v}) + \vec{\nabla}\cdot(\mu\vec{\nabla}Y_k) + \dot{r}_k \tag{5.1}$$

$$\frac{\partial}{\partial t}(\rho h_s) = -\vec{\nabla}\cdot(\rho h_s\vec{v}) + \vec{\nabla}\cdot(\frac{\lambda}{c_p}\vec{\nabla}h_s) + \frac{\mathrm{D}p}{\mathrm{D}t} + \dot{\vec{q}}_{reaction} \tag{5.2}$$

One can see, that equation (5.1) is based on the assumption that the ratio of momentum diffusivity and mass diffusivity is unity. Thus, the dimensionless Schmidt number Sc yields

$$Sc = \frac{\mu}{\rho D} = 1 \rightarrow \rho D = \mu\ . \tag{5.3}$$

Therefore, the mass diffusivity D has to be the same for all species and the species diffusion process is entirely controlled by the mixture viscosity, which is calculated from

Sutherland's law. Additionally, cross diffusion process based on pressure and temperature gradients are neglected.

In equation (5.2) a unity Lewis number Le as well as the disregards from above are assumed:

$$Le = \frac{\lambda}{Dc_p\rho} = 1 \rightarrow \frac{\lambda}{c_p} = \rho D \tag{5.4}$$

It has been shown by [12], that equations (5.1) and (5.2) yield large errors while simulating chemically reacting flows.

5.3.2. Implementation of Species Mass Equations

In nearly isobar combustion the flux caused by the pressure gradient in equation (3.20) can be neglected. External forces occur for example in the diffusion of electrically charged particles and will be neglected, too. With these simplifications and the relation

$$\vec{\nabla} X_j = \frac{\overline{M}}{\overline{M}_j}\vec{\nabla} Y_j - Y_j\frac{\overline{M}}{\overline{M}_j}\sum_{i=1}^{N}\frac{\overline{M}}{\overline{M}_i}\vec{\nabla} Y_i \ , \tag{5.5}$$

equation (3.20) can be rearranged to

$$\vec{j}_k = \rho\frac{\overline{M}_k}{\overline{M}}\sum_{j=1}^{N} D_{kj}\vec{\nabla} Y_j - \rho\frac{\overline{M}_k}{\overline{M}}\sum_{j=1}^{N} D_{kj}Y_j\sum_{i=1}^{N}\frac{\overline{M}}{\overline{M}_i}\vec{\nabla} Y_i - D_k^T\frac{1}{T}\vec{\nabla} T \ , \tag{5.6}$$

to obtain the mass diffusive flux in terms of mass fractions.

For numerical stability, including a discretized second derivation or diffusion term into the solution matrix is essential [7]. A direct implementation of the mass diffusive flux equation (5.6) is therefore not worthwhile, because of the missing diffusion terms which can be discretized implicit by OpenFOAM's class *fvm*. Through the use of the, in section 3.3 already mentioned, characteristics of the ordinary diffusion coefficient matrix, equation (5.6) is rearranged to

$$\vec{j}_k = \sum_{j\neq k}^{N}\rho\frac{\overline{M}_k}{\overline{M}}D_{kj}\vec{\nabla} Y_j - \sum_{j\neq k}^{N}\sum_{i\neq k}^{N}\frac{\overline{M}_k}{\overline{M}_i}\rho D_{kj}Y_j\vec{\nabla} Y_i - \sum_{j\neq k}^{H}\rho D_{kj}Y_j\vec{\nabla} Y_k - D_k^T\frac{1}{T}\vec{\nabla} T \ , \tag{5.7}$$

so that the third term on the right hand side can be discretized implicit by OpenFOAM.

Finally, by inserting the expression for the mass flux above into equation (3.7) the transport equation for species k reads

$$\begin{aligned}&\frac{\partial}{\partial t}(\rho Y_k) + \vec{\nabla}\cdot(\rho Y_k\vec{v}) - \sum_{j\neq k}^{N}\vec{\nabla}\cdot(\rho D_{kj}Y_j\vec{\nabla} Y_k) = \\ &-\sum_{j\neq k}^{N}\vec{\nabla}\cdot\left(\rho\frac{\overline{M}_k}{\overline{M}}D_{kj}\vec{\nabla} Y_j\right) + \sum_{j\neq k}^{N}\sum_{i\neq k}^{N}\vec{\nabla}\cdot\left(\frac{\overline{M}_k}{\overline{M}_i}\rho D_{kj}Y_j\vec{\nabla} Y_i\right) + \vec{\nabla}\cdot\left(D_k^T\frac{1}{T}\vec{\nabla} T\right) + \dot{r}_k \ ,\end{aligned} \tag{5.8}$$

in which all terms discretized implicit by OpenFOAM's class *fvm* stand on the left hand side and all terms included and calculated as explicit source terms by OpenFOAM's class *fvc* stand on the right hand side. Listing A.1 in the appendix shows the implementation of equation (5.8) in OpenFOAM.

5.3.3. Implementation of Sensible Enthalpy Equation

As mentioned above a diffusive flux term has to be included into the solution matrix. Therefore, it is necessary to express Fourier's Law in terms of sensible enthalpy. Starting with an expression of the enthalpy gradient for a multicomponent gas mixture given by [21],

$$\vec{\nabla} h = c_p \vec{\nabla} T + \sum_{k=1}^{N} h_k \vec{\nabla} Y_k \ , \tag{5.9}$$

one can, by express the specific enthalpy with the sum of sensible and chemical enthalpy (3.13), modify Fourier's law to

$$\begin{aligned} \lambda \vec{\nabla} T &= \frac{\lambda}{c_p} \vec{\nabla} h - \sum_{k=1}^{N} \frac{\lambda}{c_p} h_k \vec{\nabla} Y_k \\ &= \frac{\lambda}{c_p} \vec{\nabla} h_s + \sum_{k=1}^{N} \frac{\lambda}{cp} h_{c,k} \vec{\nabla} Y_k - \sum_{k=1}^{N} \frac{\lambda}{c_p} h_{c,k} \vec{\nabla} Y_k - \sum_{k=1}^{N} \frac{\lambda}{c_p} h_{s,k} \vec{\nabla} Y_k \\ &= \frac{\lambda}{c_p} \vec{\nabla} h_s - \sum_{k=1}^{N} \frac{\lambda}{c_p} h_{s,k} \vec{\nabla} Y_k \ . \end{aligned} \tag{5.10}$$

Now, the first term on the right hand side can be discretized implicit by OpenFOAM.

The diffusive species flux from equation (3.31) in terms of sensible Enthalpy gives

$$\sum_{k=1}^{N} \vec{j}_k h_k = \sum_{k=1}^{N} \vec{j}_k (h_{c,k} + h_{s,k}) \ . \tag{5.11}$$

Combining equations (3.15), (3.31), (5.10), (5.11) and neglecting viscous dissipation $\tau : \vec{\nabla} \vec{v}$ and the Soret effect one obtains

$$\begin{aligned} &\frac{\partial}{\partial t}(\rho h_s) + \vec{\nabla} \cdot (\rho h_s \vec{v}) - \vec{\nabla} \cdot \left(\frac{\lambda}{c_p} \vec{\nabla} h_s \right) = \\ &\qquad \frac{\mathrm{D}p}{\mathrm{D}t} - \sum_{k=1}^{N} \vec{\nabla} \cdot \left(\frac{\lambda}{c_p} h_{s,k} \vec{\nabla} Y_k \right) - \sum_{k=1}^{N} \vec{\nabla} \cdot (h_{s,k} \vec{j}_k) - \sum_{k=1}^{N} h_{c,k} \dot{r}_k \ , \end{aligned} \tag{5.12}$$

in which all terms discretized implicitly by OpenFOAM's class *fvm* stand on the left hand side and all terms included and calculated as explicit source terms by OpenFOAM's class *fvc* stand on the right hand side. The diffusive species flux $\vec{j}_k$ is calculated by interpolation from the species equation noted in listing A.1 as "J[k]". Listing A.2 in the appendix shows the implementation of equation 5.12 in OpenFOAM.

6. Validation of the Solver

In this chapter, the implemented solver introduced in chapter 5 is validated with focus on the transport properties and reaction rates. The reference solution is obtained from CHEMKIN/PREMIX with equal physical conditions. In addition, the effect of the grid resolution onto the solution and the chemical time scales are investigated.

6.1. Case Description

Figure 6.1 shows the physical domain of the considered case. Here, the fresh gas mixture flows from the left into the domain. Due to thermal conduction, the mixture is preheated (dark blue) until it ignites (light blue). In the following, the outburn proceeds (red zone) and the adiabatic flame temperature is reached.

λ	$Y_{CH_4,\mathrm{in}}$	$Y_{O_2,\mathrm{in}}$	$Y_{N_2,\mathrm{in}}$	T_{in} [K]	p [bar]	$v_{x,\mathrm{in}}$ [m/s]
1.0	0.055	0.2202	0.7248	300	1.0	0.4

Tab. 6.1.: Physical conditions of the simulation with OpenFOAM.

The mixture is stoichiometric with $\lambda = 1$ and the mass fractions at the inlet are $Y_{O_2,\mathrm{in}} = 0.2202$, $Y_{CH_4,\mathrm{in}} = 0.055$ and $Y_{N_2,\mathrm{in}} = 0.7248$. The fluid's temperature at the inlet is $T_{\mathrm{in}} = 300$ K with an inflow velocity $v_{x,\mathrm{in}} = 0.4$ m/s. The pressure in the whole domain as well as outside is constantly $p = 1$ bar. The physical conditions (shown in table 6.1) are chosen to be the same as in the reference calculation.

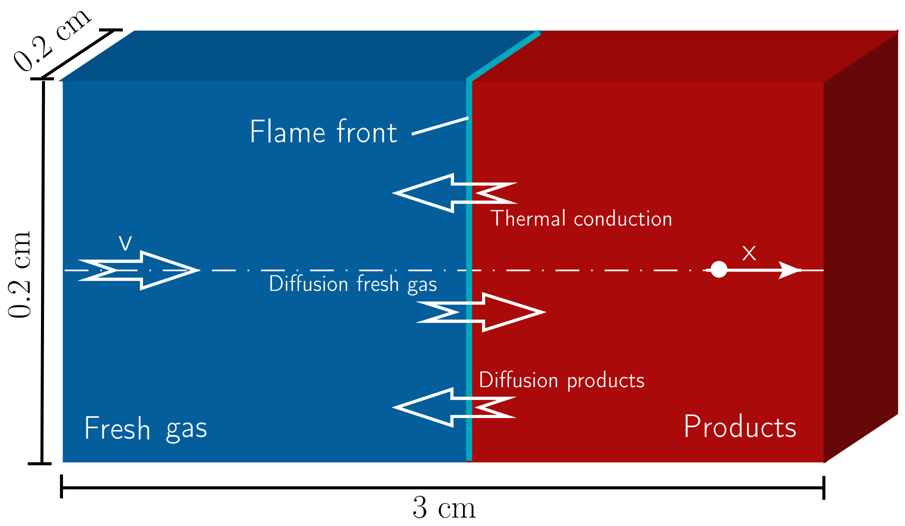

Fig. 6.1.: Physical domain of the laminar, premixed flame.

6.2. Numerical conditions

Table 6.2 shows the numerical setup of the various simulations performed with the new solver in OpenFOAM. The different grids consist of 600 and 3000 equidistant rectangular cells built with the blockMesh tool from OpenFOAM. The grid resolutions are $\Delta x = 5 \times 10^{-5}$m and $\Delta x = 1 \times 10^{-5}$m for each cell. The time steps are set to $\Delta t = 5 \times 10^{-7}$s on the coarse grid and $\Delta t = 1 \times 10^{-7}$s on the fine mesh. For all simulation, including the reference calculation, the mechanism by [38], which contains 17 species and 58 reactions (see Appendix B), is used to calculate the chemical kinetics. For thermodynamic and molecular properties the same input files are used in OpenFOAM and CHEMKIN/PREMIX.

	Domain [m]	Cells	Δt [s]	Δx [m]
Coarse	0.03	600	5×10^{-7}	5×10^{-5}
Fine	0.03	3000	1×10^{-7}	1×10^{-5}

Tab. 6.2.: Numerical setups.

Figure 6.2 shows the numerical domain. At the inlet, fixed values are chosen as boundary condition for all physical properties. The boundary conditions at the outlet differ because of different requirements. For mass fraction and temperature a simple zero gradient boundary condition is adequate. To control possible backflow at the outlet a mixed boundary condition, the inletOutlet condition, is needed for the velocity. It switches the boundary condition between a fixed value and zero gradient depending on direction of the velocity. Since the numerical domain is just a clipping of the infinity large physical domain, the front, back, top and bottom sides are defined empty. Additionally, the numerical domain is only divided into cells, marked by the dashed lines, in the x-direction. Therefore, only a solution in this direction is approached leading to a one dimensional case.

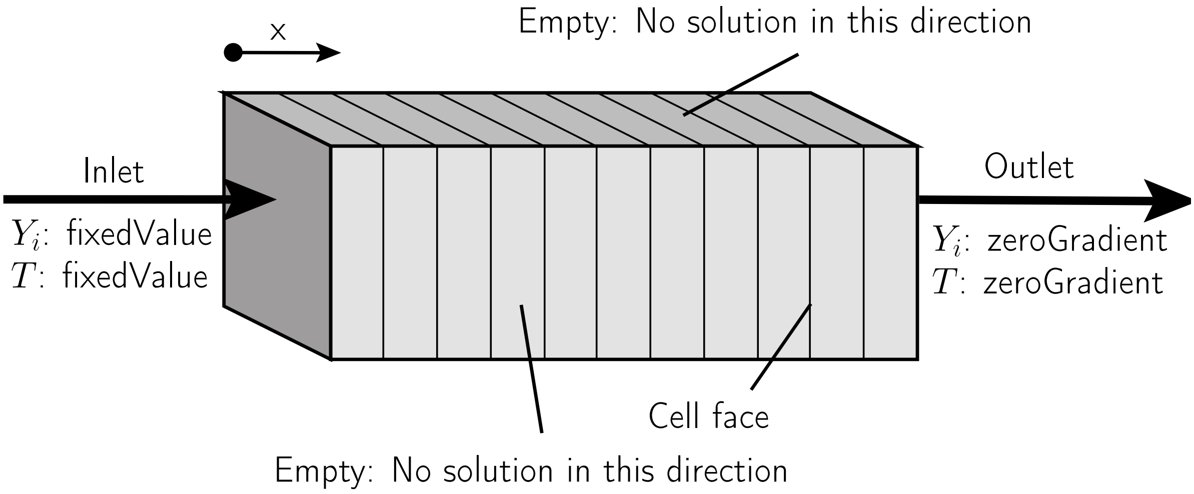

Fig. 6.2.: Numerical domain of the laminar, premixed flame.

In the calculations the implicit Euler method is used to discretize the time integral. Like mentioned in chapter 4 the implicit Euler method is of order $\mathcal{O}(1)_t$. The spatial discretization is accomplished via Gaussian integration and using linear interpolating schemes of order $\mathcal{O}(2)_x$. The discretization schemes used in OpenFOAM for the different terms in the transport equations are listed in table 6.3.

Mathematical Term	Expression in OpenFOAM	Discretization	Order
$\partial\phi/\partial t$	ddt	Euler implicit	$\mathcal{O}(1)_t$
$\nabla(\phi)$	grad	Gauss linear	$\mathcal{O}(2)_x$
$\nabla\cdot\phi$	div	Gauss linear/limitedLinear 1	$\mathcal{O}(2)_x$
$\nabla\cdot(\nabla\phi)$	laplacian	Gauss linear corrected	$\mathcal{O}(2)_x$
–	interpolation	linear	$\mathcal{O}(2)_x$

Tab. 6.3.: In OpenFOAM used discretization schemes for the mathematical terms.

To obtain similar conditions like in CHEMKIN/PREMIX, no transport equation for velocity and pressure is solved. The remaining transport equations are solved using different iterative solver and preconditioners, namely for density the Preconditioned conjugate gradient (PCG) solver with Diagonal incomplete-Cholesky (DIC) preconditioning and for sensible enthalpy and species mass fractions the Preconditioned biconjugate gradient (PBiCG) solver with Diagonal incomplete-LU (DILU) preconditioning.

For the initialization of the simulation a homogeneous flow field with conditions like at the inlet is shortly ignited. Therefore, an spatial limited source term is included in the transport equation of enthalpy with an ignition strength of 5×10^9 J/m^3·s for 0.0002 s. In the ignited cells two flame fronts develop. The downstream traveling flame front leaves the domain and is not of further interest.

6.3. Comparison with CHEMKIN/PREMIX

To validate the OpenFOAM solver a simulation with 3000 cells is performed. The focus of the validation are the newly implemented equations for species and sensible enthalpy as well as the transport coefficients and reaction rates obtained from Cantera. The calculations are compared to results obtained from CHEMKIN/PREMIX with similar conditions. CHEMKIN/PREMIX is a highly specialized computer code for premixed laminar flames in 1D [20]. The code presumes isobar conditions and a steady state flame. To get similar conditions in OpenFOAM® no pressure correction is performed and the velocity field is calculated from

$$\frac{\dot{m}}{A} = \rho_0 v_0 = \rho v = \text{const.} \rightarrow v = \frac{\rho_0 v_0}{\rho} \quad \text{and} \quad \rho = \frac{p}{T R_s} \ . \tag{6.1}$$

in which $R_s = R/\overline{M}$ is the specific gas constant of the mixture. For this reason, the pressure is forced to remain constant at all times in the complete domain. Furthermore, in CHEMKIN/PREMIX the inlet mass flux $\dot{m}$ is changed during the simulation in such a way to hold the flame front steady. Due to missing moving mesh features implemented into the new solver it is therefore not possible to completely prevent a motion of the flame front. CHEMKIN/PREMIX is capable of dynamic mesh refinement and applies for the current case a total number of only 185 grid points with the minimal distance of $\Delta x_{\min} \approx 3 \times 10^{-6}$m.

6.3.1. Temperature and Species Mass Fractions

In figure 6.3 the temperature profile from 3000 Cells (solid) is compared to the reference calculation (circles). The solutions are nearly the same, a part from two notable disagreements at the beginning and the end of the inner reaction zone with maximum variations of about 17 K. One can see, that the grid used by CHEMKIN/PREMIX at the beginning of the reaction zone is too low against the fine mesh from OpenFOAM. Additionally, the discretization scheme used by CHEMKIN/PREMIX is the backward-difference scheme. It is of order one and introduces, similar to the upwind interpolation (UDS) mentioned before, artificial diffusion. Therefore, CHEMKIN/PREMIX spreads out the solution on a coarse mesh [7, 20] and can not resolve the large gradient as accurately as OpenFOAM.

Case	$Y_{CO_{max}}$	$Y_{CH_3\,max}$	$Y_{OH_{max}}$	$Y_{H_2\,max}$	s_L [m/s]
CHEMKIN/PREMIX	0.051967	0.00277	0.00510	0.00163	0.373
OpenFOAM	0.05240	0.00274	0.00497	0.00163	0.376
Deviation[a]	0.83 %	1.08 %	2.54 %	0.00 %	0.8 %

[a] Deviation based on CHEMKIN/PREMIX calculations

Tab. 6.4.: Comparison of distinctive results obtained with OpenFOAM and CHEMKIN/PREMIX.

In figure 6.3 the species mass fractions Y_{CH_4}, Y_{O_2}, Y_{CO_2}, Y_{CO} and Y_{H_2O} (solid) are compared to the reference calculation (circles). As one can see, the solution obtained from

OpenFOAM is very similar to the reference solution. The profiles of intermediate species mass fractions Y_{OH}, Y_O, Y_{H_2}, Y_{CH_3} and Y_{CH_2O} as well as the minimal species mass fractions Y_{CH_2}, Y_H, $Y_{H_2O_2}$, Y_{HCO} and Y_{HO2} confirm that conclusion, although there are tiny discrepancies (see table 6.4) between the reference calculation and OpenFOAM which have a maximum variation at Y_{OH} of about 1.3×10^{-4}. Again, the different grid resolution between OpenFOAM and CHEMKIN/PREMIX in these regions may be the reason. By using equation (2.4) the laminar burning velocity calculates to $s_L = 0.376$ m/s. As one can see in table 6.4 this is very similar to the value obtained from CHEMKIN/PREMIX. Additionally, a comparison with experimental results, e.g. [39], confirms a very good agreement of OpenFOAM's value, too.

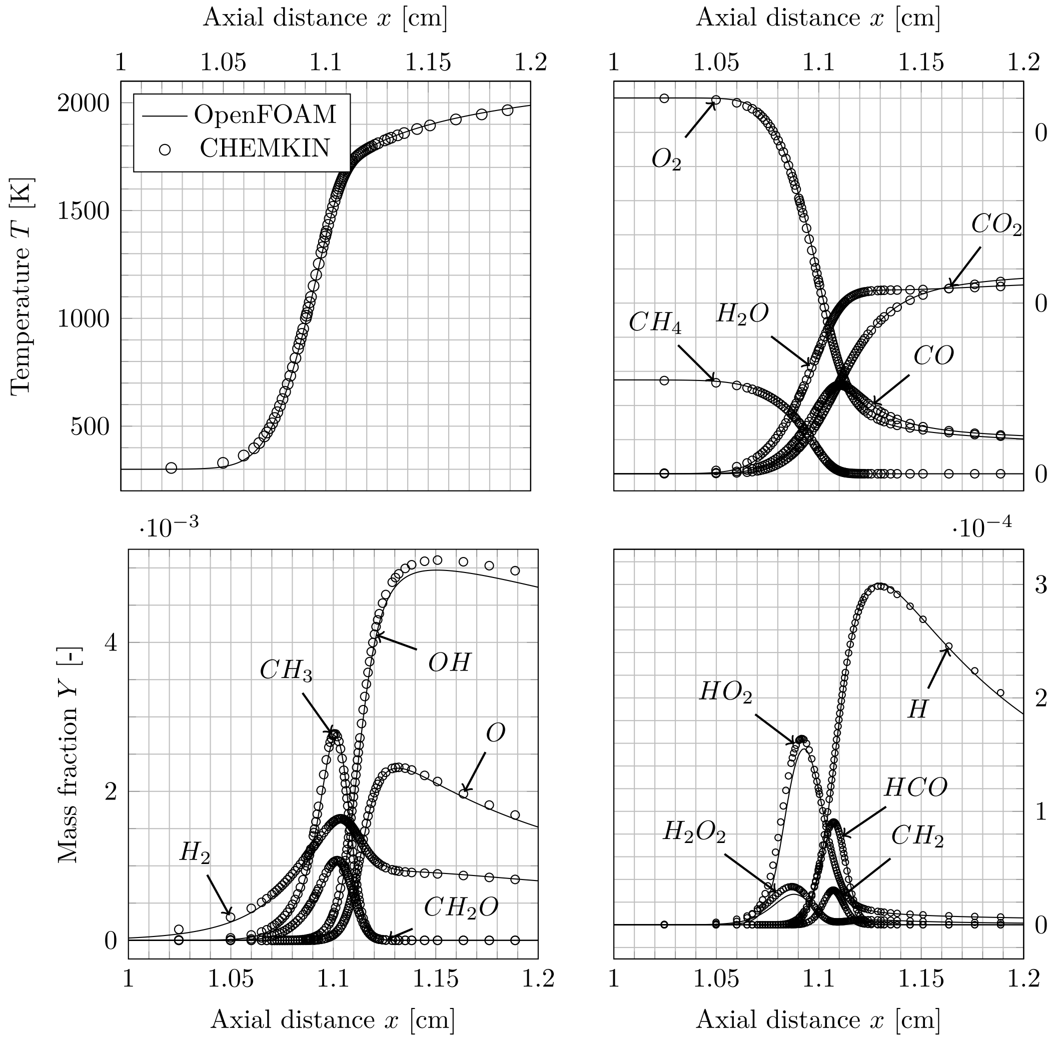

Fig. 6.3.: Temperature and species mass fraction profiles obtained from OpenFOAM (lines) and from CHEMKIN/PREMIX (circles).

6.4. Influence of the Grid Resolution

The calculation time in a numerical simulation highly scales with the number of cells [7]. Therefore, knowledge about the influence of the grid resolution and the time step onto the solution is essential. In this section the validated grid from the previous section with 3000 cells is compared to the coarse grid with 600 cells. The CFL number was kept the same leading to time steps up to $\Delta t = 5 \times 10^{-7}$ s.

In figure 6.4 the temperature profile obtained with 600 Cells (dashed) is compared to the one obtained with 3000 Cells (cross). The two noticeable disagreements at the beginning and the end of the inner reaction zone show maximum variations of about 15 K.

The heat release calculated from equation 3.16 is shown at the bottom of figure 6.4. One can see, that the reaction zone is slightly displaced and thicker in the solution obtained with the coarse grid compared to the one obtained with the finer grid. This is consistent with the mass fraction profiles mentioned further down. Moreover, one can identify, that most of the heat is released at higher temperatures. Upstream from the flame front, no chemical reactions occur and the temperature rises due to thermal conduction.

Case	$Y_{CO_{max}}$	$Y_{CH_3\,max}$	$Y_{OH_{max}}$	$Y_{H_2\,max}$	s_L [m/s]
600 Cells	0.05080	0.00248	0.00497	0.00162	0.385
3000 Cells	0.05240	0.00274	0.00497	0.00163	0.376
Deviation[a]	3.14 %	10.0 %	0.0 %	0.61 %	2.33 %

[a] Deviation based on 600 Cells calculations

Tab. 6.5.: Comparison of distinctive results obtained with 600 Cells and 3000 Cells.

Figure 6.4 shows the major species mass fractions Y_{CH_4}, Y_{O_2}, Y_{CO_2}, Y_{CO} and Y_{H_2O} for 600 Cells (dashed) compared to 3000 Cells (line). One can see, that the solution obtained from the coarser grid is very similar to the one obtained with the finer grid. The profiles of the minor species mass fractions Y_{OH}, Y_O, Y_{H_2}, Y_{CH_3} and Y_{CH_2O} as well as the minimal species mass fractions Y_{CH_2}, Y_H, $Y_{H_2O_2}$, Y_{HCO} and Y_{HO2} in figure 6.4 confirm this conclusion. From the differences between the mass fractions profiles of 600 Cells and 3000 Cells one can see, that the peaks of the intermediate species differ only slightly in magnitude and spatial position. The laminar burning velocity s_L is nearly the same for all grids.

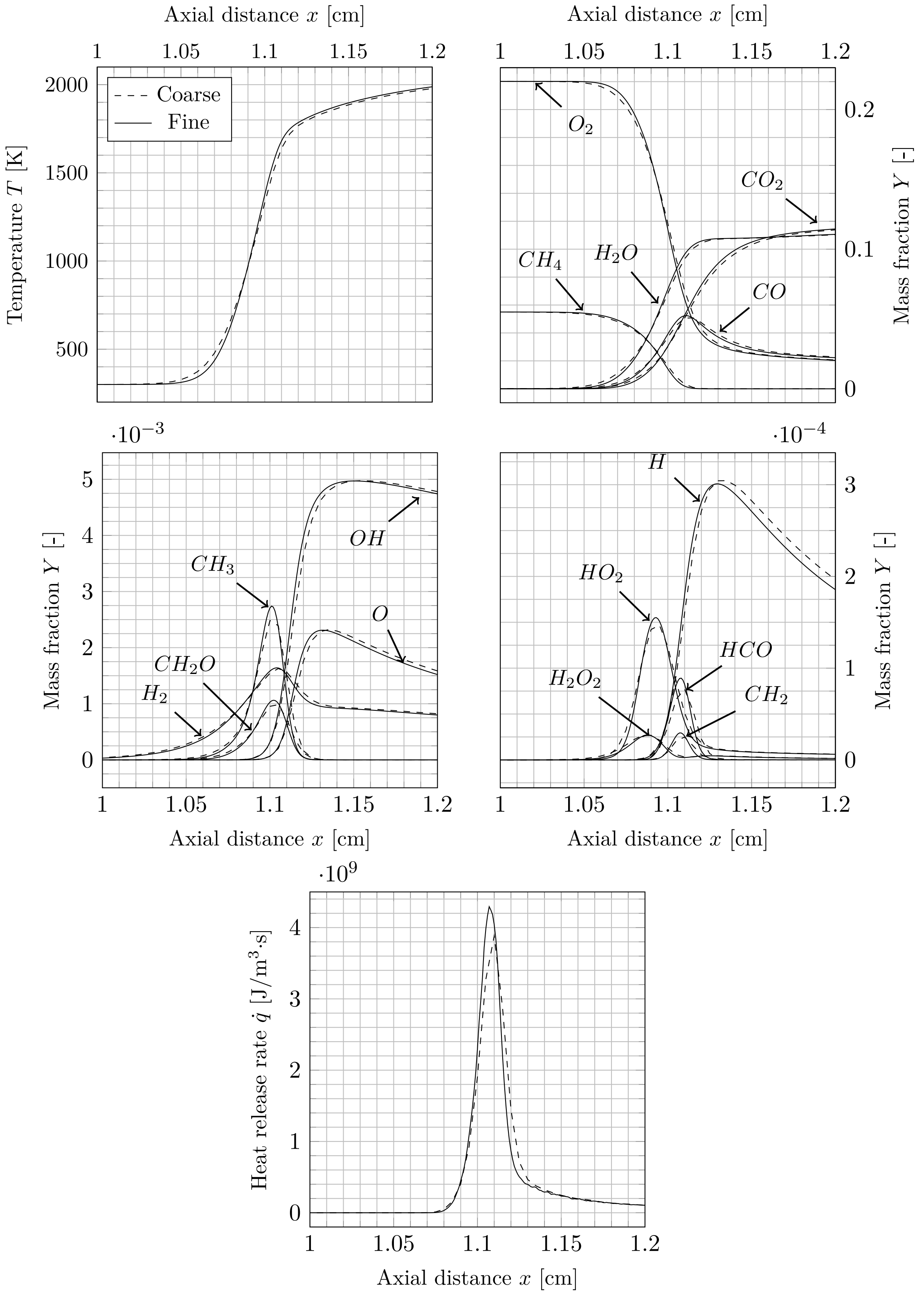

Fig. 6.4.: Profiles obtained from OpenFOAM with 600 Cells (dashed line) and 3000 Cells (line).

6.5. Species Production Rates

In figure 6.5 the computed reaction rates for the species are shown. By comparing the reaction rates with the mass fraction profiles from the section before, one can see, that species with high diffusion coefficients (e.g. H radicals) diffuse upstream against the flow into the preheat region. Here, HO_2 begins to form which is later, at higher temperatures, transformed into OH radicals. The OH radicals become relatively abundant versus the O or H radicals to latter stages of the reaction zone. Throughout the reaction zone, CO molecules build up until they are rapidly consumed by the abundant OH radicals. This is responsible for the major heat release of the combustion process and also marks the end of the reaction zone.

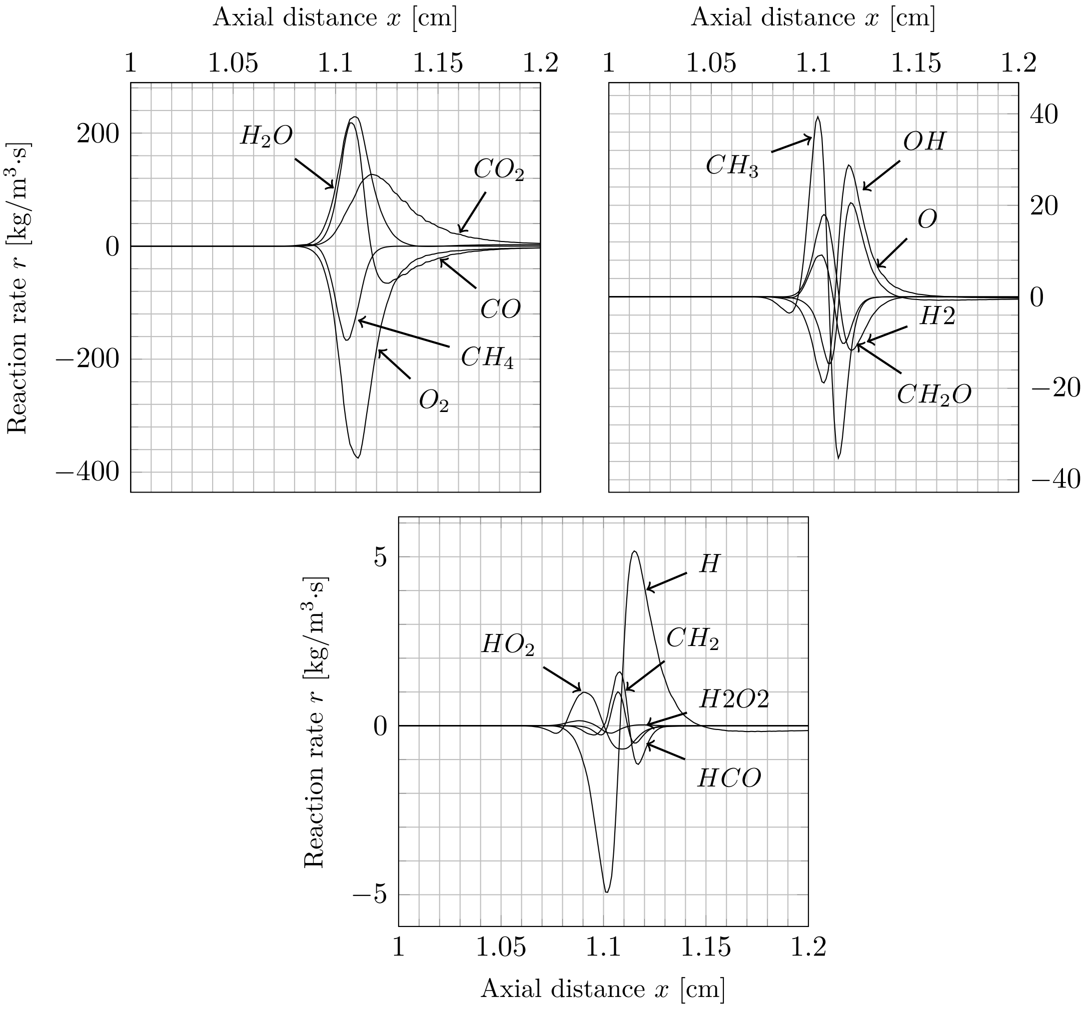

Fig. 6.5.: Reaction rates obtained from OpenFOAM.

6.6. Computation of Chemical Time Scales

To determine the necessary time step for the numerical simulation of the turbulent premixed flame the chemical time scales were calculated from equation (3.39). Figure

6.6 shows the time scales for T, O_2 and OH against the temperature.

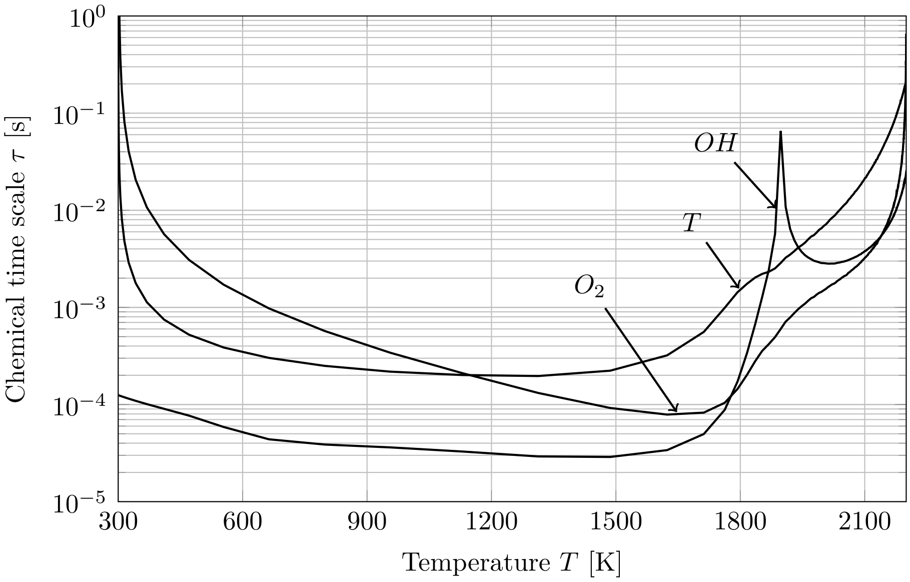

Fig. 6.6.: Calculated chemical time scales τ for T, O_2 and OH.

One can see, that in an region, which corresponds to the main reaction zone, the time scales are almost constant. For intermediate species the time scales show characteristic peaks which correspond to formation and oxidation reactions. The minimum of the calculated time scales determines the chemical time scale of the simulation. Here, the time scale ranges between 10^{-5} s and 10^{-4} s. The time scale of temperature, which is the most important for combustion modeling, is 2×10^{-4} s in this case.

7. Direct Numerical Simulation of a Turbulent Premixed Flame in 3D

In this chapter, the simulation of a turbulent premixed flame in 3D is presented. The main purpose is to show that a direct numerical simulation of turbulent reacting flow using the validated solver is in principle possible. Therefore, the results are only briefly discussed with main focus on the flame topology. In addition, since the simulations are carried out on a Cray XE6 supercomputer [40], the parallel performance of the solver in conjunction with the supercomputer is investigated.

7.1. Numerical Setup

Based on the results from the last chapter, the physical and numerical conditions are chosen, see table 7.1. It has been shown before, that a numerical grid with $\Delta x = \Delta y = \Delta z = 5 \times 10^{-5}$m and a time step $\Delta t = 5 \times 10^{-7}$s is a good compromise between the desired accuracy and calculation time. This conclusion is later approved by the calculated Kolmogorov scales. Figure 7.1 shows the computational domain for the three dimensional case. The dimension is $(1.5 \times 0.75 \times 0.75)$ cm^3, which is thus half the length to the one dimensional case, with $300 \times 150 \times 150 = 6.75 \times 10^6$ equidistant cells of size $\Delta x = \Delta y = \Delta z = 5 \times 10^{-5}$m. The physical conditions listed in table 7.1 are similar to the one dimensional case in chapter 6 except for the inlet velocity field which has the mean value $\vec{\bar{v}}_{\text{in}} = 4.0$ m/s and fluctuates with $\vec{v}'_{\text{in}} = 2$ m/s. The Reynolds number based on integral dimensions at the inlet is $Re = 2000$. The main differences, besides the turbulence, are the three dimensional resolution of the shorter physical domain and the needed solution of three additional transport equations for velocity and one equation for pressure-correction. For chemical kinetics and thermodynamic properties as well as molecular quantities the same input files are used as before. The total simulation time is approximately 5 ms, which leads to a wall-clock time of approximately 48 hours on the Cray XE6 with 2048 CPUs.

λ	$Y_{CH_4,\text{in}}$	$Y_{O_2,\text{in}}$	$Y_{N_2,\text{in}}$	T_{in} [K]	p [bar]	$\vec{\bar{v}}_{\text{in}}$ [m/s]	$\vec{v}'_{\text{in}}$ [m/s]	Re
1.0	0.055	0.2202	0.7248	300	1.0	4.0	2.0	2000

Domain [m]	Cells	Δt [s]	Δx [m]	Discret. Order
$(0.015 \times 0.0075 \times 0.0075)$	6.75×10^6	5×10^{-7}s	5×10^{-5}	$\mathcal{O}(2)_x$ $\mathcal{O}(1)_t$

Tab. 7.1.: Physical and numerical conditions of the turbulent, premixed flame in 3D.

Since the flow is three dimensional, solutions in directions outside the domain (in addition to the in- and outlet) are needed. To ensure, that no mass is lost, the front, back, top and bottom faces of the domain are defined as symmetry planes. The boundary conditions at the in- and outlet for temperature T and species mass fractions Y_i are the same as for the one dimensional case. To avoid reflections of pressure waves at the in- and outlet, non-reflective boundary conditions (NRBC) are used for pressure. To generate a turbulent flow at the inlet a turbulent inflow generator by [41] is used for the velocity field. The boundary condition is based on digital filtering of random data to provide temporally and spatially correlated velocities. The mean flow velocity is $\overline{\vec{v}_{\mathrm{in}}} = 4.0$ m/s with a turbulence level of 50 %. The turbulent length scale l_T and the turbulent time scale τ_T are set to $l_T = 1$ mm and $\tau_T = 0.5$ ms.

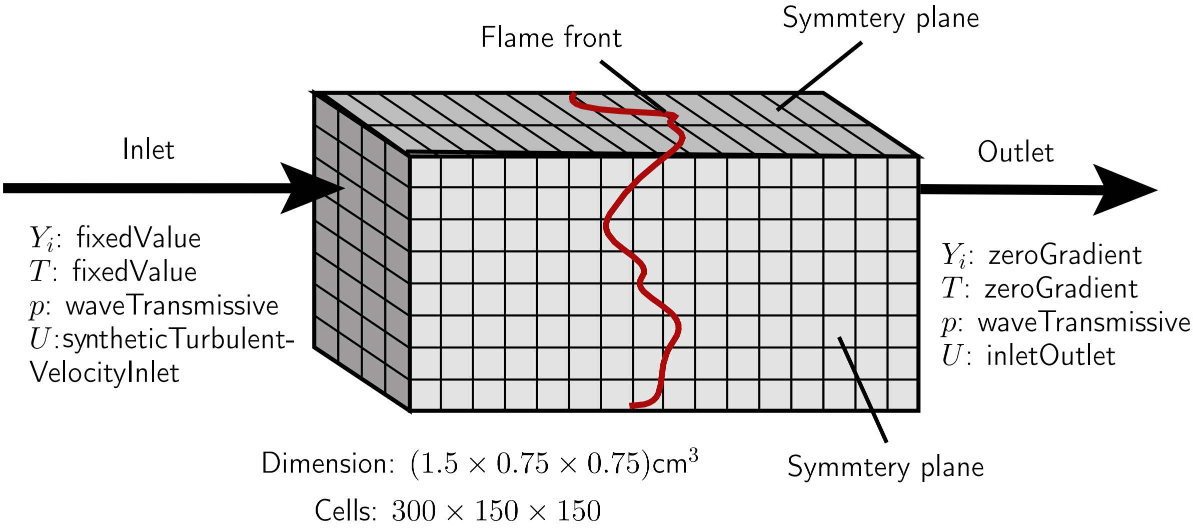

Fig. 7.1.: Numerical domain of the turbulent, premixed flame.

7.2. Initial conditions

The initial conditions have been produced by a DNS on the same computational grid, however, without chemical reactions. Then, the flat flame front with suitable temperature and mass fraction fields from the one dimensional case has been superimposed into the flow. Figure 7.2 shows the initial velocity and the temperature fields right before the simulation starts. One can see, that the flame front (marked with red colors) is still planar since the passed time is short.

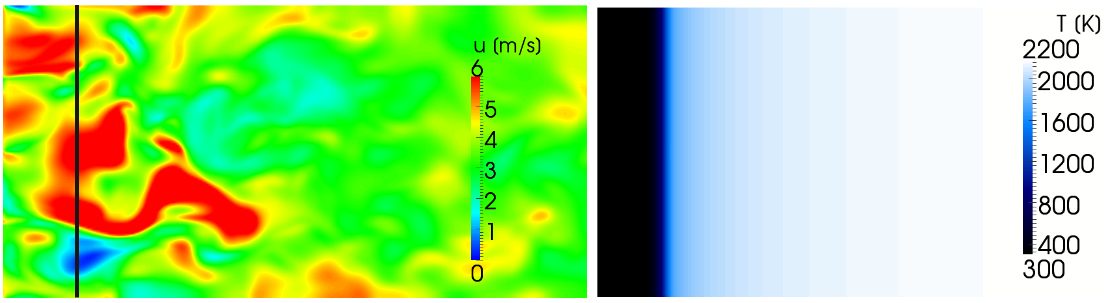

Fig. 7.2.: Initial velocity and temperature fields of the premixed turbulent flame.

Figure 7.3 shows the Kolmogorov scales. They are calculated from the produced initial conditions by using equations (2.6). One can see, that the Kolmogorov length scale l_L varies between 2×10^{-5} m and 2.5×10^{-4} m. Since the small values occur mostly near the inlet the used grid spacing can be expected to resolve the smallest eddies near the flame front sufficiently. The Kolmogorov time scale τ_K varies between 2×10^{-5} s and 2.83×10^{-3} s. Since τ_K is relatively great, the used time step is limited by the Courant number due to numerical stability reasons.

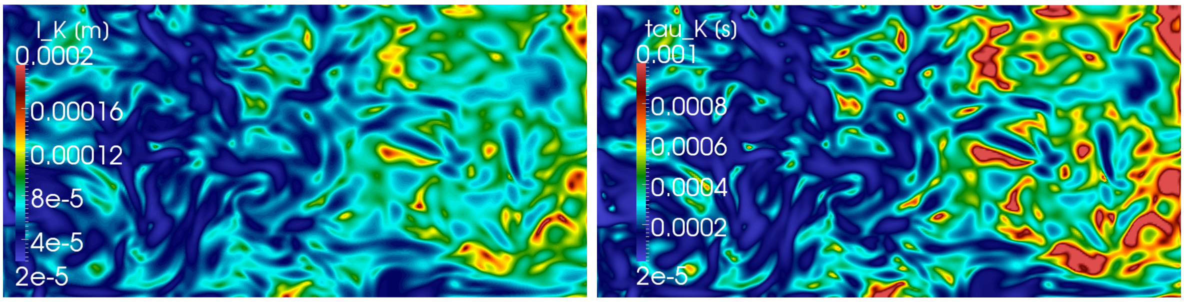

Fig. 7.3.: From the initial conditions calculated Kolmogorov scales.

7.3. Topological Results

The main purpose is to show, that a direct numerical simulation of turbulent reacting flow using the validated solver is possible. Therefore, the results are given only briefly with main focus on the flame topology. Figure 7.4 shows an isosurface of the heat release rate, which can be taken as the position of the flame front. The heat release rate is calculated by using equation (3.16). A corrugated surface, which is caused by the turbulent flow shown in the back can be identified. On the bottom of figure 7.4 the eddy dissipation rate is shown. The eddy dissipation rate is very high at the flame front and near zero downstream, since no eddies are present anymore.

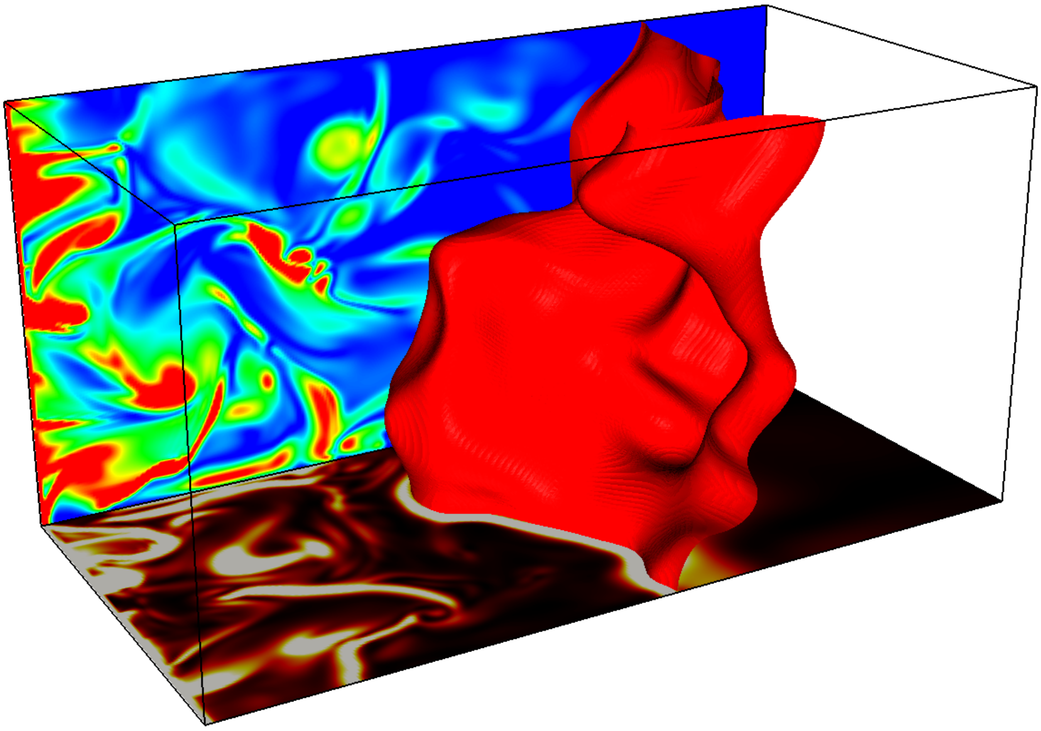

Fig. 7.4.: Isosurface of Y_{CH_4}. Bottom with eddy dissipation rate and backside with vorticity.

The ratio of mean flame surface and turbulent flame surface calculates to $\frac{A_T}{A} = 2.33$. Additionally, by using equation 2.10 the turbulent burning velocity calculates to $s_T = 0.704$ m/s, which is less than two times the laminar burning velocity calculated for the laminar flame in the chapter before. The difference in the laminar flame velocities is caused by the curved and stretched turbulent flame.

Figure 7.5 shows contour plots of the temperature and velocity field. The flame front is marked by iso-contours of the heat release rate. The structure of the turbulence is visible at the inlet and the flow is accelerated by passing the flame front. The temperature increases over the flame front from 300 K up to 2200 K. The velocity gradients in the burnt mixture are clearly smaller caused by high viscosity. A distinct feature of the flame topology is the presence of an instantaneous wrinkled flame front. In areas in which the flame front wrinkles with a convex geometry into the burnt gas, the

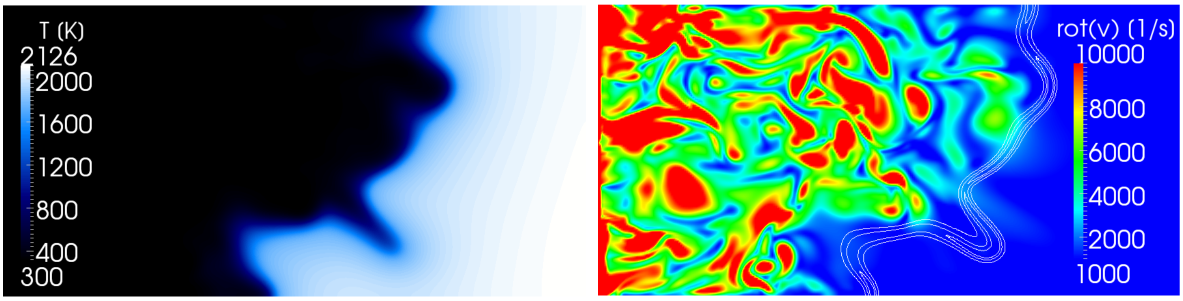

Fig. 7.5.: Two dimensional slices of the temperature field and the vorticity field with heat release rate of the turbulent premixed flame.

heat release is higher. Additionally, due to the turbulent flow, the flame front varies in thickness. In this case, small eddies penetrate the flame front and enhance the diffusion process, leading to a thickened flame front. In areas where the cold flow is slow the flame front even propagates into the unburnt gas. It is evident that the vorticity is greatly damped in the burned gas side because the gas viscosity increases with temperature.

Additionally, in figure 7.6 the mass fractions of CH_4, O_2, H_2 and OH as well as the corresponding reaction rates are shown. One can see, how the reaction zone varies in thickness and the strain of the flame front influences the reaction rates.

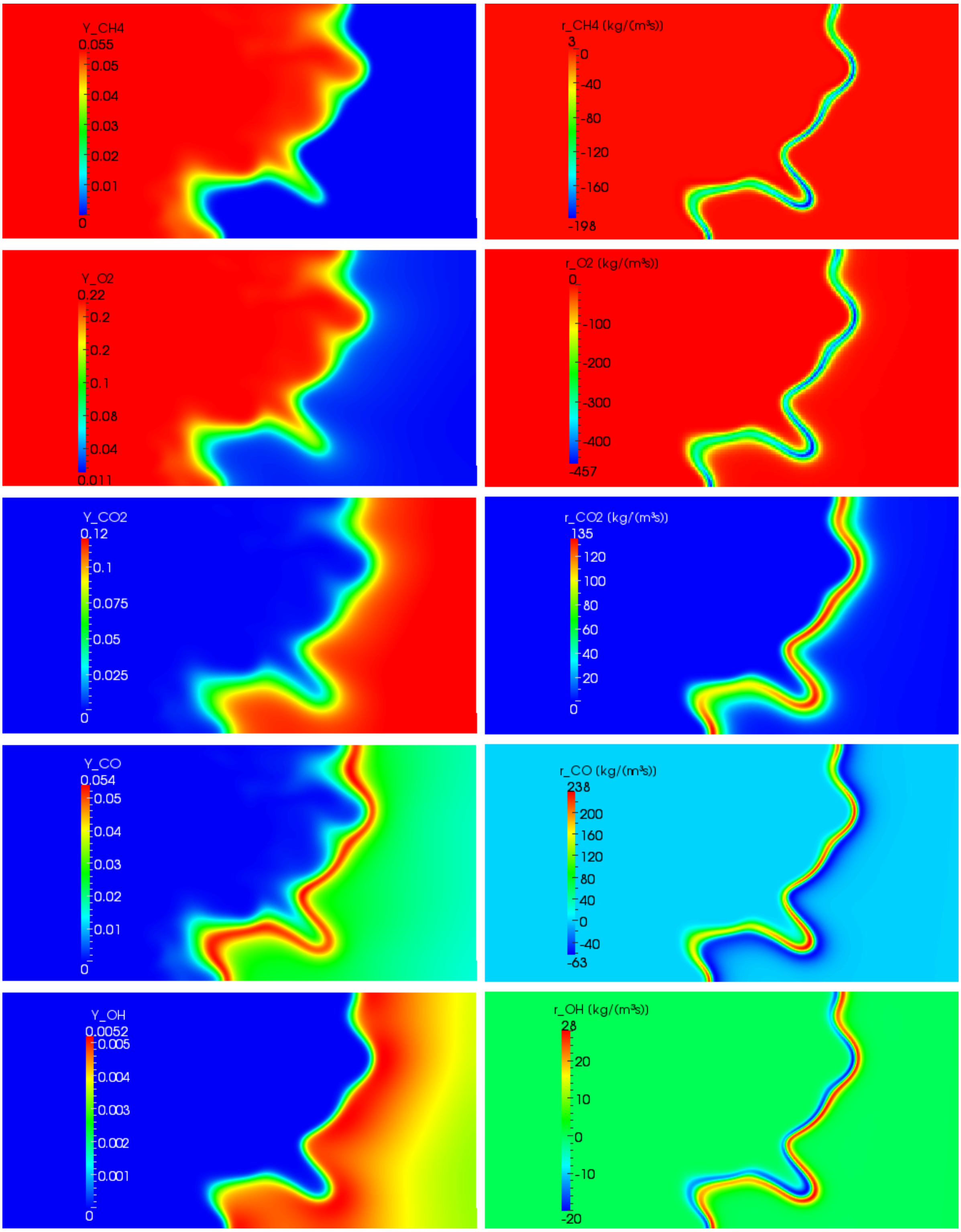

Fig. 7.6.: Different mass fractions and reaction rates.

7.4. Parallel Performance of the Cray XE6 (HERMIT)

To perform direct numerical simulations of chemically reacting flows, at least three velocity equations, one mass equation, one enthalpy equation and $K-1$ species equations have to be solved iteratively. The number even increases if pressure-velocity coupling schemes are used [7]. In addition, calculations for the chemical reaction rates or transport properties have to be performed at least once per time step and cell. In particular, the calculation time of multicomponent mass diffusion coefficients, shown in section 3.3, requires the inversion of a $K \times K$ matrix [16]. Further more, if turbulent flow is considered, it can be shown, that the number of grid points in each direction must be at least l/l_K [7].

Due to the small time steps and the high grid resolution, direct numerical simulations of chemically reacting flows have to be performed in parallel on high performance computing clusters. Since the architecture of each cluster is very unique and an efficient usage of existing resources is essential one has to know about the speed-up particularly of high performance clusters.

The numerical simulation of the turbulent premixed flame in 3D presented before was performed on the Cray XE6 (HERMIT) operated by the High Performance Computing Center Stuttgart (HLRS). It is a massively parallel supercomputer with 3552 compute nodes. Each compute node has two sockets with each 16 AMD Interlagos processors. Each XE6 node has either 32 or 64 GB of DDR3 SDRAM memory. The node-node interconnection is based on CRAY Gemini. The XE6 is operated with the Cray Linux Environment in version 3, including SUSE Linux Enterprise Server and Cray's Compute Node Linux [40, 42].

For the current investigation, several calculations have been carried out to obtain information about parallel efficiency and scalability. The new OpenFOAM solver (see chapter 5) has been used and the conditions for all calculations where the same except for the number of cores. The mesh consists of about 2×10^6 cells.

Figure 7.7 shows the intranode scale-up (left) and internode scale-up (right) calculated with $S_n = t_s/t_n$ [7]. Here t_s is the execution time with the reference processor number and t_n is the execution time using n processors. The reference processor number for the intranode scale-up is 1 CPU and for the internode scale-up equal the number of CPUs per node (32 CPUs). One can see, that the intranode speedup is far from the linear speed up marked as a solid line. Thus, it is not worthwhile using the Cray XE6 with only a few CPUs on one node. On the contrary, the internode performance scales very well up to 2048 processors. Between 64 processors and 1024 processors the scale-up is even superlinear[1][7]. The scale-up indicates a very good scalability of Cray XE6 in conjunction with the solver even for 6.75×10^6 Cells. Therefore, in the DNS of the turbulent flame in the section before 2048 CPUs have been used.

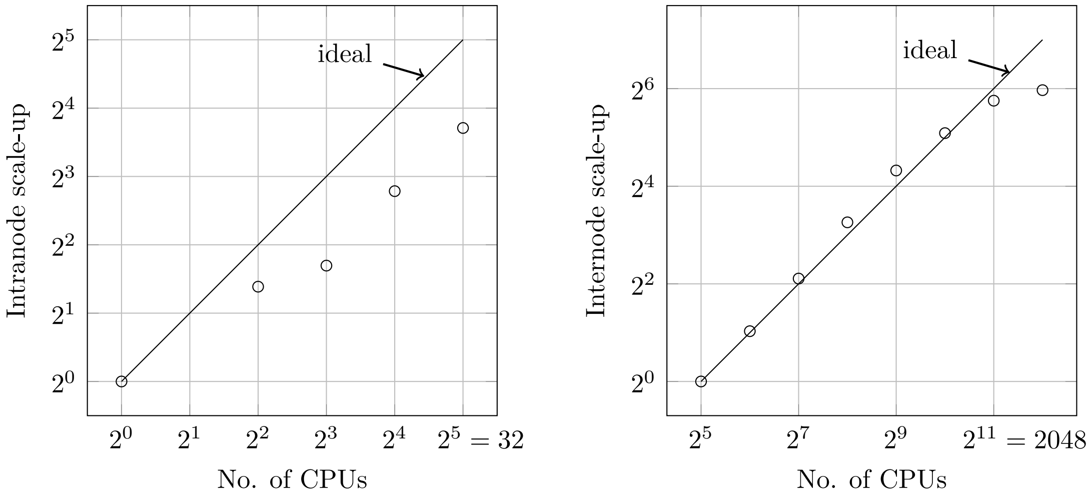

Fig. 7.7.: Scale-up on Cray XE6 with a grid of 2×10^6 cells.

[1] Superlinear is a speedup of more than p if p processors are used. It occurs rarely in parallel computing. [43] showed with a simple example how a superlinear speedup is achieved.

8. Summary and Perspective

A solver for direct numerical simulations of chemically reacting flows has been implemented and validated for the free CFD framework OpenFOAM 2.0.1. The solver takes detailed multicomponent diffusive fluxes into account. The molecular transport coefficients as well as the chemical kinetics and thermodynamic quantities were obtained via a coupling interface from Cantera 1.8. The solver as well as the coupling library were successfully compiled with an MPI implementation on the Cray XE6 for the parallelized computation of DNS of turbulent reacting flows.

To validate the solver, numerical calculations of a laminar premixed flat flame have been carried out. The results were subsequently compared to reference data obtained from the CHEMKIN package. The agreement with respect to the computed flame profiles for T and Y was very good. The calculated flame speed showed reasonable results with experimental data. Additional calculations were performed on a coarse grid to investigate the influence of the gird resolution on the computed flame structure. The results were shown to depend very little on whether the coarse or fine mesh had been used. Subsequently, direct numerical simulation of a turbulent premixed flame in three dimensions have been performed on the Cray XE6 supercomputer with 2048 CPUs. The interaction between turbulence and the flame front has been identified. Thereby, the capability of the solver for complex 3D turbulent flows with chemical reactions as well as the very good scale-up on the Cray XE6 was shown.

For future work, the validated DNS solver will be used for further investigations of the interplay between turbulence and chemical reactions as well as characteristic values like strain rate, the curvature or the turbulent burning velocity. The influence of selective diffusion mechanisms (like e.g. the Soret effect) on the numerical solution of turbulent premixed flames can be studied. Furthermore, essential statistical information, which is needed to develop and validate models used in future RANS and LES simulation, can be provided by the DNS solver.

Thanks to the usage of C++ programming principles it is possible to extend the solver by including new numerical or computational utilities in order to study additional physical phenomena. The investigation of details like the radiative heat transport or the Dufour effect can further enrich such a study. In order to reduce the CPU time in numerical simulations a mixture average approach for the diffusive fluxes can be included as well as an adaptive mesh refinement based on characteristic gradients. Equations for the stretch rate or flame velocity and models for soot or NOx formation can also be incorporated and validated for further examination of turbulent premixed flames.

A. Code Fragments for OpenFOAM in C++3

The following listings show code fragments from the implemented OpenFOAM solver. Listing A.1 shows the implementation of the transport equation (5.8) for species and listing A.2 shows the implementation of the transport equation (5.12) for sensible enthalpy. In addition, listing A.3 shows the implementation of the initialization of the coupling interface and a downcast to obtain access to derived class functions. Note, that for the sake of clarity, the codes are simplified and will not compile without modifications.

```
fvScalarMatrix YEqn
(
    fvm::ddt(rho, Y[k])
  + mvConvection->fvmDiv(phi, Y[k])
  ==
    chemistry.RR(k)
);

forAll(Y, j)
{
  if(j!=k)
  {
    forAll(Y, i)
    {
      if(i!=k)
      {
        YEqn -= fvc::laplacian(W[k]/W[i]*rho*composition.D(k,j)*Y[j],
            Y[i], "laplacian(D,Y[k])");
        J[k] -= fvc::interpolate(W[k]/W[i]*rho*composition.D(k,j)*Y[j]) *
            (fvc::interpolate(fvc::grad(Y[i])) & mesh.Sf());
      }
    }

    YEqn -= fvm::laplacian(rho*composition.D(k,j)*Y[j], Y[k],
        "laplacian(D,Yi)");
    J[k] -= fvc::interpolate(rho*composition.D(k,j)*Y[j]) *
        (fvc::interpolate(fvc::grad(Y[k])) & mesh.Sf());

    YEqn += fvc::laplacian(rho*W[k]*UnitMole*Wm*composition.D(k,j), Y[j],
        "laplacian(D,Y[k])");
    J[k] += fvc::interpolate(rho*W[k]*UnitMole*Wm*composition.D(k,j)) *
        (fvc::interpolate(fvc::grad(Y[j])) & mesh.Sf());
  }
}

YEqn -= fvc::laplacian(composition.DT(k)/T, T, "laplacian(D,Y[k])");
```

```
J[k] -= fvc::interpolate(composition.DT(k)/T) *
    (fvc::interpolate(fvc::grad(T)) & mesh.Sf());

YEqn.relax();
solve(YEqn, mesh.solver("Y[k]"));
```

Listing A.1: Implementation of the species equation 5.8 in OpenFOAM.

```
fvScalarMatrix hsEqn
(
    fvm::ddt(rho, hs)
  + mvConvection->fvmDiv(phi, hs)
  - fvm::laplacian(turbulence->alphaEff(), hs)
  ==
    DpDt
  + chemistrySh
);

forAll(Y, k)
{
  hsEqn -= fvc::laplacian(thermo.alpha()*hsi[k], Y[k]);
  hsEqn -= fvc::div(J[k], hsi[k], "div(Ji,hsi)");
}

hsEqn.relax();
hsEqn.solve();
```

Listing A.2: Implementation of the energy equation 5.12 in OpenFOAM.

```
autoPtr<psiChemistryModel> pChemistry ( psiChemistryModel::New(mesh) );
psiChemistryModel& chemistry = pChemistry();
hsCombustionThermo& thermo = chemistry.thermo();

canteraMixture<canteraThermo>& composition =
    dynamic_cast<canteraMixture<canteraThermo>&>(thermo.composition());
```

Listing A.3: Initialization during run time and performing a downcast to obtain access to derived class functions in OpenFOAM.

B. Reaction Mechanism

Listing B.1 shows the chemical reaction mechanism by Kee et al [38], which was used in the numerical simulations of the 1D flame as well as in the direct numerical simulations. It consists of 17 species and 58 reactions.

```
!Kee, Gracr, Smooke, Miller: SAND Report 85-8240, 1985 Livermore, California, USA
!58 Reactions, 17 Species

ELEMENTS
H O C N
END
SPECIES
H2 O2 H2O H O OH HO2 CO CO2 H2O2
CH4 CH3 CH2 CH CH2O HCO N2
END
REACTIONS
! Units in cm / s / K / cal / mol
CH3+H+M=CH4+M              8.00E26   -3.000       0.
CH4+O2=CH3+HO2             7.90E13    0.000   56000.
CH4+H=CH3+H2               2.20E04    3.000    8750.
CH4+O=CH3+OH               1.60E06    2.360    7400.
CH4+OH=CH3+H2O             1.60E06    2.100    2460.
CH3+O=CH2O+H               6.80E13    0.000       0.
CH3+OH=CH2O+H2             1.00E12    0.000       0.
CH3+OH=CH2+H2O             1.50E13    0.000    5000.
CH3+H=CH2+H2               9.00E13    0.000   15100.
CH2+H=CH+H2                1.40E19   -2.000       0.
CH2+OH=CH2O+H              2.50E13    0.000       0.
CH2+OH=CH+H2O              4.50E13    0.000    3000.
CH+O2=HCO+O                3.30E13    0.000       0.
CH+O=CO+H                  5.70E13    0.000       0.
CH+OH=HCO+H                3.00E13    0.000       0.
CH+CO2=HCO+CO              3.40E12    0.000     690.
CH2+CO2=CH2O+CO            1.10E11    0.000    1000.
CH2+O=CO+H+H               3.00E13    0.000       0.
CH2+O=CO+H2                5.00E13    0.000       0.
CH2+O2=CO2+H+H             1.60E12    0.000    1000.
CH2+O2=CH2O+O              5.00E13    0.000    9000.
CH2+O2=CO2+H2              6.90E11    0.000     500.
CH2+O2=CO+H2O              1.90E10    0.000   -1000.
CH2+O2=CO+OH+H             8.60E10    0.000    -500.
CH2+O2=HCO+OH              4.30E10    0.000    -500.
CH2O+OH=HCO+H2O            3.43E09    1.180    -447.
CH2O+H=HCO+H2              2.19E08    1.770    3000.
CH2O+M=HCO+H+M             3.31E16    0.000   81000.
CH2O+O=HCO+OH              1.81E13    0.000    3082.
```

```
HCO+OH=CO+H2O             5.00E12     0.000         0.
HCO+M=H+CO+M              1.60E14     0.000     14700.
HCO+H=CO+H2               4.00E13     0.000         0.
HCO+O=CO2+H               1.00E13     0.000         0.
HCO+O2=HO2+CO             3.30E13    -0.400         0.
CO+O+M=CO2+M              3.20E13     0.000     -4200.
CO+OH=CO2+H               1.51E07     1.300      -758.
CO+O2=CO2+O               1.60E13     0.000     41000.
HO2+CO=CO2+OH             5.80E13     0.000     22934.
H2+O2=2OH                 1.70E13     0.000     47780.
OH+H2=H2O+H               1.17E09     1.300      3626.
H+O2=OH+O                 5.13E16    -0.816     16507.
O+H2=OH+H                 1.80E10     1.000      8826.
H+O2+M=HO2+M              3.61E17    -0.720         0.
   H2O/18.6/ CO2/4.2/ H2/2.86/ CO/2.11/ N2/1.26/
OH+HO2=H2O+O2             7.50E12     0.000         0.
H+HO2=2OH                 1.40E14     0.000      1073.
O+HO2=O2+OH               1.40E13     0.000      1073.
 2OH=O+H2O                6.00E08     1.300         0.
H+H+M=H2+M                1.00E18    -1.000         0.
H+H+H2=H2+H2              9.20E16    -0.600         0.
H+H+H2O=H2+H2O            6.00E19    -1.250         0.
H+H+CO2=H2+CO2            5.49E20    -2.000         0.
H+OH+M=H2O+M              1.60E22    -2.000         0.
   H2O/5.0/
H+O+M=OH+M                6.20E16    -0.600         0.
   H2O/5.0/
H+HO2=H2+O2               1.25E13     0.000         0.
HO2+HO2=H2O2+O2           2.00E12     0.000         0.
H2O2+M=OH+OH+M            1.30E17     0.000     45500.
H2O2+H=HO2+H2             1.60E12     0.000      3800.
H2O2+OH=H2O+HO2           1.00E13     0.000      1800.
!
END
```

Listing B.1: The used reaction mechanism in CHEMKIN format

Bibliography

[1] R. J. Kee, M. E. Coltrin, and P. Glarborg. *Chemically reacting flow : theory and practice*. Hoboken, NJ: Wiley Interscience, 2003.

[2] *World energy outlook*. Paris, 2010.

[3] S. R. Turns. *An introduction to combustion: concepts and applications*. McGraw-Hill series in mechanical engineering. Boston: McGraw-Hill, 2000.

[4] N. Peters. *Turbulent combustion*. Cambridge monographs on mechanics. Cambridge: Cambridge University Press, 2000.

[5] E. O. Lawrence. “Mixing and chemical reactions in turbulent flow reactors”. PhD thesis. University of California, 1964.

[6] T. Echekki, ed. *Turbulent combustion modeling : advances, new trends and perspectives*. Fluid mechanics and its applications ; 95. Springer, 2011.

[7] M. Ferziger J. H. ; Perić. *Numerische Strömungsmechanik*. Berlin: Springer, 2008.

[8] J. H. Chen. “Petascale direct numerical simulation of turbulent combustion”. In: *Proceedings of the Combustion Institute* 33 (2011), pp. 99–123.

[9] C. Cabrit and F. Nicoud. “Direct simulations for wall modeling of multicomponent reacting compressible turbulent flows”. In: *Physics of Fluids* 21.055108 (2009).

[10] J. Warnatz, U. Maas, and R. W. Dibble. *Combustion : physical and chemical fundamentals, modeling and simulation, experiments, pollutant formation*. Berlin: Springer, 2006.

[11] M. Désilets, P. Proulx, and G. Soucy. “Modeling of multicomponent diffusion in high temperature flows”. In: *International Journal of Heat and Mass Transfer* 40.18 (1997), pp. 4273 –4278.

[12] J. Keller. “Influence of Molecular Diffusion Flux Modeling on the Structure of Non-Premixed and Partially Premixed Flames”. MA thesis. Technische Universität Bergakademie Freiberg, 2011.

[13] A. Spille-Kohoff, E. Preuß, and K. Böttcher. “Numerical solution of multi-component species transport in gases at any total number of components”. In: *International Journal of Heat and Mass Transfer* 55.19–20 (2012), pp. 5373 –5377.

[14] A. W. Cook. “Enthalpy diffusion in multicomponent flows”. In: *Physics of Fluids* 21.055109 (2009).

[15] H. Jasak, A. Jemcov, and Z. Tukovic. “OpenFOAM: A C++ Library for Complex Physics Simulations”. In: *International Workshop on Coupled Methods in Numerical Dynamics*. 2007.

[16] G. Dixon-Lewis. “Flame Structure and Flame Reaction Kinetics. II. Transport Phenomena in Multicomponent Systems”. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 307 (1986), pp. 111–135.

[17] D. Goodwin. *Cantera: Object-oriented software for reacting flows.* September 2012. URL: `\\http://code.google.com/p/cantera/`.

[18] M. Rehm. “Numerische Strömungssimulation der Hochdruckvergasung unter Berücksichtigung detaillierter Reaktionsmechanismen”. PhD thesis. Technische Universität Bergakademie Freiberg, 2010.

[19] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport phenomena.* New York: Wiley, 2002.

[20] F.M. Rupley, R.J. Kee, and J.A. Miller. “PREMIX: A Fortran Program for Modeling Steady Laminar One-dimensional Premixed Flames”. In: *Sandia National Laboratories Tech. Report* SAND85-8240 (1995).

[21] T. Poinsot and D. Veynante. *Theoretical and numerical combustion.* Philadelphia, PA: Edwards, 2001.

[22] S. B. Pope. *Turbulent flows.* Cambridge University Press, 2000.

[23] L. P. H. de Goey and J. H. M. ten Thije Boonkkamp. “A Mass-Based Definition of Flame Stretch for Flames with Finite Thickness”. In: *Combustion Science and Technology* 122.1-6 (1997), pp. 399–405.

[24] R. Brdička. *Grundlagen der physikalischen Chemie.* Berlin: Dt. Verl. d. Wiss., 1977.

[25] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird. *Molecular theory of gases and liquids.* Structure of matter series. New York [u.a.]: Wiley [u.a.], 1954.

[26] S. Chapman and T. G. Cowling. *The Mathematical theory of non-uniform gases : an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases.* Cambridge: Univ.Pr., 1970.

[27] M. Frenklach et al. “GRI-Mech - An Optimized Detailed Chemical Reaction Mechanism for Methane Combustion”. In: *Gas Research Institute Topical Report* GRI-95/0058 (1995).

[28] C. R. Wilke. “A Viscosity Equation for Gas Mixtures”. In: *The Journal of Chemical Physics* 18 (1950), pp. 517–519.

[29] J. I. Steinfeld, J. S. Francisco, and W. L. Hase. *Chemical kinetics and dynamics.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[30] H.-P. Schmid. “Ein Verbrennungsmodell zur Beschreibung der Wärmefreisetzung von vorgemischten turbulenten Flammen”. PhD thesis. Universität Karlsruhe (TH), 1995.

[31] *OpenFOAM - The Open Source CFD Toolbox, Programmer’s Guide.* 2.1.0. OpenFOAM Foundation. 2011.

[32] *OpenFOAM - The Open Source CFD Toolbox, User Guide.* 2.1.1. OpenFOAM Foundation. 2012.

[33] D. G. Goodwin. *Cantera C++ User's Guide*. California Institute of Technology. 2002.

[34] Regents of the University of California. *BSD 3-Clause License*. September 2012. URL: `http://opensource.org/licenses/BSD-3-Clause`.

[35] R.J. Kee et al. *CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics*. Tech. rep. SAND96-8216. Sandia National Laboratories Report, 1996.

[36] B. Stroustrup. *The C++ programming language*. Boston [u.a.]: Addison-Wesley, 2009.

[37] Scott Meyers. *Effective C++ : 55 specific ways to improve your programs and designs*. Addison-Wesley professional computing series. Upper Saddle River, NJ: Addison-Wesley, 2007.

[38] R. J. Kee et al. Tech. rep. SAND85-8240. Sandia National Laboratories Report, 1985.

[39] K.J. Bosschaart and L.P.H. de Goey. "The laminar burning velocity of flames propagating in mixtures of hydrocarbons and air measured with the heat flux method". In: *Combustion and Flame* 136 (2004), pp. 261–269.

[40] *Cray XE6*. Cray Inc. 2010.

[41] M. Klein, A. Sadiki, and J. Janicka. "A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations". In: *Journal of Computational Physics* 186 (2003), pp. 652–665.

[42] B. Krischok. *Cray XE6 (HERMIT)*. 2012. URL: `http://www.hlrs.de/systems/platforms/cray-xe6-hermit/`.

[43] D. Parkinson. "Parallel efficiency can be greater than unity". In: *Parallel Computing* 3 (1986), pp. 261–262.