Technische Universität Wien

University of L'Aquila

Joint Master's Programme - MathMods

Mathematical Modelling in Engineering: Theory, Numerics, Applications

Diplom-Ingenieur	Laurea Magistrale
Technical Mathematics	Ingegneria Matematica
Technische Universität Wien (TUW)	University of L'Aquila (UAQ)

Master's Thesis

Artificial Neural Networks for the Application to Reactive Flow Simulations

Supervisor

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Michael Harasek

Candidate

Marlon Danilo Cabrera Ormaza

Student ID (UAQ): 261391
Student ID (TUW): 11848042

Co-advisor

Projektass. Dipl.-Ing. Eva-Maria Wartha

Academic Year 2019/2020

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Contents

Affidavit **i**

Acknowledgements **ii**

Abstract **iii**

List of Figures **v**

List of Tables **vii**

Acronyms **viii**

List of Symbols **x**

1 Introduction **1**

1.1 Motivation and Problem Statement 1

1.2 Aim of the Work and Objectives 3

1.3 Literature Review 4

2 Reactive Flows **7**

2.1 Fluid Dynamics 7

2.1.1 Fluids and Gases 7

2.2 Governing Equations 8

2.2.1 Navier Stokes Equations 8

2.2.2 Effect of Compressibility 10

2.2.3 Equations for Reacting Flows 11

2.2.4 The Pressure Implicit with Split Operator (PISO) Algorithm . . 13

2.3 Turbulence Modelling 14

2.3.1 Turbulence 14

2.3.2 Modelling approaches 15

2.3.3 Turbulence Closure 18

2.4 Chemical Modelling 20

2.4.1 Chemical Kinetics 20

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.4.2 Plug Flow Reactor . . . 23
2.5 Turbulence-Chemistry Interaction . . . 25
2.5.1 Eddy Dissipation Concept . . . 25
2.6 ODE Theory . . . 28

3 Artificial Neural Networks 34
3.1 Machine Learning . . . 34
3.1.1 General Remarks . . . 34
3.1.2 Training Data . . . 35
3.1.3 Model Accuracy . . . 36
3.2 Artificial Neural Networks . . . 37
3.2.1 General Remarks: The Perceptron . . . 37
3.2.2 Feedforward Neural Networks . . . 39
3.2.3 Activation Functions . . . 41
3.2.4 Loss Function . . . 43
3.2.5 Training via Backpropagation . . . 44
3.3 Approximation Properties of Neural Networks . . . 49
3.3.1 Theoretical Results . . . 49

4 Methodology 53
4.1 General Remarks . . . 53
4.2 Preliminaries . . . 55
4.2.1 Computational Tools . . . 55
4.2.2 Hardware . . . 57
4.3 Sample Data Generation . . . 58
4.3.1 Random Initial Conditions . . . 60
4.3.2 Evolution of Chemical Species . . . 73
4.4 Neural Network Training and Evaluation . . . 74
4.4.1 Architecture Selection . . . 74
4.4.2 Training and Testing . . . 76
4.5 Turbulent Combustion Coupling . . . 79

5 Results and Discussion 81
5.1 Training Results . . . 81
5.1.1 Step size: 1×10^{-4} . . . 82

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.1.2 Step size: 1×10^{-5} 88
5.1.3 Step size: 1×10^{-6} 93
5.2 Selected Architectures 97
5.2.1 Network for step 1×10^{-4} 97
5.2.2 Network for step 1×10^{-5} 100
5.3 Simulation Results 105
5.3.1 Radial profiles at x/d=1 105
5.3.2 Radial profiles at x/d=2 107
5.3.3 Radial profiles at x/d=15 109
5.3.4 Radial profiles at x/d=30 111
5.3.5 Radial profiles at x/d=60 113

6 Conclusions and Outlook **115**
6.1 Conclusions 115
6.2 Outlook 118

References **120**

A Appendix **133**
A.1 Additional Results 133
A.1.1 Radial profiles at x/d=3 133
A.1.2 Radial profiles at x/d=7.5 135
A.1.3 Radial profiles at x/d=45 137

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Affidavit

I confirm, that going to press of this thesis needs the confirmation of the examination comittee.

I, Marlon Danilo Cabrera Ormaza, declare in lieu of oath

1. that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.
2. that this work is original and has not been submitted elsewhere for any examination nor is it currently under consideration for a thesis elsewhere.

Date

Signature

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Acknowledgements

This work would not have been possible without the support and contribution of various individuals. Thus is my desire to thank all of them:

- To my thesis adviser Michael Harasek, for accepting me into his workgroup, allowing me to work into his supervision, and guidance during the workgroup's weekly meetings.
- To Eva-Maria Wartha, for her constant support during the whole research work, her constant questions and suggestions.
- To all members of the Computational Fluid Dynamics Research Divison for their weekly feedback, with special mention to Markus Bösenhofer who was constantly interested in my work.
- To all the professor under which I had the pleasure to study at TU Wien, and the knowledge the imparted into me. Special thanks to professor Anton Arnold, coordinator of the MathMods program at TU for his constan help with any issue that ocurred during my studies.
- To professor Bruno Rubino of the University of L'Aquila, and all members and teaching staff involved in the MathMods program. I am thankful to have had the opportunity to be a part of this program.
- To the government of the Abruzzo Region, the University of L'Aquila, and the people of Italy, for co-financing my education, and for those many ones who became my friends.
- To my sisters Verónica and Victoria, to my dear brother José Manuel, and my parents Verónica and Marlon. Without their emotional and economical support through these last two years this would have been an impossible task.
- To all those friends that impacted significantly my life: Edison, Xavier, Jaime my old friends back in Ecuador; Mohamed, Daniel, Nina, Juan Pablo, Jobin and the friends I left in Italy; Andrés, Julien, and Gaia who became my family here in Vienna.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Abstract

Combustion reactions abound in nature and are of extreme importance in today's human civilization, with many critical industrial activities depending -or relying heavily on them. Metal refining, energy production, transportation, and even space exploration are among the myriad of ventures that are contingent on combustion processes.

Simulations using Computational Fluid Dynamics (CFD) allow for the optimization of combustion related processes by enabling the testing of different scnenarios, without the need of setting up costly experimental procedures. Reactive turbulent flows are an application that can benefit greatly from the use of well calibrated computational models, however, there is an added complexity by needing to account for the interaction between turbulence and chemical reactions. The Eddy Dissipation Concept (EDC) is chosen as a combustion model, since it offers an acceptable balance between accuracy and computational cost, and can be applied in combination with any turbulence model. Unfortunately, its use requires the solution of a system of stiff ordinary differential equations for the calculation of the reaction rates of chemical species. This can become computationally expensive as the number of species and reactions grows.

Though proposed in the 1950's, the use of Artificial Neural Network (ANN) have exploded in popularity over the last decade with use over a broad range of fields like autonomous driving, image recognition and natural language processing. More interestingly, mathematical results from the early 1990's show their ability to approximate any continuous function under the right conditions. The applicability of ANN as a replacement of a numerical integrator used for solving the Ordinary Differential Equation (ODE) system resulting from the EDC in turbulent reacting flows was tested. Several ANN architectures were trained by an appropiately generated dataset, and then tested against an ODE numerical solver. The error propagation and computational speed-up were studied experimentally, and limits were determined for the complexity of a network architecture in terms of its speed-up.

ANN's with the best performance were coupled with OpenFOAM® in order to

simulate the Sandia Flame D, a piloted methane-air jet flame developed by the Sydney University and compare it with the wide available experimetal data from the Sandia National Laboratories. The results reveal that ANN provide an accurate substitute of an ODE solver, however still with room for improvement, provided better training can be achieved under the limits of complexity that were also obtained.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Figures

2.1 PISO algorithm in OpenFOAM® 14
2.2 Schematics of a Plug Flow Reactor. Adapted from Schmidt [59] 23
2.3 Species transfer between fine structures and surrounding volume 25

3.1 Examples of underfit, proper fit and overfit 37
3.2 ReLU and LeakyReLU functions 42

4.1 Proposed Methodology 54
4.2 Training Data Generating Procedure 59
4.3 Training Data Structure 60
4.4 Distribution of CH_4 initial conditions 68
4.5 Distribution of CO initial conditions 69
4.6 Distribution of CO_2 initial conditions 69
4.7 Distribution of H_2 initial conditions 70
4.8 Distribution of H_2O initial conditions 71
4.9 Distribution of N_2 initial conditions 72
4.10 Distribution of O_2 initial conditions 72
4.11 Data flow for 5-layer 64-neuron feed forward network. 76
4.12 Difference of prediction between regular ANN and those used as integrators 78
4.13 Sandia Flame D geometry 80

5.1 Speedup for step size 1×10^{-4} 83
5.2 Mean Squared Error for step size 1×10^{-4} 84
5.3 Error deviation for step size 1×10^{-4} 84
5.4 Validation Loss for step size 1×10^{-4} 85
5.5 Error growth for step size 1×10^{-4} 85
5.6 Speedup for step size 1×10^{-5} 88
5.7 Mean Squared Error for step size 1×10^{-5} 89
5.8 Error deviation for step size 1×10^{-5} 90
5.9 Validation Loss for step size 1×10^{-5} 90
5.10 Error growth for step size 1×10^{-5} 91
5.11 Speedup for step size 1×10^{-6} 93

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.12 Mean Squared Error for step size 1×10^{-6} 94
5.13 Error deviation for step size 1×10^{-6} 94
5.14 Validation Loss for step size 1×10^{-6} 95
5.15 Error growth for step size 1×10^{-6} 95
5.16 Initial condition I evaluation 98
5.17 Initial condition I evaluation (Extended) 98
5.18 Initial condition III evaluation 99
5.19 Error propagation for 1000 samples 99
5.20 MSE vs Initial Temperature for 1000 samples 100
5.21 Initial condition I evaluation 101
5.22 Initial condition I evaluation (Extended) 102
5.23 Initial condition II evaluation 102
5.24 Initial condition III evaluation 103
5.25 Error propagation for 1000 samples 103
5.26 MSE vs Initial Temperature for 1000 samples 104
5.27 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 1$ 105
5.28 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 1$ . . 106
5.29 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 2$ 107
5.30 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 2$ . . 108
5.31 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 15$ 109
5.32 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 15$. 110
5.33 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 30$ 111
5.34 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 30$. 112
5.35 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 60$ 113
5.36 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 60$. 114

A.1 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 3$ 133
A.2 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 3$ . . 134
A.3 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 7.5$ 135
A.4 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 7.5$. 136
A.5 Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 45$ 137
A.6 Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 45$. 138

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Tables

2.1 Standard coefficients for the $k - \epsilon$ model 19

4.1 Hardware Specifications 58
4.2 Reaction Mechanism Thermodynamical Properties 61
4.3 Parameters of random samples 64
4.4 Fuel and Oxydizer Elemental Mass Fractions 65
4.5 Summary of Initial Conditions generated via Least Squares 66
4.6 Summary of Initial Conditions generated via Non-negative Least Squares 66
4.7 Initial conditions chosen for visualization purposes 78

5.1 Dataset parameters 81
5.2 Training results for step size 1×10^{-4} 87
5.3 Training results for step size 1×10^{-5} 92
5.4 Training results for step size 1×10^{-6} 96
5.5 Parameters of Neural Network for Step 1×10^{-4} 97
5.6 Parameters of Neural Network for Step 1×10^{-5} 101

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Acronyms

ANN Artificial Neural Network. iii–v, 2–6, 30, 32, 53, 58, 68, 69, 74, 76, 90, 113–117

API Application Programming Interface. 57

BDF Backwards Differentiation Formulae. 31, 32, 113

CFD Computational Fluid Dynamics. iii, 1–4, 28, 57

CPU Central Processing Unit. 2, 4, 56, 58

CUDA Compute Unified Device Architecture. 58

DI Direct Integration. 2

DNS Direct Numerical Simulation. 1, 4, 5, 16

EDC Eddy Dissipation Concept. iii, 2, 3, 5, 25, 32, 76, 113

EDM Eddy Mixing Model. 4

FVM Finite Volumen Method. 13

GPU Graphics Processing Unit. 58, 116

ISAT In-situ Adaptive Tabulation. 2

IVP Initial Value Problem. 28–31

LEM Linear Eddy Model. 5

LES Large Eddy Simulation. 1, 3–5

LS Least Squares Method. 67, 68, 113

ML Machine Learning. 34

MSE Mean Squared Error. 4, 5, 73, 78–80, 82, 85, 88, 89, 91, 114, 115

NNLS Non-negative Least Squares Method. 67, 68, 113

NSE Navier-Stokes Equations. 1, 3, 9, 11, 13, 15–17

ODE Ordinary Differential Equation. iii, iv, 1–5, 28, 32, 58, 59, 69, 76, 78, 102, 104, 106, 108, 113, 116

PDF Probability Density Function. 4, 5

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

PFR Plug Flow Reactor. 23, 24, 28, 32, 56, 113

PISO Pressure Implicit with Split Operator. 13

RAM Random Access Memory. 58

RANS Reynolds Averaged Navier-Stokes Equations. 1, 3, 5, 16

SGD Stochastic Gradient Descent. 45, 47

VODE Variable-coefficient Ordinary Differential Equation Solver. 32, 70, 77, 78, 82, 88, 90, 113, 115

w.r.t. with respect to. 10, 29, 30, 45, 46, 48

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

List of Symbols

Physics Constants

k_b	Boltzmann constant	$1.380649 \times 10^{-23}\,\mathrm{J \cdot K^{-1}}$
N_a	Avogadro constant	$6.02214076 \times 10^{23}\,\mathrm{mol^{-1}}$
R	Ideal Gas constant	$8.314462618\,\mathrm{J \cdot K^{-1} \cdot mol^{-1}}$

Chemical Species

C	Elemental Carbon
CH_4	Methane
CO	Carbon Monoxide
CO_2	Carbon Dioxide
H	ElementalHydrogen
H_2	Molecular Hydrogen
H_2O	Water
N	Elemental Nitrogen
N_2	Molecular Nitrogen
O	Elemental Oxygen
O_2	Molecular Oxygen

Fluid Dynamics

$-\overline{\rho}\widetilde{u_j'' h''}$	Enthalpy Turbulent Flux	$\mathrm{J \cdot kg \cdot m^{-2} \cdot s^{-1}}$
$-\overline{\rho}\widetilde{u_j'' u_i''}$	Reynolds Stresses	$\mathrm{kg \cdot m^{-1} \cdot s^{-2}}$
$-\overline{\rho}\widetilde{u_j'' Y_l''}$	Species Turbulent Flux	$\mathrm{kg \cdot m^{-2} \cdot s^{-1}}$
δ_{ij}	Kronecker's Delta	
$\dot{\omega}_T$	Combustion heat release	$\mathrm{W \cdot m^3}$
ϵ	Energy Dissipation Rate	$\mathrm{m^2 \cdot s^{-3}}$
η	Kolmogorov Length Scale	m
λ_T	Thermal Conductivity	$\mathrm{W \cdot m^{-1} \cdot K^{-1}}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

τ	Deviatoric Stress Tensor	$\mathrm{N \cdot m^{-2}}$
μ	Dynamic Viscosity	$\mathrm{Pa \cdot s}$
μ_t	Turbulent Viscosity	$\mathrm{Pa \cdot s}$
ν	Kinematic Viscosity	$\mathrm{m^2 \cdot s^{-1}}$
$\overline{\dot{\omega}}_l$	Mean Reaction Rate of Species l	$\mathrm{kg \cdot m^3 \cdot s^{-1}}$
$\overline{\Psi}$	Expected value of a scalar quantity	
Ψ	Scalar quantity	
Ψ'	Fluctuating part of a scalar quantity related to an expected value	
Ψ''	Fluctuating part of a scalar quantity related to a mass average	
ρ	Density	$\mathrm{kg \cdot m^{-3}}$
σ_ϵ	Coefficient for $k-\epsilon$ model	1.30
σ_k	Coefficient for $k-\epsilon$ model	1.00
σ_t	Coefficient to model turbulent fluxes	0.7
τ_η	Kolmogorov Time Scale	s
$\mathbf{q}$	Heat Flux	$\mathrm{W \cdot m^{-2}}$
$\mathbf{u}$	Velocity	$\mathrm{m \cdot s^{-1}}$
$\widetilde{\Psi}$	Mass average of a scalar quantity	
c	Speed of sound in a given medium	$\mathrm{m \cdot s^{-1}}$
C_{ϵ_1}	Coefficient for $k-\epsilon$ model	1.44
C_{ϵ_2}	Coefficient for $k-\epsilon$ model	1.92
C_μ	Coefficient for $k-\epsilon$ model	0.09
D	Ordinary Mass Diffusion Coefficient	$\mathrm{m^2 \cdot s^{-1}}$
e	Internal Energy	J
g	Volume Force	$\mathrm{N \cdot m^{-3}}$
h	Specific Enthalpy	$\mathrm{J \cdot kg^{-1}}$
k_e	Kinetic Energy	J
L	Characteristic Length	m
M	Mach Number	

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Re	Reynolds Number	
S_{ji}	Mean Strain Tensor	
t	Time i	s
u	Local Flow Velocity	$\mathrm{m \cdot s^{-1}}$
u_η	Kolmogorov Velocity Scale	$\mathrm{m \cdot s^{-1}}$

Chemical Kinetics

α_i	Order of reaction with respect to species i	
$\dot{\omega}_i$	Reaction Rate of Species i	$\mathrm{mol \cdot m^3 \cdot s^{-1}}$
$[\mathcal{M}_i]$	Molar Concentration of Chemical Species i in a reaction	$\mathrm{mol \cdot m^{-3}}$
$\mathcal{M}_i$	Chemical Species i in a reaction	
$\overline{\rho}$	Mean Gas Density	$\mathrm{kg \cdot m^{-3}}$
$\overline{W}$	Mean Molar Mass	$\mathrm{kg \cdot mol^{-1}}$
υ_i	Coefficient for NASA Polynomials	
A	Pre-Exponential Factor	
a_i	Stoichometric Coefficient of Species i in a left side of a reaction	
A_s	Cross Section of PFR	$\mathrm{m^2}$
b_i	Stoichometric Coefficient of Species i in a right side of a reaction	
c_P	Specific Heat Capacity	$\mathrm{J \cdot K^{-1} \cdot kg^{-1}}$
c_P^0	Molar Heat Capacity at reference pressure P_0	$\mathrm{J \cdot mol^{-1} \cdot K^{-1}}$
E_a	Activation Energy	$\mathrm{J \cdot kg^{-1}}$
F_i	Molar Flow of species i through PFR	$\mathrm{mol \cdot s^{-1}}$
h	Sensible Enthalpy	J
h_0	Molar Enthalpy at reference pressure P_0	$\mathrm{J \cdot mol^{-1}}$
h_i	Sensible Enthalpy of Species i	$\mathrm{J \cdot kg^{-1}}$
h_i°	Enthalpy of Formation of Species i	$\mathrm{J \cdot mol^{-1}}$
k	Reaction Rate Constant	
K_j^c	Equilibrium Constant of species j	
k_{bj}	Backwards Reaction Rate Constant for species i	

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

k_{fj}	Forward Reaction Rate Constant for species i	
m_i	Mass of Species i	kg
n	Number of Moles	mol
p	Pressure	Pa
P_0	Reference Pressure	Pa
$p_\circ$	Standard Pressure at one atmosphere	101.325×10^3Pa
R_j	Gas Constant for Species j	$\mathrm{J \cdot K^{-1} \cdot mol^{-1}}$
s_0	Molar Entropy at reference pressure P_0	$\mathrm{J \cdot mol^{-1} \cdot K^{-1}}$
s_j	Entropy of species j	$\mathrm{J \cdot K^{-1}}$
T	Temperature	K
T_0	Reference Temperature	K
u	Axial Flow Velocity of PFR	$\mathrm{m \cdot s^{-1}}$
V	Volume	$\mathrm{m^3}$
v	Volumetric Flow of PFR	$\mathrm{m^3 \cdot s^{-1}}$
X_i	Mole Fraction of Species i	
Y_i	Mass Fraction of Species i	

Eddy Dissipation Concept

χ	Fraction of the fine structures	
$\dot{m}$	Mass Transfer per unit of mass of fluid	$\mathrm{s^{-1}}$
$\dot{m}^*$	Mass transfer rate from surrounding fluids to fine structures	$\mathrm{kg \cdot s^{-1}}$
γ^*	Volume fraction of the fine Structures	$\mathrm{m^3}$
$\overline{c}_i^*$	Mean Concentration of species i inside the control volume	$\mathrm{mol \cdot m^3}$
$\overline{h_i}$	Mean Enthalpy of Species i in control volume	$\mathrm{J \cdot kg^{-1}}$
$\overline{Y}_k^\circ$	Mean Mass Fraction of Species k in control volume	
τ^*	Residence time inside mass fraction	s
c_i^*	Concentration of species i inside the fine structures	$\mathrm{mol \cdot m^3}$
c_i°	Concentration of species i in the surrounding fluid	$\mathrm{mol \cdot m^3}$
h_i^*	Enthalpy of Species i in fine structures	$\mathrm{J \cdot kg^{-1}}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

L^*	Characteristic Lenght of the fine structures	m
q^*	Energy stored inside the fine structures	
u^*	Characteristic Velocity of the fine structures	$\mathrm{m} \cdot \mathrm{s}^{-1}$
Y_k^*	Mass Fraction of Species k in fine Structures	
Y_k°	Mass Fraction of Species k in fine Surrounding Fluid	

ODE Theory

α_j	Coefficient of multi-step method
β_j	Coefficient of multi-step method
$\Delta u_l^{(i)}$	Correction of $u_l^{(i)}$ at iteration i
$\dot{\mathbf{U}}$	Derivative of solution of target IVP
$\forall$	For all
$\hat{q}$	Characteristic polynomial of multi-step method
λ	Eigenvalues of Jacobian Matrix
$\mathbb{C}$	Set of Complex Numbers
$\mathbb{I}$	Interval
$\mathbb{I}_0$	Interval containing x_0
$\mathbb{R}$	Set of Real Numbers
$\mathcal{D}$	Domain of stability of multi-step method
Ω	Domain
ϕ	Iteration function
$\mathbf{C}$	Set of Continuous Functions
$\mathbf{C}^1$	Set of Continuous Differentiable Functions
$\mathbf{f}$	RHS of target IVP
$\mathbf{U}$	Solution of target IVP
ζ	Characteristic polynomial root of multi-step method
$\top$	Traspose operator
f	Right Hand Side Function of IVP
I	Identity Matrix

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

J	Jacobian Matrix	
O_p	Order of numerical method	
$Re(\lambda)$	Real part of eigenvalue λ	
t_l	BDF time variable at step l	
u_l	BDF approximation at step l	
$u_l^{(i)}$	Newton-Raphson iteration i of approximation u_l	
x_0	Initial point coordinate	
y	Solution of IVP	
y'	Derivative of solution of IVP	
y_0	Initial point coordinate	
z	Product of time step Δt and a scalar λ	

Neural Networks

β_1	Adam model parameter	0.9
β_2	Adam model parameter	0.999
$\delta^{(l)}$	Derivative of a loss function E w.r.t. weighted input $z^{(l)}$ of layer l	
ϵ°	Adam model parameter	10^{-8}
η_L	Learning rate	
$\hat{m}_\theta^t$	Exponential moving average of the gradient of a loss function E	
$\hat{v}_\theta^t$	Exponential moving average of the gradient of a loss function E squared	
$\hat{y}_j$	Vector of predictions	
σ	Activation function	
$\mathbf{K}$	Compact set	
Θ	Activation threshold	
θ	Vector of parameters	
ξ_i	Inner value in chain rule	
$a^{(l)}$	Output of layer l	
$b^{(l)}$	Vector of biases of layer l	
E	Loss function	

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

L	Number of layers in a network	
N_b	Number of samples in batch	
N_L	Number of layers	
N_n	Number of neurons	
N_s	Number of samples	
w	Matrix or Vector of weights	
$w^{(l)}$	Matrix of weights of layer l	
x_j	Feature Vector	
y_j	Vector of labels/targets	
$z^{(l)}$	Weighted input of layer l	

Training Data Samples

Δt	Time step	
δ_s	Training/Validation data split	s
μ_{ij}	Mass fraction of element j in species i	
$\sim U$	Uniformly distributed	
ξ	Mixture fraction	
ξ_{max}	Max Mixture fraction	
ξ_{min}	Minimum Mixture fraction	
a_{ij}	Atoms of element j in species i	
E_j	Atomic mass of element j	$\mathrm{kg} \cdot \mathrm{mol}^{-1}$
N	Number of chemical species in a mechanism	
T_{max}	Max temperature	K
T_{min}	Minimum temperature	K
T_n	Normalized temperature	
Z_i	Mass fraction of element i	

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1 Introduction

1.1 Motivation and Problem Statement

Although computational resources are still increasing, and more and more simulations are enabled, the computational time can still be prohibitive for reactive flow simulations using complex kinetic mechanisms. As the number of species in a reaction increases, so does the number of equations that must be solved. For example, GRI-Mech 3.0 an optimized mechanism designed to model natural gas combustion, developed by the University of California, Berkeley involves 53 species and 325 elementary reactions [1]. Current hardware developments make this situation unlikely to change in the near future, since processors are approaching the point where heat dissipation will cripple the increase in operational speed, and even in the face of increasing parallelization of algorithms, memory speed will likely continue being a huge bottleneck in computation speed [2, 3].

The simulation of turbulent flows requires resolving the Navier-Stokes Equations (NSE) over a very wide range of time and length scales. When coupled with chemistry kinetics, generally, the simulation of turbulent combustion through Computational Fluid Dynamics (CFD) is undertaken by one of the following three approaches [4]: Reynolds Averaged Navier-Stokes Equations (RANS) equations, Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). However DNS is as of now, computationally prohibitive even for non-reacting flows, and despite the fact that LES and RANS are able to reduce computational cost by ignoring the smallest length scales [5] (via a low-pass filtering of the NSE) and by solving for the mean values among all quantities, -respectively-, the representation of chemical kinetics is still a problem for all of them.

This representation requires precise information on the time rate of change of all species involved in the reaction. To compute all these changes in species composition, an ODE system must be solved. Taking the example from before, solving the 53 necessary conservation equations, as well as the 325 reactions, might not be possible in a reasonable time. Things are further complicated when the difference in the reaction

rates of the involved species varies greatly as the resulting ODE system becomes stiff [6], further straining the available computational resources by reducing the allowable time integration step.

To couple the chemistry and turbulence mechanisms, EDC, proposed -and later improved- by Magnussen [7, 8] will be used. This model attempts to incorporate the significance of fine structures in a turbulent reacting flow in which combustion is important. In order to solve the time evolution of the different species at the fine scale level, the EDC uses Direct Integration (DI) with or without In-situ Adaptive Tabulation (ISAT) which are burdened by large Central Processing Unit (CPU) and memory requirements respectively. According to Christo [9], an optimal numerical scheme for the chemistry representation must offer the advantage of the aforementioned method, at low CPU and memory cost.

There are many different strategies to overcome this problem. Tabulation can be used, or the mechanisms can be reduced to skeletal, i.e. simpler, mechanisms. Another way is to use ANN's trained by kinetic data and use it to predict the concentration changes in combustion simulations. A lot of research is currently be done on CFD fields using Machine Learning techniques, including among others, using ANN to find more efficient discretizations, and, as mentioned before, predict solutions of a given model.

ANN have proven effective in solving many engineering problems, and though still not fully understood, there are proven mathematical results that back up their power for approximating any - at least continuous- function (under certain assumptions). These results include the Kolmogorov-Arnold [10] representation theorem, and the Universal Approximation theorem [11]. However, though powerful, the proofs given for these theorems are not constructive, and thus don't offer any algorithm to build an efficient network, and of utmost importance, no explicit estimates are provided for the number of units in a hidden layer [12].

Since, the main performance limiting factor when modeling turbulent reactive flows is related to the computational effort used in solving the chemical reactions between all the involved species, a proposed plan of action is to devise an ANN capable of approximating the chemical reactions reliably, so as to couple it later with traditional non reac-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

tive flows, where the NSE are solved with a common modeling technique (RANS/LES).

1.2 Aim of the Work and Objectives

Research on ANNs currently used in CFD, especially in reactive flow techniques should be studied. An ANN shall be trained and tested for a chemical mechanism for methane combustion. The application of the ANN for CFD simulations of flames should be possible in the end. Based upon these, this work is guided by the following approach:

1. Literature review of past works related to the coupling of ANN's to reactive flow simulations.

2. Detailing of the theoretical background pertaining to reactive flows and neural networks.

3. Implementation of a framework to create training data for a neural network based on the requirements of this work.

4. Training and selecting diverse variants of neural networks in order to solve the chemical kinetics needed for the EDC.

5. Proposal of a testing procedure in order to account for error propagation when using an ANN as an alternative to an ODE integrator.

6. Development of a Python library designed to handle the diverse operations and procedures needed for sample data generation as well as ANN training and validation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

7. Coupling of the best performing ANN's with a CFD solver in order to compare performance with known results.

1.3 Literature Review

- In 1996 Christo et al. [9] implemented the use of ANN's for a turbulent jet diffusion flame using a H_2/CO_2 mechanism with three elementary reactions, modelled with a joint Probability Density Function (PDF) transport equation. It was found ANN's only become effective for reaction mechanisms with more than three involved reactions. A 2-layer 2-neuron feedforward network was utilized to solve the chemical reactions with Tanh activation functions. Only training error is reported, computed by Mean Squared Error (MSE) .

- In 1998 Blasco et al. [13] investigated the use of ANN as replacement of ODE solvers in laminar reacting flows solved by DNS . The simulation considered a reduced methane-air reaction mechanism consisting in four reactions and eight chemical species ($CH_4, O_2, CO_2, H_2O, CO, H_2, H$ and N_2). ANN's were trained to predict mass fractions at three different time steps:$1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}$, as well as for temperature and density. The ANN's used a 8-input 7-output feedforward architecture with sigmoid activation functions and two hidden layers (one for temperature and density) with 20 neurons each. The test error is evaluated similarly to the training error, by comparing the network output for input i to the target value for input i via MSE, and then reported as a percentage with the mean target value. No study of the error propagation is presented, or mention of using predicted values as inputs. Memory use and CPU time were reduced 831 and 165 times respectively. Blasco notices that the test error grows as the time step does, posibbly because of the increase in complexity of the relation between initial and final species mass fractions.

- Sen and Menon in 2009 [14] evaluated the applicability of an ANN chemistry integrator in reactive flows simulated by LES with the Eddy Mixing Model (Eddy Mixing Model (EDM)). A 10-step, 14-species mechanism designed for the com-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

bustion of syngas is used. Data was generated by DNS of flows with different levels of turbulence. ANN's with 2 layers (10 neurons/5 neurons) and 3 layers(10/5/4) were trained to predict the chemistry at the small scales. While high in accuracy, the method required the added computational cost related to DNS .

- Kempf [15] in 2005 and Emami [16] in 2012 implemented ANN's to predict species mass fractions of H_2O, H_2, CO_2, CO, OH and temperature, by taking as inputs the mixture fraction, the mixture fraction variance, and scalar dissipation rate, in LES and RANS simulations respectively with a laminar flamelet model and using the GRI-MECH 3.0 mechanism.

- In 2014 Porumbel et al. [17] coupled an ANN with an existing Linear Eddy Model (LEM) -LES algorithm. The ANN were used to estimate chemical source terms and to replace flamelet tables. The temperature total range was divided in 14, with a network trained for each subdivision, as well as for each of the involved 12 species, testing for 2, 4, and 8 neurons each with a Tanh activation. Great reduction in computing time was found when compared to a 1D simulation using ODE integrators.

- In 2017 Laubscher [18, 19] tested an 5-step 7-species mechanism to replace the ODE integrator for the species mass fractions needed for the EDC. An architecture with 8 inputs (species plus temperature), 7 hidden layers, and 7 outputs for the chemical species was implemented with Tanh activation functions (except in the output). Reductions in computing time and memory usage were reported. Test error was reported through MSE, but no study of error propagation.

- Some interesting work is presented by many other authors as recent as of this year (2020). Most of said work is related to the use of ANN's in turbulent combustion simulations but with a PDF-LES approach. These includes the papers introduced by: Ranade and Echecki in 2019 [20] which provides a detailed framework for data based closure models; Lapeyre et al. [21] (2019) which trains convolutional ANN's to estimate small scale reactions, Zhang et al. [22] (2020)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

which combines ANN's with flamelet generated manifold (and whose accompanying code has being made publicly available), and Luo et al. [23] which replaces completely a numerical flow solver by ANN's.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2 Reactive Flows

Big whirls have little whirls,
That feed on their velocity; And
little whirls have lesser whirls,
And so on to viscosity

Lewis Fry Richardson,1922

2.1 Fluid Dynamics

2.1.1 Fluids and Gases

A ***fluid*** is any type of substance that lacks resistance to the application of external shear forces, with any of such forces causing deformation. Unlike solid which are usually studied individually, fluid particles are often considered as a continuum [24].

The flow of a fluid is caused by the influence of external forces. Their behavior is often similar, but notably, their ***macroscopic*** properties might differ broadly. The study of this properties is necessary in order to anlyze their motion. Properties of all fluids like ***density*** and ***viscosity*** are themselves functions of other ***thermodynamical*** variables (such as pressure or temperature).

Unlike liquids, which can for most cases be treated as imcompressible fluids whose density remains constant, a ***state function*** is needed to establish relations between the internal system variables of a gas.

The ***Ideal Gas Law*** provides such relations for the *density, composition, pressure and temperature* of a single-species ideal gas. It is given by:

$$pV = nN_A k_b T = nRT \tag{2.1}$$

where p is the gas pressure, V is the gas volume, n is the number density, T is the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

absolute temperature, R is the universal gas constant, k_b is the Boltzmann constant, and N_A is the Avogadro constant.

Kee et al. [25] refer to a state equation for a multispecies gas in terms of its mean ***molar mass*** $\overline{W}$:

$$\overline{\rho} = \frac{p}{RT}\overline{W} \tag{2.2}$$

here $\overline{\rho}$ is the gas mean density.

It can be further decomposed via each individual species' molar masses W_i and either their corresponding ***mass fractions*** Y_i or ***mole fractions*** X_i:

$$\overline{W} = \sum_i X_i W_i = \left(\sum_i \frac{Y_i}{W_i}\right)^{-1} \tag{2.3}$$

Remark 1 *A chemical species i in a mixture is characterized by its mass fraction Y_i defined through:*

$$Y_i = \frac{m_i}{\sum m_i} \tag{2.4}$$

where m_i is the mass of species i.

2.2 Governing Equations

2.2.1 Navier Stokes Equations

The chief mathematical expresions in the field of fluid mechanics are prescribed by the ***Navier-Stokes equations***. Developed by coincident and in an independent manner at least five different times[26] by Claude-Louis Navier (1822), Augustin-Louis Cauchy

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

(1832), Siméon Denis Poisson (1829), Jean Claude Saint-Venant (1837), and George Gabriel Stokes (1845), these set of non-linear partial differential equations are the simplest models capable of describing the motion of a viscous fluid, and were derived under the assumptions of said fluids being ***Newtonian***, i.e. that the relation between ***viscous stresses*** and ***strain rates*** is linear[27]. For a non-reactive compressible flow they are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{\mathbf{u}}) = 0 \tag{2.5}$$

Continuity eq.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \tau) = -\nabla p + \rho g \tag{2.6}$$

Momentum eq.

$$\frac{\partial}{\partial t}\left[\rho\left(\frac{|\mathbf{u}|^2}{2} + e\right)\right] + \nabla \cdot \left[\rho \mathbf{u}\left(\frac{|\mathbf{u}|^2}{2} + e\right) + \mathbf{q} + (p - \tau) \cdot \mathbf{u}\right] = \rho f \cdot \mathbf{u} \tag{2.7}$$

Energy eq.

Here τ is the deviatoric stress tensor, g represents a volume force and $\mathbf{q}$ measures the heat flux.

The three Navier-Stokes equations each represent a conservation law for physical system, namely, conservation of ***mass, momentum and energy***. The full expresions, in absence of any simplifications must be solved simultaneously and involve three unknowns: temperature T, pressure p and the vector velocity field $\mathbf{u}$. An additional four thermodynamical quantities are present in the complete equations: density ρ, viscosity μ, thermal conductivity λ_T and one of either the internal energy e or the enthalpy h. These four variables depend on both pressure and temperature, and knowledge of their relations is needed in order to give closure to the equations [28]. Because all of these variables must be solved for at the same time, solving the NSE has some complications. These complications and how one could deal with them will be disscussed in Subsection 2.2.4.

According to Poinsot and Veynante [29] reactive flows add a layer of complexity to the NSE . Continuity equations must be solved for each chemical species involved,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

and with each of them reacting at rate dependent on the mixing and temperature at each point; instabilities may also arise due to high changes in heat capacity across the geometry.

2.2.2 Effect of Compressibility

An ***incompressible flow*** refers to a flow with velocity $\mathbf{u}$ whose fluid density ρ remains constant inside a volume element dV. Because of ***mass conservation***[30] this is mathematically equivalent to the so called ***Incompressibility Condition***:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.8}$$

Notwithstanding the fact that all real fluids -such as water- are up to a certain amount compressible [31], most flows can be considered imcompressible as the difference in compressiblity between a compressible and a slightly compressible one can vary in several orders of magnitude [32].

The ***Mach number***, defined as:

$$M = \frac{u}{c}$$

or the ratio between a flow's local velocity u with respect to (w.r.t.) a boundary and the speed of sound c in the medium provides a useful approach to determine whether a fluid is incompressible in practice. Anderson [33] suggest that for $M < 0.3$, changes in density are smaller than 5%, a condition that makes a fluid essentially non-compressible.

Braack and Rannacher [34] describe most reactive flow situations -like those of burners- as having a small Mach number, thus making them ***hydrodynamically incompressible*** but yet, because of the significant effect temperature gradients posses in gases still suffering from compression and expansion processes. For this cases, special

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

considerations are needed when using the ***compressible Navier-Stokes equations***, as instabilities may appear otherwise.

Thus, despite the intended simulations having a low Mach number, a compressible flow will be considered for this work in order to account for the effects of the temperature changes. The low Mach number condition however is still useful as with it, it can be assumed that the flow density is only dependant on temperature changes [29].

2.2.3 Equations for Reacting Flows

Jones and Whitelaw [35], and Poinsot and Veynante [29] provide a version of the NSE for reacting gas flames in absence of radiation, external body forces and heat sources, and constant pressure:

Mass Conservation

$$\frac{\partial \rho}{\partial t} + \frac{\partial u_i}{\partial x_j} = 0 \tag{2.9}$$

Here u represents the flow's velocity and ρ the gas' density which is computed by the law of ideal gases as seen in Equation 2.2.

Momentum Conservation

$$\frac{\partial \rho}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.10}$$

Here τ_{ij} is the viscous tensor given by:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \tag{2.11}$$

where μ is the dynamic viscosity and δ_{ij} is the Kronecker delta.

Species Conservation

$$\rho \frac{\partial Y_l}{\partial t} + \rho u_j \frac{\partial Y_l}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\rho D \frac{\partial Y_l}{\partial x_j} \right) + \dot{\omega}_l \tag{2.12}$$

This equation is solved for every reacting chemical species. The terms Y_l and $\dot{\omega}_l$ represent the mass fraction and reaction rate of species (which is discussed later in

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Subsection 2.4.1) l respectively. D is the binary mass diffusion coefficient and can be assumed as constant for turbulent flows [36].

Enthalpy Conservation

The expression for energy conservation is given in terms of sensible enthalpies h_s (for readibility the subscript s is dropped):

$$\rho\frac{\partial h}{\partial t} + \rho u_j \frac{\partial h}{\partial x_j} = \frac{\partial p}{\partial t} + \rho u_j \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j}\left(\lambda_T \frac{\partial T}{\partial x_j} + \rho D \sum_{l=1}^{N} h_l \frac{\partial Y_l}{\partial x_j}\right) + \tau_{ij}\frac{\partial u_i}{\partial x_j} - \sum_{l=1}^{N} h_k^{\circ}\dot{\omega}_l \tag{2.13}$$

here λ_T is the thermal conductivity, h_l° is the enthalpy of formation of species l, and h_l is the sensible enthalpy of species l computed as:

$$h_l = \int_{T_0}^{T} c_{P_l}\, dT \tag{2.14}$$

c_{P_l} is the specific heat of species l at constant pressure and T_0 is a temperature of reference.

The last term corresponds to the heat release due to combustion $\dot{\omega}_T$:

$$\dot{\omega}_T = -\sum_{l=1}^{N} h_k^{\circ}\dot{\omega}_l \tag{2.15}$$

The term:

$$\rho D \sum_{l=1}^{N} h_l \frac{\partial Y_l}{\partial x_j} \tag{2.16}$$

is usually set to zero, as it is mostly negligible in comparison to $\dot{\omega}_T$ [29]. Due to the assumption of constant temperature $\frac{\partial p}{\partial t}$ is zero as well.

Thus Equation 2.13 can be simplified to:

$$\rho\frac{\partial h}{\partial t} + \rho u_j \frac{\partial h}{\partial x_j} = \rho u_j \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j}\left(\lambda_T \frac{\partial T}{\partial x_j}\right) + \tau_{ij}\frac{\partial u_i}{\partial x_j} - \sum_{l=1}^{N} h_k^{\circ}\dot{\omega}_l \tag{2.17}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.2.4 The PISO Algorithm

The ***PISO*** algorithm, short for ***Pressure Implicit with Split Operator*** is a non-iterative procedure for solving the pressure-velocity coupling of the Navier-Stokes equations in conjuction with a finite-volume discretization and a backwards difference approximation for the time derivatives [37]. It is applicable to both compressible and incompressible flows discretized via the Finite Volumen Method (FVM).

Taking as an example an isotropic incompressible fluid in absence of volume forces. The flow equations then simplify to the following:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) & = \underbrace{-\dfrac{1}{\rho}\nabla p}_{\text{Source Term}} + \underbrace{\nu \Delta \mathbf{u}}_{\text{Diffusion Term}} \\ \nabla \cdot \mathbf{u} & = 0 \end{cases} \tag{2.18}$$

Even this simpler situation serves to illustrate the complexities of the NSE . Its non-linearity means the velocity field $\mathbf{u}$ is needed to evaluate its own transport, which incidentally, also includes the other sought variable -the pressure p- as part of a source term. Thus one gets stuck in an endless loop, where the velocity field cannot be solved for until the pressure is known, but the pressure cannot be detrmined without knowledge of the speed components.

The strategy followed by schemes like PISO is guessing one of the quantities, solve for the other, and the correcting the first until no further corrections are needed. In PISO [38], the solution is then split into several steps in which the pressure is decoupled from the velocity, and that yields a set of split equation solvable via standard techniques. This method reduces the computational costs associated with iterative solutions, whilst possesing a high level of accuracy. Analysis of the procedures gives an error bound that is proportional to the powers of the time-step selected for the discretization.

Figure 2.1 gives an overview of the mechanism behind the PISO algorithm imple-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

mentation in OpenFOAM® .[1]

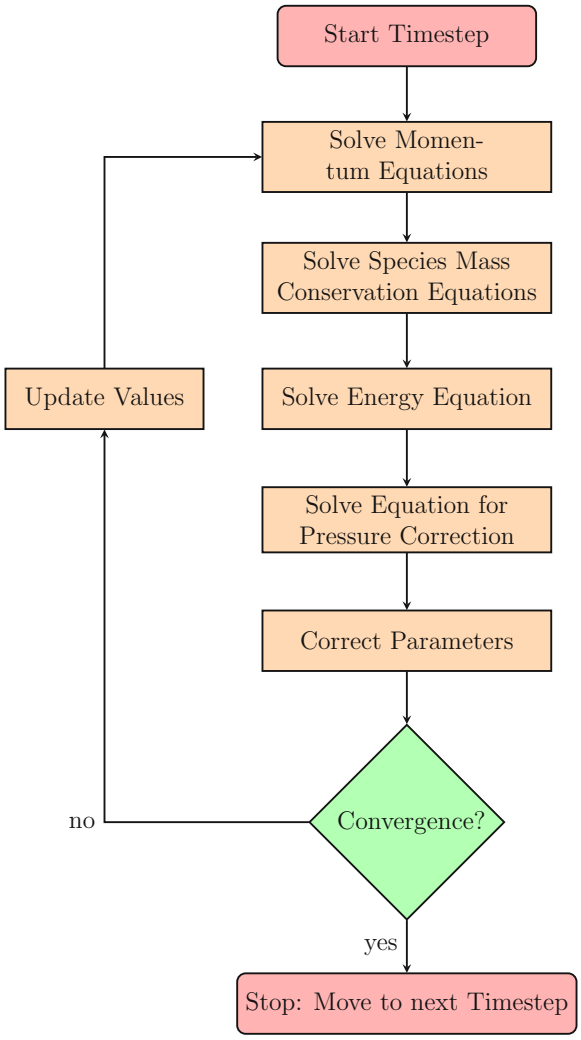


Figure 2.1: PISO algorithm in OpenFOAM®

2.3 Turbulence Modelling

2.3.1 Turbulence

The behavior of fluids is affected in numerous ways by their flow velocity. At very slow speeds for example, inertial effects on particles can be disregarded, while at higher speed it increases in importance. However, particles may still move following smooth

[1]See Subsection 4.2.1

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

trajectories within a large range of speeds and the flow is referred as ***laminar***. Once the flow attains very high speeds instability arises and the fluid behaves in a more chaotic manner that is as of yet not fully understood mathematically, and which posses in general a higher complexity when modelling.

The way turbulent flows are characterized is usually through the use of a dimensionless quantity known as the Reynolds number ***Re***, which is a measure of the ratio between the inertial and viscous forces in a fluid. It was named by Arnold Sommerfeld in honor of Osborne Reynolds and is defined as [39]:

$$Re = \frac{\rho u L}{\mu} = \frac{uL}{\nu} \tag{2.19}$$

where u is the flow velocity, ρ is the fluid density, μ is the dynamic viscosity of the fluid, ν is the kinematic viscosity, and L is a characteristic linear dimension (for example one used to derive an adimensional version of the NSE). Typically, a flow is considered turbulent when $Re > 4000$ [40].

2.3.2 Modelling approaches

One of the reasons it is difficult to model a turbulent flow is that to fully characterize, one needs to take into consideration a huge range of space and time scales [41]:from the large ones -related to the geometry of the domain and the boundaries- where energy is produced, to the smallest ones, the so called ***Kolmogorov Scales*** where energy is completely dissipated [42]. Equations 2.20, 2.21 and 2.22 provide expressions for the Kolmogorov lenght scale η, time scale τ_η, and velocity u_η in terms of the fluid's kinematic viscosity ν and the energy dissipation rate ϵ.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4} \tag{2.20}$$

Length Scale

$$\tau_\eta = \left(\frac{\nu}{\epsilon}\right)^{1/2} \tag{2.21}$$

Time Scale

$$u_\eta = (\nu\epsilon)^{1/4} \tag{2.22}$$

Velocity Scale

Three main type of approaches are usually considered for simulating turbulent flows. The first of these types is known as ***Direct Numerical Simulations*** or ***DNS***, and solves the NSE directly for all scales. This provides the greatest amount of accuracy, but is at the same time the computationally most expensive.

The other techniques involve the use of Reynolds decomposition, which decomposes a quantity Ψ in two: its expected value $\overline{\Psi}$ and a fluctuating part Ψ'.

Large Eddy Simulations or ***LES*** achieve this by applying a low-pass filtering to the NSE, while ***Reynolds Averaged Navier Stokes*** or ***RANS*** models solve a time-averaged version of the equations. In general, ***LES*** require smaller grids which adds to the computational costs.

For the purposes of this thesis ***RANS*** will be used for the numerical simulations of reacting flows. In RANS, variables are decomposed in the following way [24]:

$$\Psi(x_i, t) = \overline{\Psi}(x_i) + \Psi'(x_i, t) \tag{2.23}$$

with:

$$\overline{\Psi}(x_i) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \Psi(x_i, t)\, dt \tag{2.24}$$

However, Canuto [43] mentions how for compressible flows a ***Favre*** [44] or mass averaging is more suitable(Equation 2.25).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\Psi(x_i,t) = \widetilde{\Psi}(x_i) + \Psi''(x_i,t) = \frac{\overline{\rho\Psi}(x_i)}{\overline{\rho}(x_i)} + \Psi''(x_i,t) \tag{2.25}$$

where $\overline{\rho\Psi}$ and $\overline{\rho}$ are Reynolds averaged quantities.

Poinsot and Veynante [29] provide the NSE for a reactive flow (see Subsection 2.2.3) in Favre averaged form:

Mass Conservation:

$$\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_i}{\partial x_i} = 0 \tag{2.26}$$

Momentum Conservation:

$$\frac{\partial\overline{\rho}\widetilde{u}_i}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_i\widetilde{u}_j}{\partial x_i} + \frac{\partial\overline{p}}{\partial x_j} = \frac{\partial}{\partial x_j}\mu\left[\frac{\partial\widetilde{u}_i}{\partial x_j} + \frac{\partial\widetilde{u}_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial\widetilde{u}_i}{x_i}\right] + \frac{\partial}{\partial x_i}(-\widetilde{\overline{\rho}u_j''u_i''}) \tag{2.27}$$

The term $(-\widetilde{\overline{\rho}u_j''u_i''})$ is a new set of quantities known as ***Reynolds Stresses*** which need additional assumptions and/or equations in order to give them closure (see Subsection 2.3.3).

Species Conservation:

$$\frac{\partial\overline{\rho}\widetilde{Y}_l}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_i\widetilde{Y}_l}{\partial x_i} = \overline{\dot{\omega}}_l - \frac{\partial}{\partial x_i}(\widetilde{\overline{\rho}u_i''Y_l''}) + \overline{\rho D\frac{\partial Y_l}{\partial x_i}} \tag{2.28}$$

$\widetilde{\overline{\rho}u_i''Y_l''}$ is known as the species turbulent flux and is also given closure in Subsection 2.3.3. The last term is usually approximated by:

$$\overline{\rho D\frac{\partial Y_l}{\partial x_i}} \approx \overline{\rho}D\frac{\partial\widetilde{Y}_l}{\partial x_i} \tag{2.29}$$

The mean reaction rate $\overline{\dot{\omega}}_l$ needs to be modelled, this will be discussed in Subsection 2.5.1.

Energy Conservation:

$$\frac{\partial\overline{\rho}\widetilde{h}}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u}_j\widetilde{h}}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\overline{\lambda}_T\frac{\partial\widetilde{T}}{\partial x_i} - \widetilde{\overline{\rho}u_i''h''}\right) + \widetilde{u}_i\frac{\partial\overline{p}}{x_i} + \overline{\tau_{ij}\frac{\partial u_i}{\partial x_j}} + \widetilde{\dot{\omega}}_T \tag{2.30}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The term $\widetilde{\overline{\rho} u_i'' h''}$ is knows as the Enthalpy Turbulent Flux and will be given closure in Subsection 2.3.3.

2.3.3 Turbulence Closure

The Boussinesq Hypothesis

As mentioned during the paragragraph following Equation 2.28, some information is needed in order to give closure to the Reynolds Stresses $(-\widetilde{\overline{\rho} u_j'' u_i''})$.

Josep Valentin Boussinesq, in 1877 provided an expression [45] that relates these turbulent stresses to the flow's turbulent kinetic energy and mean velocity gradients through the help of a new quantity μ_t known as the turbulent viscosity.

This expression, called the ***Boussinesq Hypothesis*** reads:

$$(-\widetilde{\overline{\rho} u_j'' u_i''}) = 2\mu_t S_{ji} - \frac{2}{3}\left(\rho k_e + \mu_t \delta_{ji} \frac{\partial u_k}{\partial x_k}\right) \tag{2.31}$$

with the turbulent kinetic energy:

$$k_e = \frac{1}{2} \sum_{l=1}^{3} \widetilde{u_l'' u_l' l} \tag{2.32}$$

and the mean strain tensor rate:

$$S_{ji} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right) \tag{2.33}$$

$k - \epsilon$ Turbulence Model

In Equation 2.31, the expresion for the Reynolds stresses still has one variable left to be modeled: the turbulent viscosity μ_t.

$$\mu_t = \rho C_\mu \frac{k_e^2}{\epsilon} \tag{2.34}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The $k-\epsilon$ turbulence model proposed by Spalding and Launder [46] provides the final closure of the flow, by computing the turbulent viscosity as a function of the turbulent kinetic energy k_e and the dissipation rate of the kinetic energy ϵ (Equation 2.34). The variables k_e and ϵ are obtained by solving a transport equation for each of them. These equations are:

$$\rho\frac{Dk}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k_e}{\partial x_j}\right] + \left(2\mu_t S_{ij} - \frac{2}{3}\rho k_e \delta_{ij}\right)\frac{\partial u_i}{\partial x_i} - \rho\epsilon \tag{2.35}$$

$$\rho\frac{D\epsilon}{Dt} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{\epsilon_1}\frac{\epsilon}{k_e}\left(2\mu_t S_{ij} - \frac{2}{3}\rho k_e \delta_{ij}\right)\frac{\partial u_i}{\partial x_i} - C_{\epsilon_2}\rho\frac{\epsilon^2}{k_e} \tag{2.36}$$

The equation constant coefficients were obtained through data fitting. The standard values are:

Table 2.1: Standard coefficients for the $k-\epsilon$ model

C_μ	σ_k	σ_ϵ	C_{ϵ_1}	C_{ϵ_1}
0.09	1.00	1.30	1.44	1.92

Using this model, the enthaly and species turbulent fluxes $\widetilde{\overline{\rho} u_i'' h''}$ and $\widetilde{\overline{\rho} u_i'' Y_l''}$ can also be closed. According to Chen [47], the turbulent flux of a scalar quantity Ψ like a species concentration, energy or temperature can be calculated as:

$$\widetilde{\overline{\rho} u_i'' \Psi''} = -\frac{\mu_t}{\sigma_t}\frac{\partial \widetilde{\Psi}}{\partial x_i} \tag{2.37}$$

with $\sigma_t = 0.7$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.4 Chemical Modelling

2.4.1 Chemical Kinetics

Reaction Mechanism

A stoichometric chemical reaction reveals information about the relative number of moles of species that are being consumed and produced. In real life, most chemical processes don't occurr in a single reaction, but incurr in many intermediate steps or elementary reactions[2] before reaching their final form.The collection of all intermediate steps needed for a reaction is called a ***reaction mechanism*** [48].

A general elementary reaction of N species $\mathcal{M}_i$ can be represented in terms of their stoichometric coefficients a_i, b_i and reaction rate constant[3] k as follows [49]:

$$a_1\mathcal{M}_1 + a_2\mathcal{M}_2 + \cdots + a_N\mathcal{M}_N \xrightarrow{k} b_1\mathcal{M}_1 + b_2\mathcal{M}_2 + \cdots + b_N\mathcal{M}_N \tag{2.38}$$

For a mechanism of K reactions j, it can be written as:

$$\sum_{i=1}^{N} a_{ij}\mathcal{M}_i \xrightarrow{k_j} \sum_{i=1}^{N} b_{ij}\mathcal{M}_i \quad j = 1, \cdots, K \tag{2.39}$$

Reaction Rates

One of the most important variables when characterizing a chemical reaction is the ***reaction rate***, which measures the speed at which species consumed are turned into products.

A ***rate equation*** or ***rate expresion*** is a differential equation of the form[50]:

$$\dot{\omega}_i := \frac{d[\mathcal{M}_i]}{dt} = f([\mathcal{M}_1], \cdots, [\mathcal{M}_N]) \tag{2.40}$$

it provides knowledge of the functional dependency of the reaction rate $\dot{\omega}_i$ of species i on the concentrations $[\mathcal{M}_j]$ of species j, as well as other quantities such as the tem-

[2]Reaction that forms product in a single step and transition step
[3]A more appropiate term would be coefficient, as k is dependant on temperature

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

perature. It is based on the ***law of mass action*** which states that the reaction rate is directly proportional to the product of the concentrations of the reactants.

Often the right hand side Equation 2.40 expressed by a function which depends additionally on temperature [51]:

$$\dot{\omega} = k(T) \prod_{i=1}^{N} [\mathcal{M}_i]^{\alpha_i} \tag{2.41}$$

where T is the absolute temperature, $k(T)$ is the rate coefficient dependent on the temperature, and α_i is the so called order of the reaction with respect to the species $\mathcal{M}_i$.

Additionally, the total order of the reaction can be computed by:

$$\textbf{Order of reaction} = \sum_{i=1}^{K} \alpha_i \tag{2.42}$$

The reaction orders α_i must be obtained empirically from experimental data as they are not the same as the stoichometric coefficients a_i -or b_i for the reverse reaction- [51]. They are real numbers usually of small absolute value.

Way back in 1889 Svante Arrhenius [52] provided a way to understand the temperature dependence of reaction rates with a formula capable of connecting them. This formula, known as ***Arrhenius Equation*** is the following:

$$k(T) = Ae^{\frac{-E_a}{RT}} \tag{2.43}$$

where A is the reaction pre-exponential factor, E_a is the activation energy, and R is the universal gas constant.

Based on the fact, that in principle it is possible that an elementary reaction may proceed in the reverse direction, Peter [53] presents in Equation 2.44 an extension of the net reaction rate for a mechanism with K reactions involving N chemical species.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\dot{\omega}_l = \sum_{j=1}^{K}(b_{ji} - a_{ji})k_{fj}\prod_{i=1}^{N}[\mathcal{M}_i]^{a_{ji}} - k_{bj}\prod_{i=1}^{N}[\mathcal{M}_i]^{b_{ji}} \tag{2.44}$$

Note here that the subindices f and b indicate whether it refers to a backwards or forward reaction. Also, the reaction order exponents are taken as equal to the stoichometric coefficients, i.e $\alpha_{fji} = a_{ji}$ and $\alpha_{bji} = b_{ji}$. Equation 2.44 can be written too in terms of the species mass fractions Y_i:

$$\dot{\omega}_l = \sum_{j=1}^{K}(b_{ji} - a_{ji})\left[k_{fj}\prod_{i=1}^{N}\left(\frac{\rho Y_i}{W_i}\right)^{a_{ji}} - k_{bj}\prod_{i=1}^{N}\left(\frac{\rho Y_i}{W_i}\right)^{b_{ji}}\right] \tag{2.45}$$

For reversive reactions, the forward and backwards reactions rates k_{fj} and k_{bj} for species j are related via the equilibrium constant K_j^c:

$$K_j^c = \frac{k_{fj}}{k_{bj}} = \left[\frac{p_\circ}{RT}\right]^{\sum_j (b_{ji}-a_{ji})} exp\left[\sum_j\left(\frac{s_j}{R_j} - \frac{h_j}{R_jT}\right)\right] \tag{2.46}$$

$p_\circ$ is the standard pressure at one atmosphere, s_j and h_j are the entropy and enthalpy of species j. How these properties are computed is explained in the following paragraphs. These results are valid for elementary reactions. Other type of more complex reactions taking into account species which are not incorporated into a reaction exist but are not discussed in this work. These can three-body, Troe, SRI, or fallof reactions [**Troe1983**, 54, 55].

NASA 7-Polynomials

In the early 1970's, due to the need of fast computations of the gas equilibrium thermodynamical properties used in computer programs dealing with the calculations for planetary entry vehicles, the United States' National Aeronautics and Space Administration developed a set of polynomials capable of approximating said properties with small errors[56].

These polynomials -up to degree nine- were formulated by curve fitting via coefficient averaging. They can be used to approximate accurately ***molar heat capacity*** c_p^0, ***molar enthalpy*** h^0 and ***molar entropy*** s^0 for a ***reference pressure*** P_0 and a

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

range of temperatures $\boldsymbol{T}$:

$$\frac{c_p^0(T)}{R} = \upsilon_0 + a_1 T + \upsilon_2 T^2 + \upsilon_3 T^3 + \upsilon_4 T_4 \tag{2.47}$$

$$\frac{h^0(T)}{RT} = \upsilon_0 + \frac{\upsilon_1}{2}T + \frac{\upsilon_2}{3}T^2 + \frac{\upsilon_3}{4}T^3 + \frac{\upsilon_4}{5}T^4 + \frac{\upsilon_5}{T} \tag{2.48}$$

$$\frac{s^0(T)}{R} = \upsilon_0 ln(T) + \upsilon_1 T + \frac{\upsilon_2}{2}T^2 + \frac{\upsilon_3}{3}T^3 + \frac{\upsilon_4}{4}T^4 + \upsilon_6 \tag{2.49}$$

2.4.2 Plug Flow Reactor

A chemical reactor is a structure inside which chemical reaction take place, and which is of utmost importance in the field of Chemical Engineering. These reactors are designed in a manner in which the physical phenomena involved are easy to analyze and so to predict results [57]. These machines are constructed based on the study of ***ideal reactors***, mathematical models in which all processes are exactly defined.

The ***plug flow reactor*** or ***Plug Flow Reactor (PFR)*** is one of such ideal models. In a PFR all chemical reactions take place along the length of cylindrical volume and can be representede by a set of differential equations based on the following assumptions [58]: a single reaction takes place, only gradients along the axis are considered, the flow and density are constant throughout the volume -plug flow-, diffusion can be neglected -valid for turbulent flows-, and changes in pressure do not produce work or variations in temperature -steady state-.

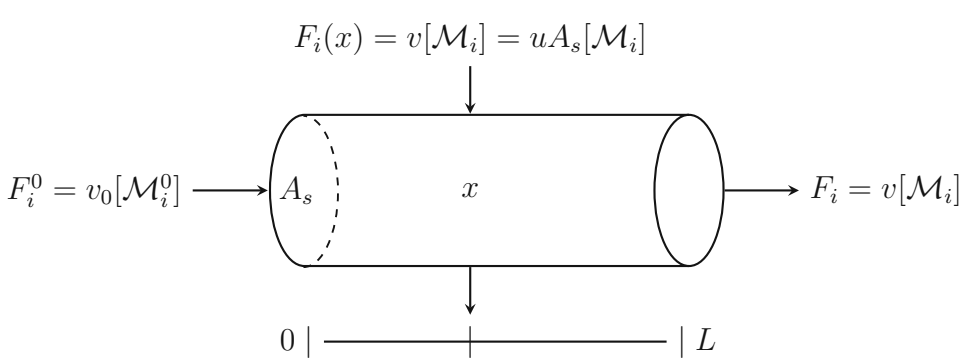


Figure 2.2: Schematics of a Plug Flow Reactor. Adapted from Schmidt [59]

Figure 2.2 shows the schematics for a cylindrical plug flow reactor of length L and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

cross-section A_s. F_i^0, F_i represent the inlet and oulet molar flows of chemical species i, $[\mathcal{M}_i]^0$ and $[\mathcal{M}_i]$ are the inlet and outlet molar concentrations of species i. And u represents the flow velocity in the axial direction. The following expresion relates molar flow to molar concentration:

$$F_i(x) = uA_s[\mathcal{M}_i(x)] \tag{2.50}$$

According to Schmidt[59] , if all plug flow reactor assumptions are met, the mass balance for a species i intervening in a reaction across a length dx and reaction rate $\dot{\omega}_i$ can be expressed as:

$$\underbrace{F_i(x)}_{inflow} - \underbrace{F_i(x+dx)}_{outflow} + \underbrace{\dot{\omega}_i A_s dx}_{\text{production/consumption}} = 0 \tag{2.51}$$

Together with Equation 2.50 and rearranging terms:

$$uA_s \frac{[\mathcal{M}_i(x+dx)] - [\mathcal{M}_i(x)]}{dx} = \dot{\omega}_i A_s \tag{2.52}$$

Simplifying and taking the limit $dx \to 0$:

$$u\frac{[\mathcal{M}_i(x)]}{dx} = \dot{\omega}_i \tag{2.53}$$

Similarly to Equation 2.45 it can be rewritten in terms of the species mass fraction Y_i, and with $u = dx/dt$:

$$\frac{dY_i}{dt} = \frac{\dot{\omega}_i W_i}{\rho} \tag{2.54}$$

The equation for the energy reads[60]:

$$\frac{dT}{dt} = -\frac{1}{\rho c_P} \sum_{i=1}^{N} \frac{h_i \dot{\omega}_i}{W_i} \tag{2.55}$$

where h_i is the partial molar enthalpy of species i. Naturally, in order to obtain the evolution of the species reacting inside the PFR it would be necessary to solve

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Equation 2.54 for each individual species, as well as for the temperature of the system.

2.5 Turbulence-Chemistry Interaction

2.5.1 Eddy Dissipation Concept

In 1981 Bjørn Magnussen presented his **Eddy Dissipation Concept** or EDC as a way to connect combustion reactions and turbulent flows[8]. The proposal divides a control volume of a reacting fluid in two definite regions: ***fine structures*** where reactions occurr, and the ***surroundings*** where they don't.

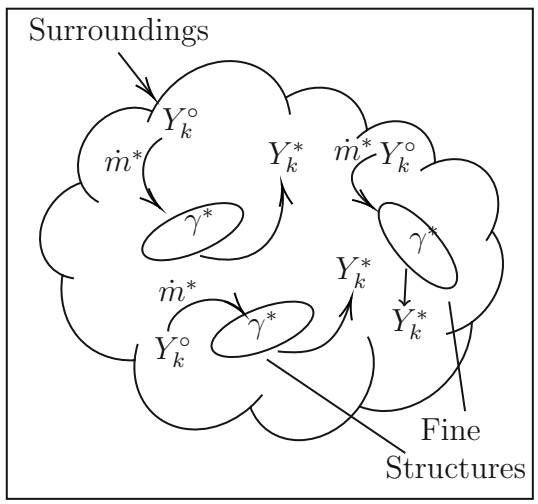


Figure 2.3: Species transfer between fine structures and surrounding volume

Figure 2.3 presents an schematic of the species transfer occurring between the fine structures and their surroundings. Mass fractions Y_k° of the surrounding fluid are transfered at a mass transfer rate $\dot{m}^*$ to the volume fractions γ^* of the fine structures due to the turbulent mixing. The different k species react inside these structures over a resident time τ^*, after which the new fine structure mass fractions Y_k^* are transported back to the surroundings again by the turbulent mixing.

As mentioned in Subsection 2.3.2, turbulent flows contain a wide spectrum of spatial scales. Energy is generated at vortices in the largest scales, and then transferred

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

-with a small fraction being dissipated- over and over to smaller scales until it reaches the Kolmogorov scales (Equations 2.20-2.22) where it is transformed into heat. This process is known as the ***energy cascade*** [61].

Using the energy cascade, energy and mass transfers to the fine structures can be calculated. These can be expressed in terms of the fine structures characteristic velocity, length and viscosity. Magnussen provides these characteristic scales for the fine structures[7]:

$$u^* = 1.75\,(\epsilon\,\nu)^{\frac{1}{4}} \tag{2.56}$$

Characteristic Velocity

$$L^* = 1.43\left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}} \tag{2.57}$$

Characteristic Length

Here u^* is the mass averaged fine structure velocity, L^* is the fine structures characteristic length. ν and ϵ are the kinematic viscosity and the energy dissipation rate.

The mass fraction occupied by the fine structures γ^*, the mass transfer per unit of mass $\dot{m}^*$ and the residence time τ^* can also be expressed in terms of these variables:

$$\gamma^* = 4.6\left(\frac{\nu\,\epsilon}{k^2}\right)^{\frac{1}{2}} \tag{2.58}$$

$$\dot{m}^* = 2.45\left(\frac{\epsilon}{\nu}\right)^{\frac{1}{2}} \tag{2.59}$$

$$\tau^* = 0.41\left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}} \tag{2.60}$$

with the relation:

$$\tau^* = \frac{2}{\dot{m}^*} \tag{2.61}$$

The mass trasfer per unit of mass of fluid $\dot{m}$ and the one per unit of mass of the fine structures $\dot{m}^*$ are related by:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\dot{m} = \gamma^* \, \dot{m}^* \tag{2.62}$$

The Mixing and Reaction Process

For a chemical species i, its concentration c_i^* inside a fraction χ of the fine structures, in the surrounding fluid c_i° can be connected via the mean concentration $\overline{c}_i$ by:

$$\frac{\overline{c}_i}{\overline{\rho}} = \frac{c_i^*}{\rho^*} \, \gamma^* \, \chi + \frac{c_i^\circ}{\rho^\circ} \, (1 - \gamma^* \, \chi) \tag{2.63}$$

or in terms of mean mass fraction $\overline{Y}_i$, mass fraction Y_i^* inside a fraction χ of the fine structures, and mass fraction of the surrounding fluid Y_i°:

$$\overline{Y}_i = Y_i^* \, \gamma^* \, \chi + Y_i^\circ \, (1 - \gamma^* \, \chi) \tag{2.64}$$

The rate at which molecules are mixed is a function of the mass transfer rate between the fine structures and the surrounding volume. The mean net mass transfer rate $\overline{R}_i$ of a species i can be written as:

$$\overline{R}_i = \overline{\rho} \, \dot{m} \, \chi \left(\frac{Y_i^\circ - Y_i^*}{W_i} \right) \overset{\text{Eq. 2.64}}{=} \frac{\overline{\rho} \, \dot{m} \, \chi}{1 - \gamma^* \, \chi} \left(\frac{\overline{Y}_i - Y_i^*}{W_i} \right) \tag{2.65}$$

The mass transfer rate might also be expressed per unit volume of the fine structure fraction χ:

$$R_i^* = \rho^* \, \dot{m}^* \left(\frac{Y_i^\circ - Y_i^*}{W_i} \right) \overset{\text{Eq. 2.64}}{=} \frac{\rho^* \, \dot{m}^*}{1 - \gamma^* \, \chi} \left(\frac{\overline{Y}_i - Y_i^*}{W_i} \right) \tag{2.66}$$

Assuming the fine structures are homogenously mixed, the net consumption rate of a specie i is dependent on its reaction rate, which is itself a function of the temperature. Because of this, an additional equation for the energy q^* stored inside the fine structures must be solved:

$$q^* = \frac{\rho^* \dot{m}^*}{1 - \gamma^* \, \chi} \sum_i^N \left(\frac{\overline{Y}_i \overline{h}_i}{W_i} - \frac{Y_i^* h_i^*}{W_i} \right) \tag{2.67}$$

with h_i and h_i^*, the total enthalpies of species i.

The mass fractions of the fine structures Y_i^* can be obtained by modeling the fine

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

structures as PFR [62], and solving the corresponding ordinary differential equations (Equation 2.54 and Equation 2.55) in terms of the fine structures residence time τ^*.

$$\begin{aligned} \frac{dY_i^*}{d\tau^*} &= \frac{W_i \dot{\omega}_i}{\rho} \\ \frac{dT}{d\tau^*} &= -\frac{1}{\rho\, c_P} \sum_{i=1}^{N} \frac{h_i \dot{\omega}_i}{W_i} \end{aligned} \tag{2.68}$$

The molar reaction rates $\dot{\omega}_i$ which contain exponetial terms are a source of "stiffness" for the overall system. Stiff problems. A broader discussion of what this "stiffness" means takes place in Section 2.6, but in general they make their solution via a numerical method more complex. According to Stone [63], their solution can consume more than 90% of the computing time in a CFD simulation.

2.6 ODE Theory

An useful concept in the analysis of ODE systems is the notion of the Initial Value Problem (IVP).

Presented are a useful definition and lemma by Ganesh [64]:

Definition 1 *Initial Value Problem*
Let $\Omega \subseteq \mathbb{R}^d$ be open. Let $\mathbb{I} \subseteq \mathbb{R}$ be an open interval. Let $f \in \boldsymbol{C}\big(\mathbb{I} \times \Omega, \mathbb{R}^d\big)$. Let $(x_0, y_0) \in \mathbb{I} \times \Omega$.

An Initial Value Problem (IVP) for a first order ODE is a relation satisfied by and unknown function y such that:

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases} \tag{2.69}$$

Let $\mathbb{I}_0 \subseteq \mathbb{I}$ be a subinterval containing x_0. A function $u \in \boldsymbol{C}^1(\mathbb{I}_0 \times \Omega, \mathbb{R}^d)$ is called a solution of the IVP if $\forall x \in \mathbb{I}_0$, the tuple $(x, u(x)) \in \mathbb{I} \times \Omega$ satisifes Equation 2.69.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Lemma 1 *Let $\Omega \subseteq \mathbb{R}^n$ and $\mathbb{I} \subseteq \mathbb{R}$. Let $f \in \boldsymbol{C}(\mathbb{I} \times \Omega, \mathbb{R}^n)$ defined by $(x, \boldsymbol{y}) \mapsto f(x, \boldsymbol{y})$.*

A continuous function $\boldsymbol{y}$ defined on an interval $\mathbb{I}_0$ containing the point x_0 is a solution of the IVP if and only if $\boldsymbol{y}$ satisfies the integral equation:

$$\boldsymbol{y}(x) = \boldsymbol{y}_0 + \int_{x_0}^{x} f(s, \boldsymbol{y}(s))\, ds \quad \forall x \in \mathbb{I}_0 \tag{2.70}$$

The main idea behind this work is to avoid the numerical integration of the system of ode's in Equation 2.68 after a given time step Δt. Thus, the targeted function is the solution for an IVP of the form:

$$\begin{aligned} \dot{\mathbf{U}}(t) &= \mathbf{f}(\mathbf{U}(t), \psi) \\ \mathbf{U}(t_0) &= [Y_1(t_0), \cdots, Y_N(t_0), T(t_0)]^\top \end{aligned} \tag{2.71}$$

where N is the number of chemical species in the system, t is the time, ψ is a set of not integrated parameters (such as the presure, density, molar weight, etc.), and the right hand side(RHS) $\mathbf{f}$ and solution vector $\mathbf{U}(t) \in \mathbb{R}^{N+1}$ defined by:

$$\mathbf{U} = [Y_1, \cdots, Y_N, T]^\top \tag{2.72}$$

$$\mathbf{f}_j = \begin{cases} \frac{W_i \dot{\omega}_i}{\rho} & \text{if } j = 1, \cdots, N \\ -\frac{1}{\rho\, c_P} \sum_{j=1}^{N} \frac{h_i \dot{\omega}_i}{W_i} & \text{if } j = N+1 \end{cases} \tag{2.73}$$

The continuity of the RHS $\mathbf{f}$ provides the existence of a solution of the IVP by the Peanno Theorem [65]. If f has also the stronger property of being Lipschitz continuous -which is the case at least locally [66]- in $\mathbf{U}$ and continuous in t, then the solution is unique by the Picard–Lindelöf theorem [67]. Lemma 1 provides an alternative representation of this function:

$$\mathbf{U}(t) = \mathbf{U}(t_0) + \int_{t_0}^{t} \mathbf{f}(s, \mathbf{U}(s))\, ds \tag{2.74}$$

this lemma also implies that if $\mathbf{f} \in \mathbf{C}^0$ then $\mathbf{U} \in \mathbf{C}^1$[68] w.r.t. time.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Using the representation from Equation 2.74, we can define a new function $\hat{\mathbf{U}} : \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N$ by:

$$\hat{\mathbf{U}} = [\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N] = [Y_1, Y_2, \cdots, Y_N] \tag{2.75}$$

this function is the target whose approximation will be accomplished via the use of an ANN. The temperature component from the solution vector $\mathbf{U}$ is dropped, as it will be computed from the enthalpy balance of the species.

As mentioned in Subsection 2.5.1, the IVP presented in Equation 2.71 is ***stiff***. Stiffnes is in itself a property that is difficult to characterize, as there is no mathematical definition of the subject. Kee [25] for example characterizes an IVP as stiff if the time step needed for stability is much smaller than the one needed to maintain accuracy. Spijker [69] gives a more formal criterion based on a model linear problem:

Definition 2 *Model Problem*
Consider the IVP :

$$U'(t) = f(t, U(t)) = AU(t)U(t_0) = U_0 \tag{2.76}$$

detone with $J(t, x)$ the jacobian matrix w.r.t. x of $f(t, x)$, with eigenvalues λ. The IVP is stiff if:

$$maxRe(\lambda) < 0 \quad and \quad \frac{max|Re\lambda|}{min|Re\lambda|} >> 1 \tag{2.77}$$

Kee provides a description of stiffness based on stability, while Spijker does in terms of a simple model problem. The notion of A-stability integrates both approaches into a concept that provides criteria to select suitable numerical solver for the stiff IVP's.

Fasshauer [70] provides useful definitions:

Definition 3 *Stability*
Consider an IVP with a bounded solution u. A numerical integrator for the IVP

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

is considered stable if small perturbations in the initial conditions do not cause the approximation $\hat{u}$ to diverge from u.

Definition 4 *Scalar Model Problem*
Consider the IVP:

$$u'(t) = \lambda u(t) \tag{2.78}$$

$$u(t_0) = u_0 \tag{2.79}$$

and from Praetorius [68]

Definition 5 *A-stability*
Consider the IVP for the scalar model problem. Consider a linear multi step (m-step) method which approximates the solution u of the IVP by:

$$\sum_{j=0}^{m} \alpha_j u_{j+1} = \underbrace{\lambda \Delta t}_{z} \sum_{j=0}^{m} \beta_j u_{j+1} \tag{2.80}$$

and whose characteristic polynomial $\hat{q}(\zeta) = \sum_{j=0}^{m} (\alpha_j - z\beta_j)\zeta^j$ satisfies the root condition:

- $\hat{q}(\zeta) = 0$ *implies* $|\zeta| \leq 1$.
- *It* $|z| = 1$ *is a root of* $\hat{q}(\zeta)$*, then* ζ *is single.*

Define the domain of stability $\mathcal{D}$ as:

$$\mathcal{D} := \{z \in \mathbb{C} | \hat{q}(\zeta) \textit{satisfies the root condition}\} \tag{2.81}$$

The m-step method is A-stable if its domain of stability $\mathcal{D} = \mathbb{C}^-$, i.e. the left side of the complex plane.

The notion of A-stability allows one to choose a numerical integrator which is stable for the model scalar problem (which is stiff), and thus, that could be applied to comparatively similar problems. By Dahlquist's Second Barrier [71], for a method to be A-stable, it must be implicit and of order at most 2. The first and second Backward Differentiation Formulae (Backwards Differentiation Formulae (BDF)) methods, are

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

a type of linear multi step methods that comply with the Dahlquist Barrier. They look:

$$u_{l+1} - u_l = \Delta t f(t_{l+1}, u_{l+1}) \tag{2.82}$$

BDF 1

$$u_{l+2} - \frac{4}{3}u_{l+1} + \frac{1}{3}u_l = \frac{2}{3}\Delta t(t_{l+2}, u_{l+2}) \tag{2.83}$$

BDF 2

or generally for a formula of order O_p, with coefficients α_i, β_0 chosen such that order O_p is achieved:

$$\sum_{i=0}^{O_p} u_{l-i} = \alpha_i u_{l-1} = \Delta t \beta_0 f(t_l, u_l) \tag{2.84}$$

The Variable-coefficient Ordinary Differential Equation Solver (VODE) developed by Browm, Byrne and Hidmarsh [72] is an BDF implementation, which is widely used for reaction mechanism development in combustion simulations for its efficiency and robustness when dealing with stiff problems [63]. It solves Equation 2.84 iteratively by defining an iteration function Φ:

$$\Phi(u_l) := u_l - \Delta t \beta_0 f(t_l, u_l) - \sum_{i=0}^{O_p} u_{l-i} \tag{2.85}$$

and performing an iteration with the jacobian J of f and a correction Δu_l:

$$\left[I - \Delta t \beta_0 J(u_l^{(i)})\right] \Delta u_l^{(i)} = -\Phi(u_l^{(i)}) \tag{2.86}$$

$$u_l^{(i+1)} = u_l^{(i)} + \Delta u_l^{(i)}$$

the step size h and order p might be adapted on the fly depending on local error estimates. The iterations are performed until tolerance conditions for $\Delta u_l^{(i)}$ or Φ are met.

The chemical species mass fractions Y_i presented in the ODE system for a PFR , and needed for the use of the EDC, will be obtained via the use of VODE . This mass fractions will then be used to generate a dataset in order to train an ANN. Precise

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

details can be found in Section 4.3.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

3 Artificial Neural Networks

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E
>
> *Tom M. Mitchell,1997*

3.1 Machine Learning

3.1.1 General Remarks

Machine Learning, often abbreviated as ***Machine Learning (ML)*** , its a subset of methods within the greater field of ***Artificial Intelligence***, which uses mathematical and statisticial techniques to build models and algorithms based on provided data without explicit programming, but by *"learning"* from it [73].

Burkov [74] summarizes ***ML*** as a way to find a solution to a given problem by:

1. Selecting an appropiate data set of examples.
2. Using one or various algorithms to construct a model with these examples.

The way in wich information is gained by the algorithm can be done via several different paradigms: ***supervised, semi-supervised, unsupervised*** and ***reinforced learning***, with most applications -including the work presented in this thesis- adhering to the ***supervised*** kind.

According to Burkov [74], the three building blocks of every learning algorithm are:

1. An ***error*** or ***loss*** function.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2. A criterion for optimization based on the chosen ***loss function.***.

3. A training routine that utilizes the available data to solve for the optimization criterion.

3.1.2 Training Data

The objective of ***supervised learning*** is to generate a mapping from a given input to a sought output, typically known as ***labels***, requiring that the algorithm's possible outcomes are known beforehand.

The important concepts of feature vectors and datasets are defined by Simovici [75]:

Definition 6 ***Feature Vector and Labels***

A d-dimensional vector $\boldsymbol{x}$ *that contains in each of its d dimensions a* ***feature*** $x^{(i)}$ *which serves to describe itself is called a* ***feature vector****.*

A ***label*** y *is an element belonging to any set -a number, a vector, a category, etc.- which holds information regarding the desired characteristic of an associated* ***feature vector****.*

Definition 7 ***Dataset***

A collection of pairs

$$\{(x_j, y_j)\}_{j=1}^{N_s}$$

,such that x_j *is a feature vector and* y_j *is a label, is called a* ***dataset****. Each individual pair is known as a* ***labeled example****. The integer* N_s *denotes the number of pairs or samples in the dataset.*

As a general practice, the data available is randomized and separated in three distinct groups of different sizes, known as ***training, validation*** and ***test*** datasets. Important assumptions for these sets are that all examples are ***independent*** of each

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

other, and that the sets are ***identically distributed*** [76]

The main reason behind choosing this approach is the idea that when building a model, it is undesirable for it to only perform well on predicting the labels of seen examples, but rather it is desired for the model to be able to ***generalize*** - i.e. provide good results for unobserved inputs - .Thus, the ***training set*** is used to adjust the model's *parameters*, the ***validation set*** allows for evaluating the so called ***hyperparameters*** and ***architecture***, and the ***test set*** gives the developers a tool for testing before final deployment.

In the past, usual practice suggested a 70 - 30 - 30 percent split for the three sets, however now, in times of big data, with the number of labeled examples now numbering in millions an split of 95 - 2.5 - 2.5 percent is quite common[74]. This is of course is dependant on the size and the quality of the available data. The procedures with which the data is gathered on and then split will be discussed in Subsection 4.3. It is worth mentioning that a significantly different method will be followed for the evaluation of the neural networks in this work, as a different type of ***test set*** will be created (Subsection 4.4.2), unlike the more traditional one, which is a subset of all the available data.

3.1.3 Model Accuracy

Underfitting and Overfitting

Two concepts of utmost importance in learning theory are those of ***underfitting*** and ***overfitting*** illustrated in Figure 3.1.

An ***underfitting*** model is one that produces a high error when evaluated with the same data used to train it. This may be caused by a lack in complexity in the selected model, or by an incapacity in the available data to provide good information.

A model that ***overfits*** might provide reliable results for the examples in the ***training set***, but fail to generalize for unseen data. Contrary to a model that underfits, possible causes include an exceedingly high complexity, or a limited number of examples with too many features. A way to handle overfitting is studying the relation

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

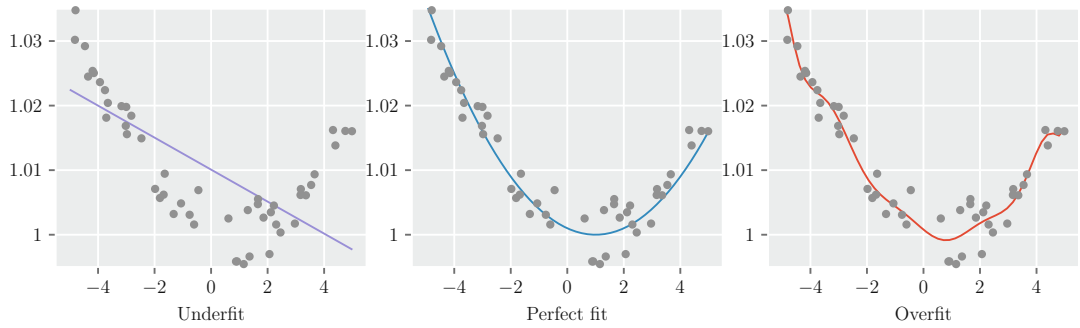


Figure 3.1: Examples of underfit, proper fit and overfit

between train and test errors.

Error Propagation

Because of the nature of the proposed application for the results from this work, the author considers another factor that is perharps more impactful than the two phenomena mentioned above: the issue of ***error propagation***. This is because for the intended use of predicting the evolution of chemical fractions in a mixture, after the first prediction, each subsequent one will carry the error from the previous.

This is a facet that the author considers a weakness of many of the previous studies regarding the application of ***artificial neural networks*** in combustion simulations, as the reported results usually deal with the error resulting from comparing prediction based on true data, but offer no insight on errors arising from predictions generated based on previously predicted values. An in-depth description pertaining to this matter will be explored in Subsection 4.4.2.

3.2 Artificial Neural Networks

3.2.1 General Remarks: The Perceptron

Inspired by the neurological processes with which brain cells or neurons respond to stimuli via structures known as synapses, that allow them to target other specific neurons (or a different type of body cell) [77], ***Artificial Neural Networks*** are a type of mathematical functions that are central to most novel ***Machine Learning*** applica-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

tions. By learning from input data via a training process, an artificial neural network is capable of capturing information coming from these examples and improving itself by storing it in its internal ***learning parameters***. Artificial Neural Networks model their biological counterparts based on basic mathematical operations such as sums, multiplications and function compositions, however, despite their simplicity, they are capable of approximating extremely complex models.

In 1943, the McCulloch-Pitts Neuron [78] was one of the first models capable of approximating an ***Input-Output*** function by separating inputs on different categories depending of the resulting sign of said function. It however had the inconvenience that the model parameters had to be set manually. Almost two decades later, Rosenblatt's Perceptron [79] became the first function of this type to be able to modify its own parameters directly from the seen data.

On a biological brain, any single neuron is only activated to release a chemical signals (its output) when the stimuli received from other neurons through synapses passes a particular excitation level or ***threshold*** [80]. The **Rosenblatt Perceptron** characterizes the process by multiplying the input data with a so called ***vector of weights*** representing the intensity of a synapse, passing its resulting weighted sum through a type of function known as the ***activation***, and then comparing it with a previously defined excitation ***threshold*** which decides whether to transmit or inhibit the resulting signal. Rossini [81] proposes a formal definition of the perceptron:

Definition 8 ***Rosenblatt Perceptron***
Let $x, w \in \mathbb{R}^n$ be a vector of inputs and a vector of weights respectively. Let $\Theta \in \mathbb{R}$ be an activation threshold value and g any activation function. The Rosenblatt Perceptron $\{-1, 1\} \ni \boldsymbol{y} : \mathbb{R}^n \to \mathbb{R}$ is defined by:

$$y(x) = g(w^\top x - \Theta)$$

With the standard vector-vector product:

$$w^\top x = \sum_{j=1}^{n} w_j x_j$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Minsky and Papert identified however in 1969 some issues with the perceptron, mainly how it is only capable of approximate functions dealing with data that is ***linearly separable***. Nevertheless, further research over the following decades led to further advances such as the ***Multilayer Perceptron*** and the ***Multilayer Feedforward*** architecture, as well as solid mathematical results showing the applicability of ***Neural Networks*** in approximating continuos functions.

The next subsections outline in deeper detail the intricacies of a ***Deep Feedforward Neural Network*** as its use its central to the results of this work. The chapter will finalize with a further discussion on the mathematical theory describing said networks as approximation functions.

3.2.2 Feedforward Neural Networks

The term Feedforward Neural Network refers to a particular type of algorithm architecture designed to work -and learn- in layers, with the layers composed themselves of a number of neurons. The name feedforward relates to the fact that the indormation flow runs in a straight manner, i.e. starting from the input, then through the intermediate computations, and then the output; while the network part comes from the fact that these types of algorithms are based on the composition of many different functions [76].

Simovici [75] describes this type of networks as an ***acyclic directed graph*** with a set of nodes N made up of neurons and whose edges direct information flow. Formally, he provides the following definitions:

Definition 9 ***Neuron***

Let $x \in \mathbb{R}^n$ be a vector of inputs. A neuron is defined by a vector of weights $w \in \mathbb{R}^n$, an activation threshold value $\Theta \in \mathbb{R}$ and an activation function $\sigma : \mathbb{R}^n \to \mathbb{R}$. The processing unit output $\boldsymbol{y} \in \mathbb{R}^n$ is computed by:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$y = \sigma(w^\top x + \Theta)$$

Remark 2 *In biological organisms, a neuron output is always discrete -the synapse either fires or not-. It is possible for the output of an artificial neuron to belong to a continouos range of real numbers, or even be a complex number. [82]*

Definition 10 ***Neural Network***

A neural network (NN) is a triplet $\mathcal{N} = (N, E, w)$, where $\mathcal{G} = (N, E)$ is a graph and $w : E \to \mathbb{R}$ that specifies the ***weight*** *$w_{ij} = w(i, j)$ of and edge (i,j) between the neurons i and j.*

Initially a linear transformation is applied to an ***input vector*** to create a ***linear layer***; then, an ***activation function*** is used to evaluate said result. Here the linear mapping essentially provides a weighting system that influences whether a neuron in a subsequent layer will activate or not. The output values from the activation function become then, the input for the next layer where the process is repeated until completion. When all the outputs of a layer serve as input for every individual layer of the next one, then said network is said to be ***fully connected***.

The number of layers in a feedforward neural represents its ***depth***, while the number or neurons in a layer is referred as its ***width***. A single-inner-layer neural network is known as ***shallow***, with the term ***deep network*** referring to those with more. However, in the current media environment it is usually used when discussing networks with thousands or even millions of layers.

Highman [83] summarizes the action of a feedforward network of L layers on and input $x \in \mathbb{R}^{n_1}$. The activation function is denoted by σ, $a^{(l)} \in \mathbb{R}^{n_l}$ is the output of layer l, and $b^{(l)} \in \mathbb{R}^{n_l}$, $w^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ represent the vector of biases and matrix of weights of layer l:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$a^{(1)} = x \tag{3.1}$$

$$a^{(l)} = \sigma\left(w^{(l)}a^{(l)} + b^{(l)}\right) \quad l = 2, 3, \cdots, L \tag{3.2}$$

Deep Neural Networks can mimic any kind of continuous function on a compact set with a selection of the correct weights and activation functions. This means is possible to construct iteratively a map $f^*(x, \theta)$ profficient in approximating a continuous function $f(x)$ belonging to a class defined as $f : \mathbb{R}^n \to \mathbb{R}^m$.

3.2.3 Activation Functions

Rectifier Linear Unit (ReLU)

Probably the most popular type of activation used in deep learning, the output of the rectifier corresponds to the positive part of its given argument. We can define it in the context of neural networks:

Definition 11 ***ReLU***

Let $x \in \mathbb{R}^d$ be a d-dimensional input vector, $ReLU : \mathbb{R}^d \to \mathbb{R}^d$ is defined as:

$$ReLU(x_i) = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Remark 3 *The partial derivatives of the ReLU are given by:*

$$\frac{\partial ReLU(x)}{\partial x_i} = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i < 0 \end{cases} \tag{3.4}$$

Note that the derivative is undefined at 0.

This function has proved useful in replacing more traditional activations such as *Tanh* and *Sigmoid*, as its lower computational cost provides an improvement in both

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

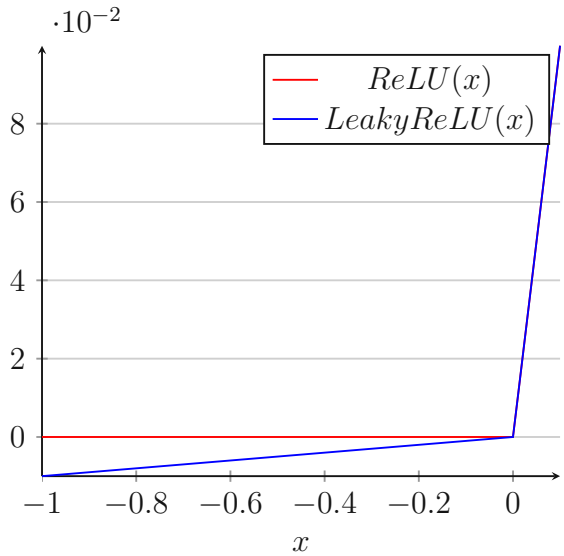


Figure 3.2: ReLU and LeakyReLU functions

training and prediction speed [84]. As an additional advantage, neural networks trained with this activation via *gradient descent* don't suffer from the *vanishing gradient problem* identified by Hochreiter in 1991 [85], an issue occurring in which as gradients tend to zero, no training is accomplished. Both, the speed-up during the backpropagation, and the avoidance of the *vanishing gradient*, can be easily understood via **Remark 3**, since aslong as the output of a given neuron remains positive, the derivative of the activation will always be one.

However, despites its advantages, the *linear rectifier* does not come without its own set of issues, mainly the so called *dying ReLU problem*. Again, from **Remark 3**, it is feasible to glimpse because of the 0 factor of the derivative in the *backpropagation*, that after the output of a neuron veers into the non-positive side of the *real line* it will persist unchanged, effectively halving its training and returning zero for any input, said neuron is thus considered *"dead"*. Lu et al [86] proved rigourously how as the number of layers in a ReLU network go to inifnity, it will die in probability. The *Leaky ReLU* variant of the rectifier, introduced in 2014, provides a valid solution without a significant impact in computing performance [87]. Figure 3.2 shows the difference in behavior between these functions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Definition 12 ***LeakyReLU***

Let $x \in \mathbb{R}^d$ be a d-dimensional input vector, $LeakyReLU : \mathbb{R}^d \to \mathbb{R}^d$ is defined as:

$$LeakyReLU(x_i) = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \alpha x_i & \text{otherwise} \end{cases} \tag{3.5}$$

Where $\alpha = 0.01$

From the definition of the LeakyReLU activation, a further alternative was proposed in 2015 by He et al. [88], which generalizes the rectifier by treating $\alpha \in \mathbb{R}_{\geq 0}$ as a learnable parameter. This function is commonly known as the ***Parametric ReLU***.

3.2.4 Loss Function

An important part of ***supervised learning*** is the notion of the ***loss function***. At its most basic, ***supervised learning*** is nothing more than the use of ***optimization algorithms*** to minimize said ***loss***. The ***loss***, also called ***error function*** itself can be any function that is able to map onto a real number the error between a model's prediction and the expected output.

An important loss function is the ***Mean Squared Error*** which is a measure of the average of the error squared of a prediction, i.e. the $L^2 - norm$ squared of the difference between and output and its corresponding target value.

Definition 13 ***Mean Squared Error***

Let $y_i, \hat{y}_i \in \mathbb{R}^d$ be the ***label*** *and* ***prediction*** *associated to an input vector x_i. Let*

$$y = \begin{bmatrix} y_1^\top \\ \vdots \\ y_{N_s}^\top \end{bmatrix}, \hat{y} = \begin{bmatrix} \hat{y}_1^\top \\ \vdots \\ \hat{y}_{N_s}^\top \end{bmatrix}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The Mean Squared Error, ***MSE*** $: \mathbb{R}^{N_s \times d} \to \mathbb{R}$ *is defined as:*

$$MSE(y, \hat{y}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \|y_i - \hat{y}_i\|_{L^2}^2$$

The Mean Squared Error loss will see heavy use during the course of this work.

3.2.5 Training via Backpropagation

The optimizer

A neural network's training consists on adjusting the weights and bias belonging to each layer so that the loss between data set targets and predicted values is minimized [83].

A popular iterative minimization algorithm of cost functions which are differentiable is the method of ***Gradient Descent*** [89], also known as Steepest descent, which tries to find a function's minimum taking steps proportional to its gradient. Provided, a function f fulfills certain conditions, the gradient descent method finds a minimum, as shown in the following theorem from Simovici [75].

Theorem 1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable, with Lipschitz continoous derivative and Lipschitz constant $\frac{1}{\eta_L}$ for some $\eta_L > 0$. Consider the sequence of x_k defined as:*

$$x_{k+1} = x_k - \eta_L \nabla f(x_k) \tag{3.6}$$

for $k \in \mathbb{N}$, and initial point x_0. Suppose x_{min} is a minimizer of f, then:

1. *$lim_{k \to \infty} x_k = x_{min}$*
2. *The sequence $\|x_k - x_{min}\|$ decreases strictly and converges to 0.*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3. $f(x_k) - f(x_{min}) \leq \frac{5}{2\eta_L \sum_{i=0}^{k-1} \|x_i - x_{min}\|}^{-2}$

The constant η_L in Theorem 1 determines how big of a step is taken in the direction of the gradient at each iteration during the minimization of the loss function. It is known as the ***learning rate*** [90]. It is part of a set of values known as ***hyperparameters***, which control the learning process, but cannot be inferred through it.

For a loss function $E(\theta)$ parametrized by a vector of parameters $\theta \in \mathbb{R}^d$ of the form:

$$E(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} E_i(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} E_i(\theta, y^{(i)}, \hat{y}^{(i)}) \tag{3.7}$$

with N_s training examples, and $y^{(i)}, \hat{y}^{(i)}$ being a prediction-target pair, gradient descent adjust the vector of parameters in the following manner:

$$\theta_{k+1} = \theta_k - \eta_L \nabla E(\theta) = \theta_k - \frac{\eta_L}{N_s} \sum_{i=1}^{N_s} \nabla E_i(\theta) \tag{3.8}$$

However, a complication of using gradient descent to adjust the weights of a neural network, is that it needs to compute the gradients of the whole dataset to complete an iteration, which might not only be slow but may actually be impossible if the data size is bigger than the memory available [91].

The ***Stochastic Gradient Descent*** or ***Stochastic Gradient Descent (SGD)*** decreases the computational costs related to the standard gradient descent by approximating the actual gradients via a sequence of gradients obtained from random samples of the data examples [92] and updating the parameters after every example is seen (Equation 3.9). It thus gets faster iterations but with the trade-off of lower convergence rates and oscillations [93].

$$\theta_{k+1} = \theta_k - \eta_L \nabla E_i(\theta) \tag{3.9}$$

A nice compromise between both methods is performing SGD in batches of N_b training examples. This can lead to a more stable convergence, and make use of the ability of modern machine learning programming libraries to compute gradients w.r.t. great ammounts with high efficiency. In batch gradient descent the gradient of the loss

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

function w.r.t. the batch has the form:

$$\nabla E(\theta) = \frac{1}{N_b} \sum_{i}^{N_b} \nabla E_i(\theta) \tag{3.10}$$

From here on, whenever mentioned, the gradient $\nabla E(\theta)$ refers to the expression in Equation 3.10.

Finally, the ***Adaptive Moment Estimation*** algorithm, or ***Adam*** is discussed. Presented by Kingma and Ba in 2014 [94] it belongs to a family of algorithms known as adaptive methods which use a different learning rate for each of its parameter. Adam-s learning rates keep track of the exponential moving averages m_θ^t of the gradients and v_θ^t squares of the gradients:

$$m_\theta^{t+1} = \beta_1 m_\theta^t + (1 - \beta_1)\nabla_\theta E^t \tag{3.11}$$

$$v_\theta^{t+1} = \beta_2 v_\theta^t + (1 - \beta_2)(\nabla_\theta E^t)^2 \tag{3.12}$$

and uses them to provide estimates for the gradient's first and second moments ($\hat{m_\theta^t}$ and $\hat{v_\theta^t}$) corrected for bias [91]:

$$\hat{m}_\theta = \frac{m_\theta^{t+1}}{1 - \beta_1^{t+1}} \tag{3.13}$$

$$\hat{v}_\theta = \frac{v_\theta^{t+1}}{1 - \beta_2^{t+1}} \tag{3.14}$$

the parameters are updated by:

$$\theta^{t+1} = \theta^t - \eta_L \frac{\hat{m}_\theta}{\sqrt{\hat{v}_\theta} + \epsilon^\circ} \tag{3.15}$$

The author proposes the values $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The value $\epsilon^\circ = 10^{-8}$ is added to the denominator to avoid division by 0. The operation $\sqrt{\hat{v}_\theta}$ is performed elementwise.

The Adam algorithm is currently one of the most widely used optimizers for neural network training, with more than thirty thousand citations as of 2020 [95]. With many

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

authors arguing whether or not it offers real advantages over SGD , and doubting the validity of the proof of convergence reported by Kingma and Ba [96]. Nevertheless, Adam will be the chosen method for minimizing the loss function of the different networks built for the purpose of this work, as it provides a faster initial progress than SGD, and is less sensible to the choice of hyperparameters [97].

Backpropagation

As has been seen, the different optimizers presented in this subsection minimize the loss function between a network's output and training targets by updating iteratively the network's inner parameter. All the algorithms discussed for this purpose however rely on the knowledge of the gradients of the loss output with respect to each of the individual parameters of the model.

These individual gradients can for most cases be computed analytically, but this might require the implementation of complicated expressions which could even be expensive to compute.

The ***backpropagation*** algorithm constitutes an efficient way to compute the necessary gradients for the parameter updates [98]. It was discovered and rediscovered many times during the 1960's, but made popular in the context of neural networks by Rumerlhart et al. in 1986 [99].

Backpropagation belongs to a broader field known as ***Automatic Differentiation***, which is concerned with methods to compute derivatives automatically [76]. Modern machine learning libraries separate the different operations applied to input values, build a computational graph for them, and then obtain numerical derivatives by applying the chain rule recursively [100].

This recursion can be done in two main ways. ***Forward Mode Accumulation*** keeps the output fixed and computes derivatives of the inner operation until reaching the variable with respect to which the gradient is desired. ***Reverse Mode Accumulation*** on the other hand keeps the input variable fixed, and computes the derivatives of inner operations in the direction of the output variable. Reverse forward accumu-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

lation has advantages when the number of outputs of a function is smaller than the number of inputs.

Equation 3.16 and Equation 3.17 show the computations for forward and backward accumulation respecitvely, p and q represent the variable differentiated and the variable w.r.t. which derivation is carried. The variables ξ represent the model inner values after each sub-operation.

$$\frac{\partial p}{\partial q} = \frac{\partial p}{\partial \xi_{n-1}} \frac{\partial \xi_{n-1}}{\partial q} = \frac{\partial p}{\partial \xi_{n-1}} \left(\frac{\partial \xi_{n-1}}{\partial \xi_{n-2}} \frac{\partial \xi_{n-2}}{\partial q} \right) = \cdots \tag{3.16}$$

$$\frac{\partial p}{\partial q} = \frac{\partial p}{\partial \xi_{1}} \frac{\partial \xi_{1}}{\partial q} = \frac{\partial p}{\partial \xi_{1}} \left(\frac{\partial \xi_{1}}{\partial \xi_{2}} \frac{\partial \xi_{2}}{\partial q} \right) = \cdots \tag{3.17}$$

Backpropagation is a special case of reverse mode accumulation, and adjust the weigths in two phases [75]: the ***forward pass*** in which inputs are evaluated and the outputs compared with correct targets, and then a ***backward pass*** which computes the derivatives and adjusts the weights.

Highman [83] provides a lemma for the computation of the gradients of the loss w.r.t. the network parameter trough backpropagation:

Lemma 2 *Backpropagation*
Let $a^{(L)} \in \mathbb{R}^{n_L}$ be the output of a feed forward neural network with N_L layers. $a^{(l)} \in \mathbb{R}^{n_l}$ is the output of layer l, and $b^{(l)} \in \mathbb{R}^{n_l}$, $w^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ represent the vector of biases and matrix of weights of layer l.

Let $x \in \mathbb{R}^{n_1}$ be a vector of inputs and $y \in \mathbb{R}^{n_L}$ be a vector of targets.

Define the weighted input $z^{(l)} \in \mathbb{R}^{n_l}$ of layer l and its layer output $a^{(l)}$ by:

$$z^{(l)} = w^{(l)} a^{(l-1)} + b^{(l)} \tag{3.18}$$

$$a^{(l)} = \begin{cases} x & \text{if } l = 1 \\ \sigma\left(z^{(l)}\right) & \text{for } l = 2, 3, \cdots, N_L \end{cases} \tag{3.19}$$

For a loss function $E = \frac{1}{2} \left\| y - a^{(l)} \right\|_2^2$ define $\delta^{(l)} \in \mathbb{R}^{n_l}$ by:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\delta^{(l)} = \frac{\partial E}{\partial z^{(l)}} \tag{3.20}$$

then, the gradients of the loss function E with respect to the model parameters $b_j^{(l)}$ and $w_{ij}^{(l)}$ can be computed as:

$$\delta^{(L)} = \sigma'\left(z^{(L)}\right) \circ \left(a^{(L)} - y\right) \tag{3.21}$$

$$\delta^{(l)} = \sigma'\left(z^{(l)}\right) \circ \left(w^{(l+1)}\right)^\top \delta^{(l+1)} \qquad for\ 2 \leq l \leq N_L - 1 \tag{3.22}$$

$$\frac{\partial E}{\partial b_i^{(l)}} = \delta_i^{(l)} \qquad for\ 2 \leq l \leq N_L \tag{3.23}$$

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)} \qquad for\ 2 \leq l \leq N_L \tag{3.24}$$

where $\circ$ represents Hadamard or element-wise product: $(x \circ y)_i = x_i y_i$.

3.3 Approximation Properties of Neural Networks

3.3.1 Theoretical Results

Many mathematical results give testament of the capabilities of neural networks for approximating certain functions, these results are usually identified as ***Universal Approximation Theorems*** and can be generally divided by whether they consider an arbitrary number of either layers -depth- or neurons -width-. However they don't provide insight on how to construct said networks, which is still a topic of research in mathematics.

One of the earliest results that started the discussion of ANN as function approximators was formulated by Andrey Kolmogorov and Vladimir Arnold in 1957 as a solution of the 13th of Hilbert's 23 problems.

Theorem 2 *Kolmogorov-Arnold Representation Theorem*
Let $\mathbb{I} = [0,1]^n, n \geq 2$. Let $x = (x_1, \cdot, x_n)$. Any vector valued continuous function f defined on $\mathbb{I}$ can be represented by the composition of single variable continuous

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

functions g_j and monotonically increasing functions $\phi_{i,j}$:

$$f(x) = \sum_{i=0}^{n} g_i \left(\sum_{j=0}^{n-1} \phi_{i,j}(x_i) \right) \tag{3.25}$$

The importance of this theorem in the field of learning theory is however an issue of dispute [101, 102], with many authors arguing if it is indeed relevant.

The next important result arrives in 1989 by the hand of George Cybenko [11], whose proof based on the Hahn-Banach theorem showed how a feed forward neural net with a specific type of activation function known as sigmoidal can approximate any continuous real-valued function defined in a compact subset of $\mathbb{R}^n$.

Definition 14 *Sigmoidal Function*
A non-decreasing function $\sigma : \mathbb{R} \to \mathbb{R}$ such that:

$$\lim_{x \to \infty} \sigma x = 1$$

and

$$\lim_{x \to -\infty} \sigma x = 0$$

is known as a sigmoid function.

Theorem 3 *Universal Approximation Theorem(Cybenko)*
Let $\mathbb{I}_n = [0,1]^n$, $x, w_i \in \mathbb{I}_n$, $\theta_i \in \mathbb{R}$. Let σ be a sigmoidal continuous function. Any continuous function $f \in C(\mathbb{I}_n)$ can be arbitrarily approximated by a function of the form:

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^\top x + \theta_i)$$

Cybenko's theorem belongs to the class of results dealing with an arbitrary number of neurons, while the layers are finite, with even just one layer being sufficient as long as the amount of neurons is allowed to grow.

Recent forms of Theorem 3 build approximations as the compositions of affine mappings instead of being limited to sigmoidal functions. This comes after advances from

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Hornik [103] and Leshno [104] which showed that this approximation property of neural networks is not tied to a specific type of function and that it is equivalent to selecting any non-polynomial activation.

Some more useful related results came much later: in 2014 Ismailov [105] showed with Theorem 2 that a two layer network of n inputs with the same amount of neurons in the first layer and $2n + 2$ in the second can approximate any continuous multivariate function with a specially crafted activation function; and in 2017 Lu et al. [106] proved that $n+4$ neurons are enough for ReLU networks to approximate any Lebesgue integrable function from $\mathbb{R}^n$ to $\mathbb{R}$ with respect to the $\mathbf{L}^1$ distance, and that n neurons are not enough.

Perhaps the most interesting theorem related to this thesis did not come up until recently in 2020 by the hands of Kidger and Lyons [107] as it gives a precise number of neurons in the hidden layers of a feed-forward network of arbitrary depth depending on the number of input and output variables.

Theorem 4 *Universal Approximation Theorem (Kidger & Lyons)*
Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an arbitrary continuous function which is not affine, continuously differentiable at at least one point, and whose derivative is not zero at said point.

Let $\boldsymbol{K} \subseteq \mathbb{R}^n$ and $f : \boldsymbol{K} \to \mathbb{R}^m$ be continuous.

It exists a feedforward neural network F of n inputs and m outputs such that $F(x)$ can approximate $f(x)$ arbitrarily for all $x \in \boldsymbol{K}$ with respect to the uniform norm, provided each hidden layer has $n + m + 2$ neurons and uses σ as activation function.

While it might be tempting to design a newtwork based on the ideas of Theorem 4 to approximate a function, it has two drawbacks, it provides no information on how to construct the weights (common to most related theorems), and places no restrictions on the amount of layers needed. Thus, while a network's width could be set in accordance to Kidger and Lyons, its depth could grow so large that an input evaluation becomes significantly slower than computing the function approximated by a more traditional method. Furthermore, it offers no other guidance regarding the width of a network, it

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

says that the set of networks with $n + m + 2$ is dense in $\mathbf{C}(\mathbf{K} \subseteq \mathbb{R}^n; \mathbb{R}^m)$, it doesn't exclude other network spaces from also having a density property.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4 Methodology

4.1 General Remarks

This chapter deals with the procedures followed in order to build an ANN approximator capable of receiving a computational cell's chemical species mass fractions, temperature and reaction time; and returning the system evolution's mass fractions.

To succesfully build this type of model proper training data must be provided in order for the Neural Network to "learn" the proper behavior of the chemical reactions; an appropiate evaluation procedure must be implemented to allow for correct architecture selection; and finally the best selected model should be coupled with an existent turbulent flow solver to provide feedback and final results. Figure 4.1 illustrates the decision processes undergone throughout this work.

Further description of the plan takes place in the following sections discussing in more detail training data generation, Neural Network design and fluid dynamics coupling.

For this work, the methodological framework proposed by Laubscher [18, 19] to generate a training data set for a network designed for seven chemical species is taken as a starting point and then expanded upon, by adding the ability to handle more species and proposing a different method for the evaluation of the neural networks' accuracy.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

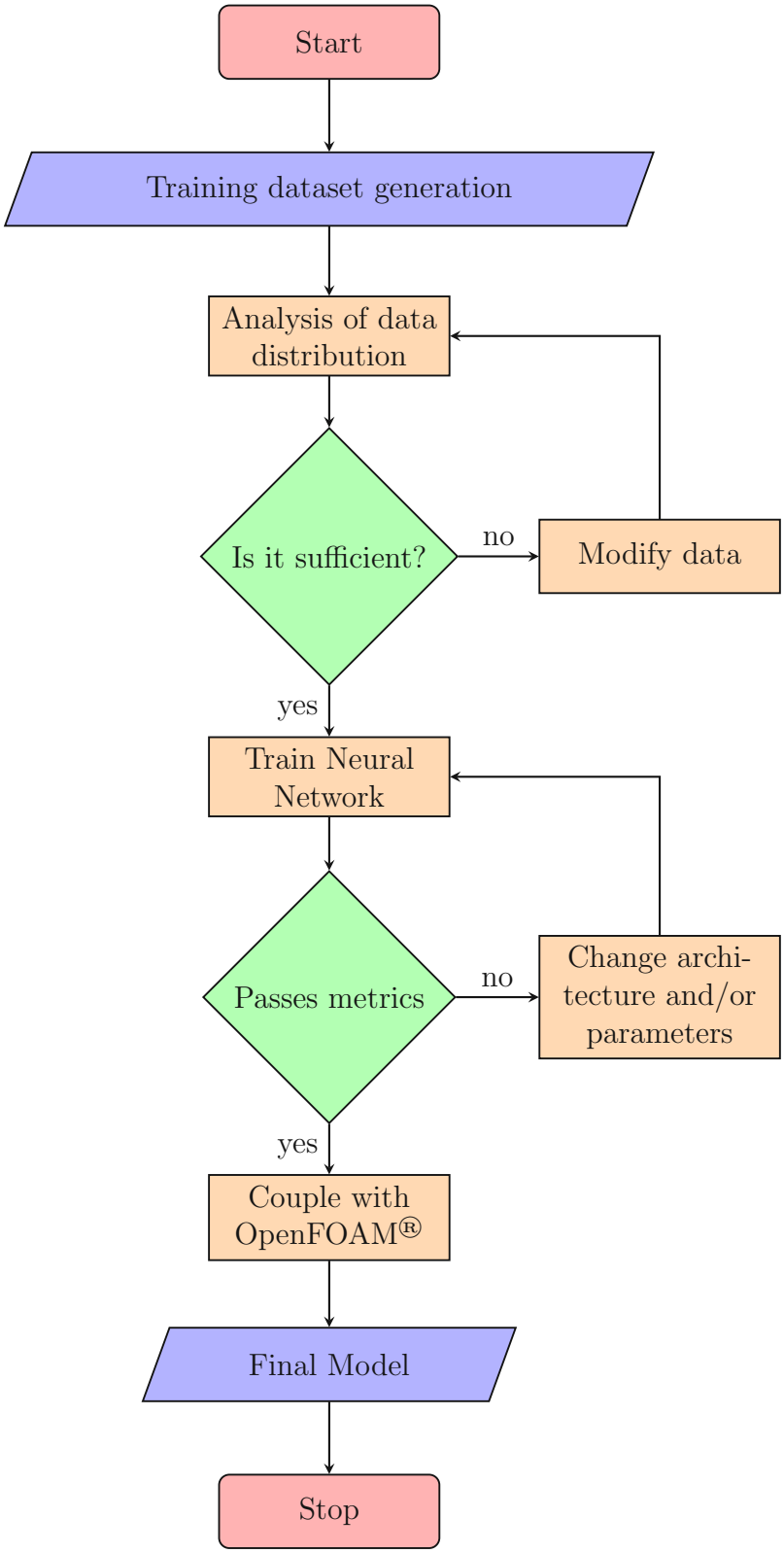


Figure 4.1: Proposed Methodology

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.2 Preliminaries

4.2.1 Computational Tools

The work presented on this thesis was developed through the use of the succeeding programming languages and libraries:

- Python 3.7.7
- Cantera 2.4.0
- SciPy 1.5.0
- Pandas 1.0.5
- PyTorch 1.6.0
- Comet.ML 3.1.17
- OpenFOAM® 7
- Matplotlib 3.2.2

Python 3

A broad portion of this work requires implementation of the computational algorithms required to train, test and evaluate the proposed Neural Network architectures. Said implementation involves the design of new specific tools and functions, with ***Python-3*** chosen as the programming language with which the aforementioned function are written. This language was picked for the easiness with which it allows for fast prototyping, and for the wide availability of libraries built with scientific purposes in mind. Python was designed by Guido van Rossum in 1991 [108].

Cantera

Cantera [109], developed by Dave Goodwin from the California Institute of Technology is an open-source suite of programs designed for the evaluation of gaseous reactions thermodynamical and kinetic properties, chemical source terms and governing equations. Designed with an object-oriented approach, it provides useful modeling interfaces for the C++, Matlab, Fortran and Python programming languages. Its main use throughout this work is providing a chemical properties database capable of computing the different species reaction rates of the species transport equations for the Plug Flow Reactor during the sample data generation phase.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Pandas

Pandas [110, 111] is an open source software library written in the C programming language and authored by Wes McKinney in 2008, while working at AQR Capital Management. The library provides the Python language with useful tools for fast data array manipulation with fancy indexing. Its intended use for this work is to allow for the easy handling of chemical species evolution train data, formatting and storage.

SciPy

SciPy [112] is an open-source collection of numerical routines used for scientific computing applications in Python. Among its many modules, it provides numerical integrators for ordinary differential equations, including stiff problems. These solvers will be used to evaluate the system of ODE's resulting from the PFR and which will be deployed as training data for the Neural Network.

PyTorch

PyTorch [113] is an open source package capable of providing the Python programming language with high-level machine learning features. It was released in 2016 by Facebook's Artificial Intelligence Research lab (FAIR), building on the code of the original Torch library [114], originally designed for the Lua language, and which ceased active development in 2018.

Among two of the most important aforementioned high-level features are its custom ***tensor*** data type which allows for enhanced operations between multi-dimensional arrays accelerated via the use of graphic processing units (CPU), and its ability to construct Neural Networks dynamically with reverse-mode auto-differentiation.

The implementation of the Neural Network, as well as its training and weight adjustment will be handled almost in its entirety by Pytorch utilities. This process will be discussed in detail on the chapter pertaining to methodology.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Comet.ml

Comet.ml is a cloud-based machine learning experimentation service [115]. Its Application Programming Interface (API) imparts many machine learning libraries -Pytorch among them- with the ability to log, track and record experimetal results, as well as practical visualization tools. All experiments and data generated during the Neural Network's training phase of this work will be stored via this platform.

OpenFOAM®

OpenFOAM® [116], acronym for ***Open-source Field Operation And Manipulation*** is a C++ class library designed for the development of custom Computational Fluid Dynamics (CFD) numerical solvers.

OpenFOAM® -7 will be used to solve the Navier-Stokes quations for the turbulent reacting flow after coupling with the Artificial Neural Network.

4.2.2 Hardware

For a correct understanding of the results obtained, it is necessary to report the conditions of the equipment on which the training and testing of the Neural Network models were performed. Table 4.1 displays information relevant to the hardware used during this work.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Table 4.1: Hardware Specifications

Component	Decription
Model	Dell Notebook G3 3579 (086F)
Operating System	Ubuntu 20.04.1 LTS Focal Fossa
Linux Kernel	5.4.0-42-generic
CPU	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz
CPU Cores	4
CPU Threads	8
Graphics Processing Unit (GPU)	Nvidia GP107M [GeForce GTX 1050 Mobile]
Driver Version	430.64
CUDA[1] Version	10.1
CUDA Cores	640
Random Access Memory (RAM)	8 GiB SODIMM DDR4 Synchronous 2667 MHz

4.3 Sample Data Generation

For the training, testing and validation of an ANN it is needed to generate a proper data set in such a way that the network can perform supervised learning from it, while keeping grounded on the physics of reality.

In lay terms, this means that it would not make sense to train a network to predict values that are not possible in real life. The intended way to generate the data samples is to solve the system of stiff ordinary differential equations and use the evolution steps of the numerical solver as input and label sets.

A number N of initial conditions is created by using a seeded random number generator, and then, after setting the proper ODE system via Cantera functionality they are integrated with the numerical solvers available in the SciPy library for a given time-step until a certain convergence criteria is reached. Figure 4.2 shows a flowchart of how this process is done.

[1]Compute Unified Device Architecture (CUDA)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

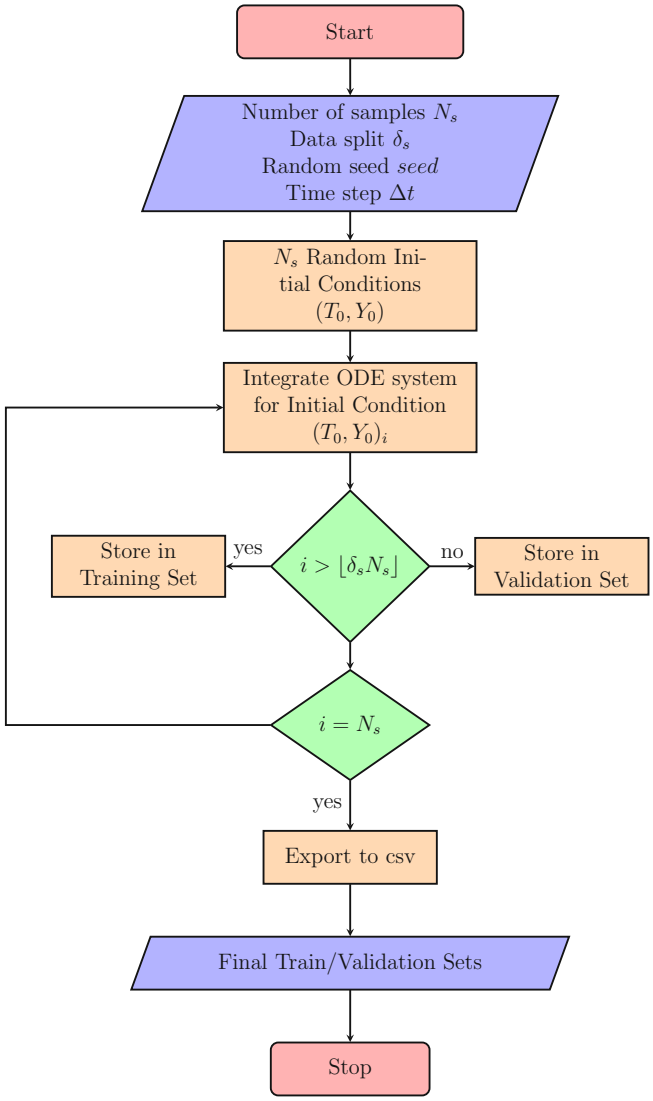


Figure 4.2: Training Data Generating Procedure

The manner in which initial conditions are created, and how these conditions will be developed via the ODE system will be discussed in detail in further subsections. All data points are stored and exported to csv files by using Pandas dataframe functions. The input-output format in which they are stored and feed to the Neural Network for the training procedures has the following structure (Figure 4.3):

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$\textbf{Input} \qquad \textbf{Output/Label}$$

$$x = \begin{bmatrix} T(t) \\ Y_1(t) \\ \vdots \\ Y_N(t) \end{bmatrix} \quad y = \begin{bmatrix} Y_1(t+\Delta t) \\ \vdots \\ Y_N(t+\Delta t) \end{bmatrix}$$

Figure 4.3: Training Data Structure

4.3.1 Random Initial Conditions

Reaction Mechanism

The whole process begins by choosing a reaction mechanism for the combustion's underlying chemistry. This choice is of utmost importance as it will determine many of the further phases of the work. It provides information about the chemical species and reactions involved in the combustion phenomena. Reaction Mechanism use the **cti** extension for file storage. Cantera provides functionality to read the mechanism and evaluate all the needed thermochemical properties.

The chosen mechanism was published by Laubscher [18] in 2017 based on the work presented by Westbrook and Dryer in 1984 [117]. It consists of four elements: O, H, C and N, seven chemical species: $CH_4, O_2, H_2O, N_2, CO, CO_2$ and H_2. These species are involved in a set of five reactions:

$$CH_4 + \frac{3}{2}O_2 \rightarrow CO + 2H_2O \tag{4.1}$$

$$CO + \frac{1}{2}O_2 \rightarrow CO_2 \tag{4.2}$$

$$CO_2 \rightarrow \frac{1}{2}O_2 + CO \tag{4.3}$$

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O \tag{4.4}$$

$$H_2O \rightarrow \frac{1}{2}O_2 + H_2 \tag{4.5}$$

It also details a series of thermodynamical parameters needed for the calculation of reaction rates. Table lists these quantities.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Table 4.2: Reaction Mechanism Thermodynamical Properties

Equation	Activation Energy J/Kg	Pre-Exponential Factor	Reaction Order
4.1	2.000×10^8	5.030×10^{11}	$[CH_4]^{0.7}[O_2]^{0.8}$
4.2	1.703×10^8	2.240×10^{12}	$[CO][O_2^{0.25}][H_2O]^{0.5}$
4.3	1.703×10^8	5.000×10^8	$[CO_2]$
4.4	1.464×10^8	5.693×10^{11}	$[H_2][O_2]^{0.5}$
4.5	3.979×10^8	2.410×10^{14}	$[H_2O]$

Note that Equations 4.2 and 4.3, and Equations 4.4 and 4.5 represent the same equation forwards and in reverse.

Mixture Fraction

It is important that these randomly generated initial conditions agree with what is physically a possibility. Mathematically speaking, it is possible to generate an infinite number of initial mass fractions containing all the species involved in a given reaction mechanism, however, the existence of some of these might not be actually plausible since they could not agree with one or more conservation laws. In particular it is desired for the samples to comply with mass conservation.

One way this is feasible is by using Bilger's Mixture Fraction Formula [118]. It is based on the idea that under the assumptions of equal diffusivities for all species, elemental mass fractions are conserved even as the species containing them are being produced or consumed [36]:

$$Z_j = \sum_{i=1}^{N} \mu_{i,j} Y_i \tag{4.6}$$

where:

Z_j is the conserved mass fraction of element j, $\mu_{i,j}$ is the mass fraction of element j in species i, Y_i is the mass fraction of species i, N is the number of species present in the reaction.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Then, according to Bilger, for the mixing and combustion of two streams A and B (e.g. fuel and oxidizer), related by a mixture fraction ξ it holds:

$$Z_j = Z_{j,A}\xi + (1-\xi)Z_{j,B} \tag{4.7}$$

where:

ξ is the stream mixture fraction, $Z_{j,A}$ and $Z_{j,B}$ are elemental mass fractions of element j present in the streams A and B.

The generated mass fractions must also comply with the condition that their sum must be equal to the unit:

$$\sum_{i=1}^{N} Y_i = 1 \tag{4.8}$$

All initial conditions will be generated based on a randomly generated mixture fraction with a seed taken from an uniform distribution, as well as the individual elemental mass fraction of a fuel-oxidizer mixture.

Bilger's Definition

As was detailed above, Bilger's Mixture Fraction Formula provides a nice framework under which initial mass fractions can be created. Provided a random mixture fraction, and elemental mass fraction for two streams, Bilger's Equations could then be solved returning a vector of initial conditions as a result. To illustrate this properly, it is possible to take the previous mechanism (with seven species and four elements) as an example and develop the related system of equations for it.

The extant species in the mechanism are: $Y_{CH_4}, Y_{N_2}, Y_{CO}, Y_{CO_2}, Y_{H_2O}, Y_{O_2}, Y_{H_2}$; with the elements: C, H, O, N.

Thus it gives the following equations:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$1 = Y_{CH_4} + Y_{N_2} + Y_{CO} + Y_{CO_2} + Y_{H_2O} + Y_{O_2} + Y_{H_2} \tag{4.9}$$

$$Z_H = \mu_{H,H_2} Y_{H_2} + \mu_{H,H_2O} Y_{H_2O} + \mu_{H,CH_4} Y_{CH_4} \tag{4.10}$$

$$Z_C = \mu_{C,CO_2} Y_{CO_2} + \mu_{C,CO} Y_{CO} + \mu_{C,CH_4} Y_{CH_4} \tag{4.11}$$

$$Z_O = \mu_{H,H_2} Y_{H_2} + \mu_{H,H_2O} Y_{H_2O} + \mu_{H,CH_4} Y_{CH_4} \tag{4.12}$$

$$Z_N = \mu_{N,N_2} Y_{N_2} \tag{4.13}$$

Generalizing for a mechanism with m number of elements and n number of species, this can be rewritten as a linear system using matrix and vector notation:

$$\overbrace{\begin{bmatrix} 1 \cdots 1 \\ \mu_{1,1} \cdots \mu_{1,n} \\ \mu_{2,1} \cdots \mu_{2,n} \\ \vdots \\ \mu_{m,1} \cdots \mu_{m,n} \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{bmatrix}}^{x} = \overbrace{\begin{bmatrix} 1 \\ z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}}^{b} \tag{4.14}$$

$$A \in \mathbb{R}^{m+1,n}, x \in \mathbb{R}^n, b \in \mathbb{R}^{m+1}$$

Using the expressions described in the previous section, it is possible to generate the matrix needed to solve a linear system of equations for the initial conditions that agrees with Bilger's Definition.

To this end, the coefficients $\mu_{i,j}$ can be computed in the following way:

$$\mu_{i,j} = \frac{a_{i,j} E_j}{W_i} \tag{4.15}$$

where:

$a_{i,j}$ is the number of atoms of element j in species i, E_j is the atomic weight of element j and W_i is the molecular mass of species i.

As for the resulting vector b, it will be created using random mixture fractions of a fuel and an oxidizer. For the results to be reproducible, it is necessary to use a seeded

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

random number generator.

Solving the linear system

Once the system of equations is set up it is necessary to solve it for the species mass fractions. Unfortunately, for most mechanisms, this will be an undetermined system (i.e. it has more unknowns than equations) and thus, in the case it can be solved, this solution will not be unique. To deal with this issue there are two possibilities using either a least-squares or a non-negative least squares solver. Both will provide an approximation of the solution by minimizing the squared euclidean norm [119]:

$$\arg\min_{x} ||Ax - b||^2 = 0 \tag{4.16}$$

whilst non-negative least squares also implements the restriction [120]:

$$x_i \geq 0 \quad \forall x_i \in x \tag{4.17}$$

However, they differ in certain aspects, and thus it is necessary to analyze the behavior of the initial conditions generated with each of them, in order to select the appropriate method.

Distribution of Random Samples

As mentioned beforehand, it is neccesary to analyze the behavior of the data that is being generated to choose the appropiate method to solve the system defined by Equation 4.14, and/or if neccesary apply some correctives. Table 4.3 contains the parameters by which said samples were created.

Table 4.3: Parameters of random samples

Symbol	Description	Value
ξ_{min}	Minimum mixture fraction	0.1
ξ_{max}	Maximum mixture fraction	1
ξ	Random mixture fraction	$f \sim U(\xi_{min}, \xi_{max})$
samples	Number of samples	10000
seed	Seed for random number generator	600

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

In addition, the elemental mass fractions of an oxydizer and fuel streams must also be chosen. Table 4.4 details the components of both the oxydizer and fuel components defined for the Flame D[18].

Table 4.4: Fuel and Oxydizer Elemental Mass Fractions

Fraction	H	O	C	N
Fuel	0.03922422	0.19663877	0.11684286	0.64729
Oxydizer	1.03625×10^{-5}	0.23298543	3.194117×10^{-5}	0.766970811

It is possible to thus see a summary of the samples generated by both methods.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Table 4.5: Summary of Initial Conditions generated via Least Squares

	CH4	O2	H2O	N2	CO	CO2	H2
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	0.028624	0.074352	0.066783	0.701748	0.057898	0.063880	0.006714
std	0.032385	0.010772	0.009221	0.031124	0.011951	0.003690	0.003091
min	-0.026787	0.055507	0.050651	0.647298	0.037449	0.057566	0.001426
25%	0.000652	0.065029	0.058802	0.674810	0.047575	0.060692	0.004044
50%	0.028350	0.074444	0.066861	0.702012	0.057796	0.063848	0.006688
75%	0.056654	0.083657	0.074748	0.728631	0.068241	0.067073	0.009389
max	0.085281	0.092784	0.082560	0.755001	0.078805	0.070335	0.012121

Table 4.6: Summary of Initial Conditions generated via Non-negative Least Squares

	CH4	O2	H2O	N2	CO	CO2	H2
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.0000000	10000.000000
mean	0.085072	0.213178	0.0	0.701748	0.0	0.0	7.553364e-07
std	0.040575	0.009452	0.0	0.031124	0.0	0.0	4.399142e-07
min	0.015648	0.196642	0.0	0.647298	0.0	0.0	2.638229e-09
25%	0.050026	0.204997	0.0	0.674810	0.0	0.0	3.753678e-07
50%	0.084728	0.213258	0.0	0.702012	0.0	0.0	7.516056e-07
75%	0.120191	0.221342	0.0	0.728631	0.0	0.0	1.136090e-06
max	0.156058	0.229351	0.0	0.755001	0.0	0.0	1.524953e-06

As can be seen in Table 4.5, the regular method of least squares sometimes produces mass fractions which are negative. This contradict whay is physically correct and thus is an inconvenience that should be dealt with. On the other hand, it fails to produce initial conditions in which the mass fractions of the products are all zero -i.e. where only products are involved at the beggining-. Some early conducted testing showed this affected the learning process negatively, as the network performs poorly when confronted with such type of inputs.

From the information available in Table 4.6 it is possible to observe that in the solutions obtained via non-negative least square, with the exception of the fractions of molecular hydrogen, those of all other species have an affect very much as the one described in the above paragraph -only products-, and which could represent a possible solution for the shortcomings of Least Squares Method (LS).

A combined approach was thus pursued, in which both solvers are applied in tamdem by alternating between them. The negative values obtained with LS are set to zero, and the vector is normalized. The same procedure was applied to the hydrogen fractions obtained via Non-negative Least Squares Method (NNLS), as it was noted in early tests, that they too affected the performance of the network's predictions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

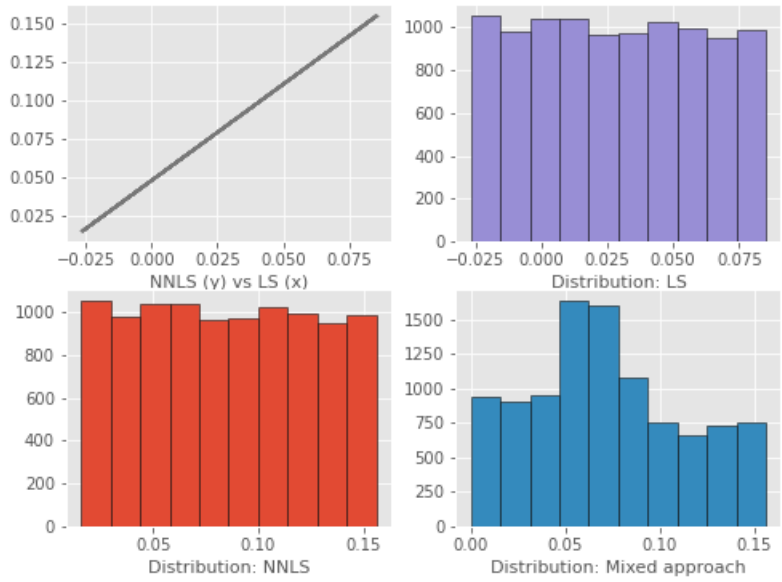


Figure 4.4: Distribution of CH_4 initial conditions

Figure 4.4 shows the distribution of methane initial mass fractions obtained through LS and NNLS. Approximately a quarter of all fractions resulting from LS are negative and thus cannot be used, while the remainder lie in the range 0-0.075. Those obtained via NNLS shows a uniform distribution of fractions between 0 and 0.15. A mixed approach shows an accumulation of samples between 0.05 and 0.075.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

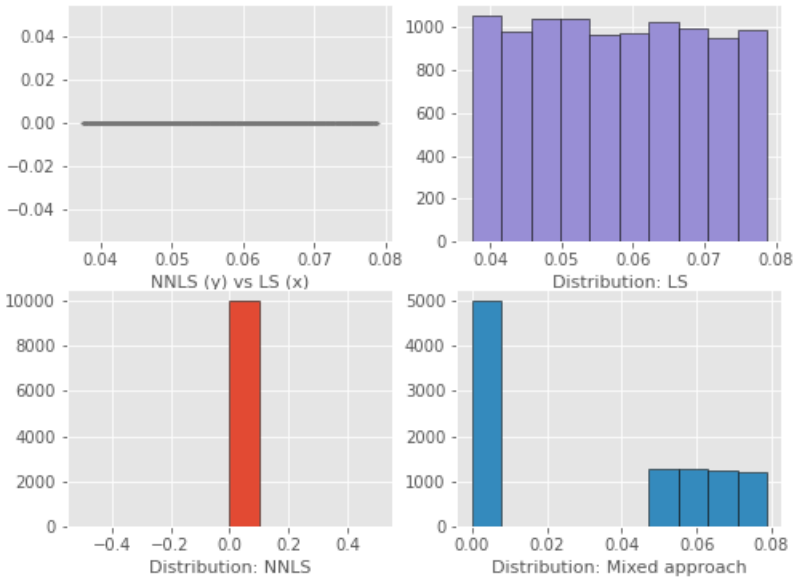


Figure 4.5: Distribution of CO initial conditions

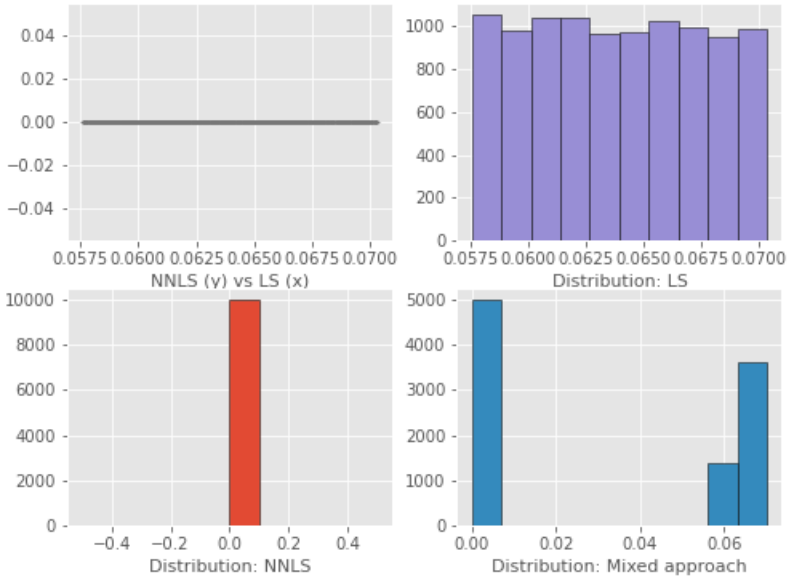


Figure 4.6: Distribution of CO_2 initial conditions

Figure 4.5 and Figure 4.6 show the distribution of initial mass fractions of Carbon

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Monoxide and Carbon Dioxide. Both show a similar behavior, having only values of zero when gathered through NNLS. The mixed approach shows both distributions separated, with a slight skew to the right for CO_2 caused by the size of the graph bins.

Figure 4.7 and Figure 4.8 show the distribution of initial mass fractions of molecular hydrogen and water. It can be seen that water mass fraction are only zero for NNLS, and the same method yields hydrogen mass fractions that are orders of magnitude smaller than those solved by LS and will be set to zero, to help with the ANN training process.

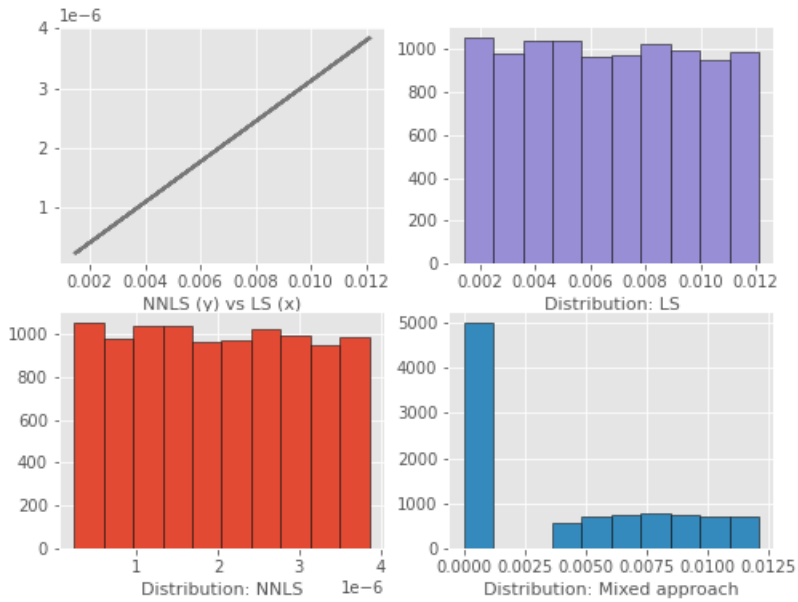


Figure 4.7: Distribution of H_2 initial conditions

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

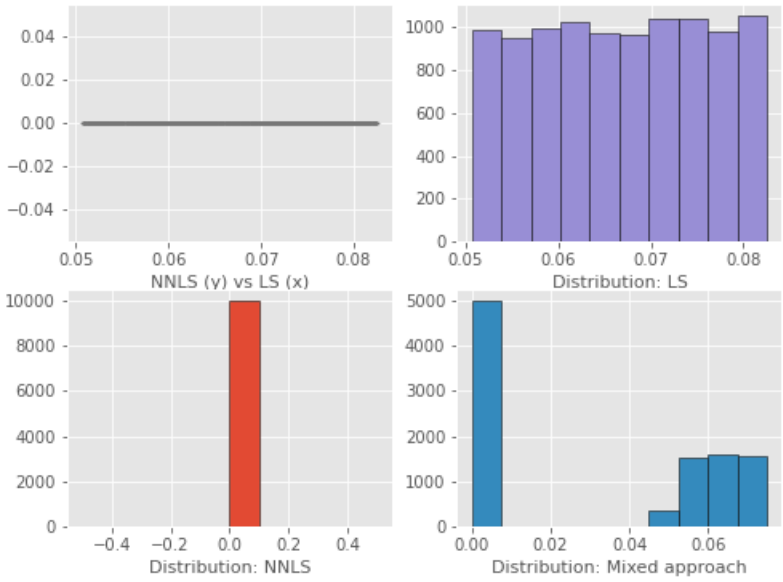


Figure 4.8: Distribution of H_2O initial conditions

Figure 4.9 and Figure 4.10 show the distribution of initial mass fractions of molecular nitrogen and oxygen. Nitrogen samples obtained through both methods are extremely similar, and it can be seen the relation betwen LS and NNLS fractions is close to the identity. Because of that, the mixed method follows an uniform distribution. Samples for oxygen have some displacement between them, this can be observed in the graphic for its mixed method fractions.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

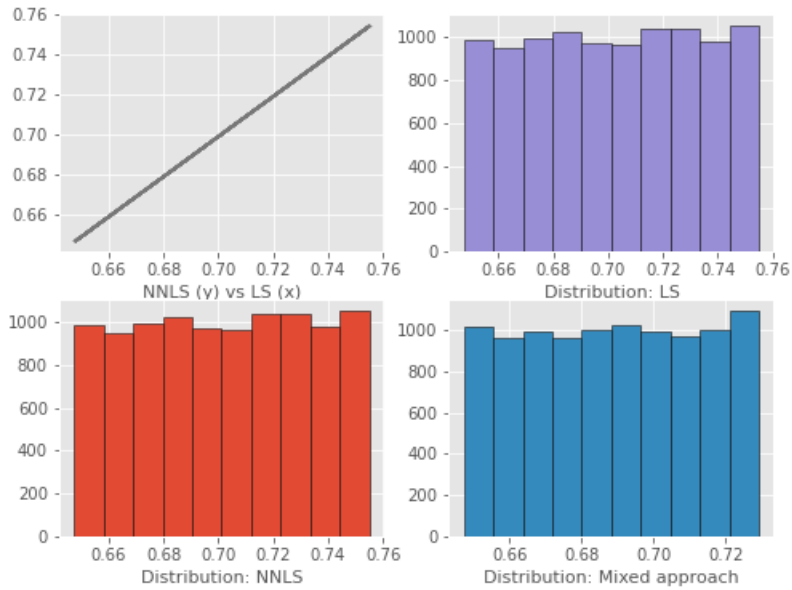


Figure 4.9: Distribution of N_2 initial conditions

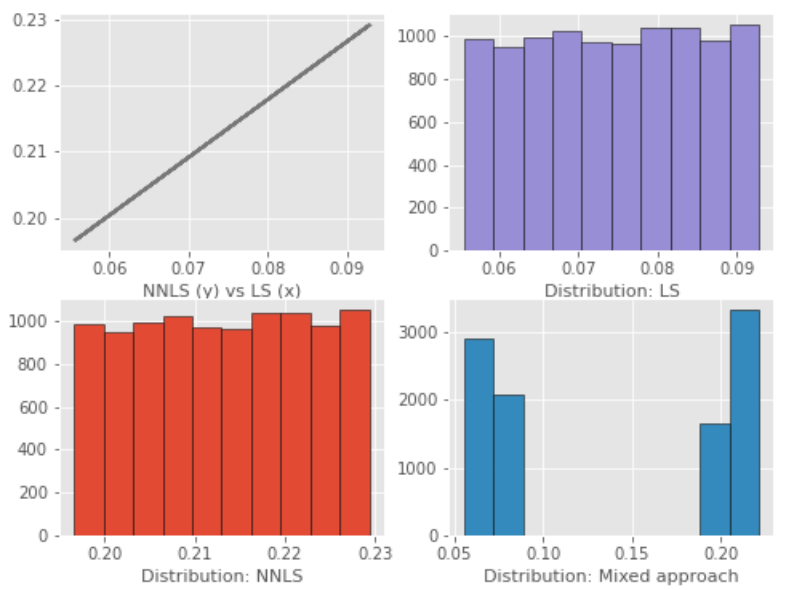


Figure 4.10: Distribution of O_2 initial conditions

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Temperature

There are really no conditions restricting the choice of a temperature, and since, a simpler procedure can be used to create them. All initial temperatures are generated randomly following a uniform distribution $T \sim U(T_{min}, T_{max})$ after providing minimimum and maximum temperature bounds T_{max} and T_{min}.

Before being feed to the ANN, temperatures are normalized according to the following procedure, -which leads to the normalized temperature T_n:

$$T_n = \frac{T - T_{min}}{T_{max} - T_{min}} \tag{4.18}$$

Normalizing data is not strictly needed, however, it can lead to a faster learning process [74].

4.3.2 Evolution of Chemical Species

Once a proper set of initial conditions is created, the next step in the process is setting up the corresponding ODE system and solve for the given conditions until ***steady state***. For experimental purposes, a bound needs to be provided, and so for the purposes of this thesis a reaction will be considered to have reached steady state when the following condition is met:

$$\left\| Y^{t+\Delta t} - Y^t \right\|_2 \leq 1 \times 10^{-9} \tag{4.19}$$

The ODEs solved for are those corresponding to the plug flow reactor, and are discretized via a backwards differentiation formula. Because of the stiffness of the system, the solution is obtained by using scipy's interface to the FORTRAN solver VODE .

All data points are stored in *.csv files following the format illustrated in Figure 4.3. An internal counter keeps track of the number of solved initial conditions, and once the threshold of samples set for the training data set is passed, it stores them on different files constituting the validation set.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

4.4 Neural Network Training and Evaluation

4.4.1 Architecture Selection

A traditional fully connected feed-forward architecture was selected for network, with the aim of testing several combinations of width and depth. Different networks were trained in order to approximate integrators with time steps of $1 \times 10^{-4}, 1 \times 10^{-5}$ and 1×10^{-6} using the back-propagation algorithm. The ***mean squared error*** was chosen as loss function, with the Adam algorithm chosen as optimizer. Initially, ReLU functions were set as activations for all outputing layers, which also provided the added benefit of enforcing the non-negativity of the mass fractions. However, during early testing ReLU networks performed poorly, and the activations were replaced instead for LeakyReLU functions which improved overall learning performance.

In order to select the appropiate combination of inner layers and neurons, both quantities were varied, layers being increased by one going from 1 to 5, and neurons increasing by powers of two, from 16 to 1024. Not all possible combinations were tested, as some might be slower than the traditional integration method, at which point they are no longer of interest. As an example, if a 3-layer network with 1024 neurons is already slower than the integration method, then it is not necessary to tests for 4 and 5 layers.

The Neural Networks are implemented and trained using Facebook's Pytorch library, with the results stored in Comet.ml's platform. Because of hardware restrictions, all networks were trained for 25 epochs and a bacth size of 500. If any given network showed potential, whenever possible, it was trained further, either increasing the number of epochs or decreasing the batch size.

The inner workings of this architecture are explained next in detail.

Input layer

An input vector is created using the temperature (initial or computed by solver) and species mass fractions (initials or approximated). The desired temperature and mass fractions should be concatenated in the following manner as shown in Equation 4.20.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Note that the temperature T here represent a normalized temperature T_n as shown in Equation 4.18.

$$x^{(0)} = concat(T, Y) = \begin{bmatrix} T & Y_{CH_4} & Y_{O_2} & Y_{H_2O} & Y_{N_2} & Y_{CO} & Y_{CO_2} & Y_{H_2} \end{bmatrix} \tag{4.20}$$

$$x^{(0)} \in \mathbb{R}^8$$

A linear mapping is applied to the input array in order to obtain a vector of size N_n which corresponds to the desired number of neurons in each layer.

$$y^{(0)} = x^{(0)} w_0^\intercal + b_0 \tag{4.21}$$

Where:

$$w_0^\intercal \in \mathbb{R}^{8 \times N_n} \text{ is the matrix of known trained weights}$$
$$b_0 \in \mathbb{R}^{N_n} \text{ is the vector of known trained biases}$$

After the transformation, a Leaky ReLU is applied element-wise to the vector y_0

$$\mathbb{R}^{64} \ni x^{(1)} = LeakyReLU(y^{(0)}) = LeakyReLU(x^{(0)} w_0^\intercal + b_0) \tag{4.22}$$

It is thus possible to move to the hidden layers.

Hidden Layers

All N_L hidden layers apply in succession a linear mapping that keeps the input size, followed by a LeakyReLU activation:

$$\mathbb{R}^{N_n} \ni x^{(i)} = LeakyReLU(x^{(i-1)} w_{N_{i-1}}^\intercal + b_{i-1}) \tag{4.23}$$

With:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

$$w_{i-1}^{\intercal} \in \mathbb{R}^{N_n \times N_n} \ , \ b_{i-1} \in \mathbb{R}^{N_n}$$
$$i = 2, \cdots N_L$$

Output Layer

A linear transformation is used in order to map the output of the previous layer into a space of the desired dimension -7-. The activation function is then applied.

$$\mathbb{R}^7 \ni x^{(out)} = LeakyReLU(x^{N_L} w_{N_L}^{\intercal} + b_{N_L}) \tag{4.24}$$

With:

$$w_{N_L}^{\intercal} \in \mathbb{R}^{N_n \times 7} \ , \ b_{N_L} \in \mathbb{R}^7$$

Figure 4.11 illustrates the model flow for a network of 3 hidden layers and 64 neurons.

$$x^{(0)} = concat(T, Y); \ x^{(0)} \in \mathbb{R}^8$$
$$\downarrow$$
$$x^{(1)} = LeakyReLU(x^{(0)} w_0^{\intercal} + b_0); \ x^{(1)}, b_0 \in \mathbb{R}^{64}, w_0^{\intercal} \in \mathbb{R}^{8x64}$$
$$\downarrow$$
$$x^{(2)} = LeakyReLU(x^{(1)} w_1^{\intercal} + b_1); \ x^{(2)}, b_1 \in \mathbb{R}^{64}, w_1^{\intercal} \in \mathbb{R}^{64x64}$$
$$\downarrow$$
$$x^{(3)} = LeakyReLU(x^{(2)} w_2^{\intercal} + b_2); \ x^{(3)}, b_2 \in \mathbb{R}^{64}, w_2^{\intercal} \in \mathbb{R}^{64x64}$$
$$\downarrow$$
$$x^{(out)} = LeakyReLU(x^{(3)} w_3^{\intercal} + b_3); \ x^{(out)}, b_3 \in \mathbb{R}^7, w_3^{\intercal} \in \mathbb{R}^{64x7}$$

Figure 4.11: Data flow for 5-layer 64-neuron feed forward network.

4.4.2 Training and Testing

The Neural Networks are trained with a labeled data set with the help of the back propagation algorithm. During each training iteration, a number of samples -***batch***

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

size- is evaluated by the network, and the result compared with the same amount of desired targets in the train set with the help of a selected ***loss function***(MSE) -this constitutes the forward propagation-. The use of the MSE provides a scalar measure of the error, and such provides useful information on how to adjust all the networks parameters to improve its performance via ***gradient descent***, in which all aforementioned parameters are changed by a quantity that is proportional to the derivative of the error with respect to each individual parameter. This last part corresponds to the back propagation. It is worth noting, that before the error is back-propagated, the validation set is also evaluated and its loss reported, however this loss obtained from the validation examples is not used in the correction of the parameters.

This procedure is iterated until all existing examples have been evaluated or "seen" by the network, at which point the overall training scheme is said to have completed one ***epoch***. The training continues until the desired number of epochs have been reached. After each inner and outer iteration, all error metrics are reported and stored online using Comet.ml.

As for the testing of the different networks, a special method must be devised to accomadate for prediction error progating during species evolution. In some of the works dwelling into the use of neural networks for combustion applications, as well as in other common machine learning applications -such as image recognition-, a portion of the overall available data is set apart and then evaluated by the network using the same loss function. However this is only valid for a setting in which a single prediction is needed based on real data. In the application selected in this thesis, a network should be able to predict the evolution of chemical species over a whole chosen time frame, where only the initial condition is known and all further data is created based on values resulting from previous predicitions, and which those already carry an approximation error. Figure 4.12 serves to illustrate this point -note that in both cases, the temperature is not predicted but solved for-.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Standard prediction procedure:

$$\text{Input:}Y^{(0)} \quad \text{Target:}Y^{(1)} \quad \text{Prediction:}\hat{Y^{(1)}} = ANN(Y^{(0)}, T^{(0)})$$

$$\vdots$$

$$\text{Input:}Y^{(t)} \quad \text{Target:}Y^{(t+1)} \quad \text{Prediction:}\hat{Y}^{(t+1)} = ANN(Y^{(t)}, T^{(t)})$$

Intended prediction procedure:

$$\text{Input:}Y^{(0)} \quad \text{Target:}Y^{(1)} \quad \text{Prediction:}\hat{Y}^{(1)} = ANN(Y^{(0)}, T^{(0)})$$

$$\vdots$$

$$\text{Input:}\hat{Y}^{(t)} \quad \text{Target:}Y^{(t+1)} \quad \text{Prediction:}\hat{Y}^{(t+1)} = ANN(\hat{Y}^{(t)}, T^{(t)})$$

Figure 4.12: Difference of prediction between regular ANN and those used as integrators

A framework is proposed in which a number of random initial conditions are created -1000- with a different seed and evaluated until steady state. The resulting temperatures are collected, and together with the initial conditions are used as input by the networks to create the corresponding time series. Then, the real data curves are compared with the predicted ones for a more accurate computation of the error. The mean error is reported, and the error expansion is plotted for every sample for visualization purposes. Additionally, a set of three initial mass fractions and temperatures are evaluated and plotted to provide visual confirmation of whether the network is working as intended or not.

These three initial conditions are the following:

Table 4.7: Initial conditions chosen for visualization purposes

Name	Temp	Y_{CH_4}	Y_{O_2}	Y_{H_2O}	Y_{N_2}	Y_{CO}	Y_{CO_2}	Y_{H_2}
Y_1	1880	0.15607	0.19664,	0.0	0.64729	0.0	0.0	0.0
Y_2	1781.7965	0.0074	0.0814	0.0728	0.7222	0.0501	0.0615	0.0047
Y_3	800	0.02483	0.07562	0.06786	0.70539	0.05649	0.06345	0.00635

During the subsection on architecture selection it was mentioned how LeakyReLU functions were chosen instead of ReLU ones because of poor performance during train-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

ing. This however had the inconvenience of failing to enforce the non-negativity of the species. Complicating matters, is the fact that all resulting output values from the network must add up to one. Initially, approaches design to comply with these condition were implemented, first by hard-enforcing it with a normalizing layer, and then with a soft-enforcing by adding it to the loss function. However, both approaches failed to give any significant result, as during the training phase little to no learning took place. A compromise was reached in which all efforts where devoted to select an architecture with the smaller reported validation error, hoping the network would learn by itself these restrictions and deviate from them in a minimum way, and then during the deployment part change the activation function of the output layer for a ReLU, followed by a normalization of the fractions. As can be seen in the chapter dealing with results, this approach was able to provide reasonable and accurate results despite these issues.

4.5 Turbulent Combustion Coupling

To test the networks created after the training procedure, selected architectures will be compared to the well known combustion data published by the Sandia National Laboratory, particulary, they will evaluated against their ***Flame D*** [121].

The Sandia D [123, 124] flame is a turbulent piloted methane/air jet flame with a Reynolds number of 22400, nozzle diameter of 7.2 mm, and pilot diameter of 18.2 mm. This flame burns a mixture consisiting of 25% CH_4 and 75% dry air (79% N_2, 21% O_2), the elemental mass fractions corresponding to these streams are presented in Table 4.4.

Radial and axial profiles -x/d- are reported for the mass fractions corresponding to $CH_4, CO, CO_2, H_2, H_2O, N_2$ and O_2. Details of the full experimental setup are shown in Barlow [121].

A transient simulation was conducted for 0.5 seconds, after which steady state is assumed, as thermophysical quantities do not change further. OpenFOAM® -7 was used to solve the flow, with a mesh consisting of 4970 cells, which agrees with grid convergence studies carried out by Wartha [125].The EDC and $\kappa - \epsilon$ models are used

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

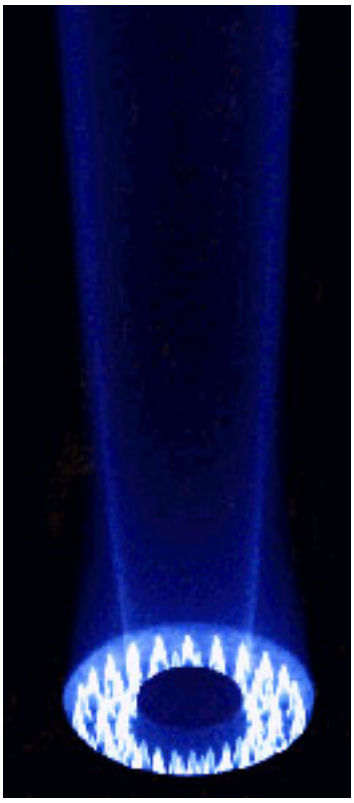


Figure 4.13: Sandia Flame D geometry
Source: Sandia National Laboratory [122]

for turbulence closures.

The integration time was approximated via two trained neural networks, one for a step size of 1×10^{-4} seconds, and the other for 1×10^{-5} seconds. The first is applied initially, while the second is used to get closer to the actual integration time. The networks are only utilized in cells whose mixture fractions is above 0.1, and whose temperature is higher than 300K.

The simulation is also carried out with a standard ODE integrator, in order to compare the result with those obtained using the ANN and the experimetal data.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5 Results and Discussion

5.1 Training Results

Over a thousand training experiments were explored in order to find the right combination of data samples - defined by time-step size, temperature/mixture fraction ranges and steady state condition -; architectures -neurons, layers, activation, loss function-; and hyperparameters: batch size, epochs, learning rate.

Here are reported results for 40 experiments run each for different data sets for the various time steps (for a total of 120 experiments). These sets were created by adjusting the time-step in the VODE solver while evaluating initial conditions until steady state.

The data sets are built starting from 10000 random initicial conditions. For a more detailed explanation please see Section 4.3. The paramters used in the dataset paramteres are listed in Table 5.1

As for the networks, some architecture combinations are presented with the number of hidden layers ranging from 1 to 6 and the number of neurons in each ranging from 8 to 1024 in multiples of two. All networks utilize LeakyReLU as activation, and were

Table 5.1: Dataset parameters

Parameter	Value
Random Seed	600
ξ_{min}	0.1
ξ_{max}	1.0
T_{min}	300
T_{max}	3000
Tolerance	1×10^{-9}
Δt	$1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}$
Stopping time	$1000\Delta t$

trained for 25 epochs with batch size 500. The learning rate was changed dynamically via one-cycle policy, with max value of 1×10^{-3}.

In order to compare the experimetal networks with the traditional ODE integration method. All network operations were converted from Pytorch classes to ones using Numpy functions to test them against Numpy's implementation of VODE .

The following values obtained for every network training process are shown in Tables 5.2, 5.3 and 5.4 in order: number of layers, number of neurons, mean squared error (MSE), error deviation (std), elapsed time in seconds, the required time to evaluate a 1000 samples with the VODE solver (Solver Time) and with a neural network (Network Time), validation loss, speed up (Time percentage), and a measure of the error propagation.

The speed up is computed as the ratio (Network Time)/(Solver Time) while the error propagation is calculated as MSE /(validation loss).

5.1.1 Step size: 1×10^{-4}

The following are training results of networks intended to replace an integrator of step-size 1×10^{-4}.

Figure 5.1 shows the computing time of several networks as a fraction of the time taken by the VODE solver. It is possible to see that as the number and layers and neurons increase, the added complexity also increases the evaluation time of an input. Networks with 6 and 7 hidden layers are only faster than VODE when having at most 64 neurons. Those with 3, 4 and 5 layers stop being effective after 128 neurons, and those with 2 after 256. From the figure it is possible to infer that the only possible modificatins to architectures that could be tested while still hoping for performance improvements are: increasing the number of layers keeping the number of neurons smaller than 64, or increasing the number of neurons for a 1-layered network.

Figure 5.2 and Figure 5.3 show mean squared error MSE and error deviation STD over 1000 samples. In general it seems that both the error and its deviation tend to decrease as the number of neurons in each hidden layer grows, but oscillations can

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

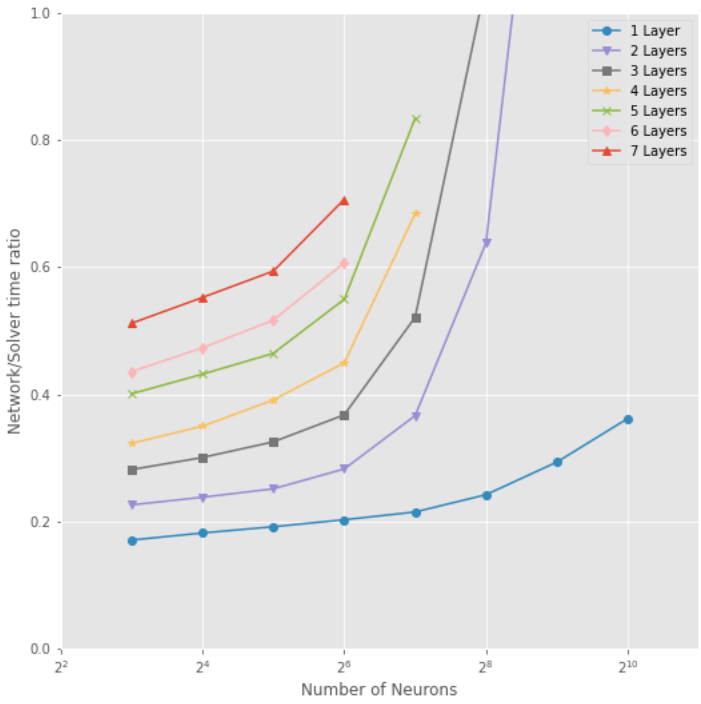


Figure 5.1: Speedup for step size 1×10^{-4}

occur, meaning it is possible for a model with less neurons to be more accurate than one with more. The relation between these two values and the number of layer seems to follow a similar behavior.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

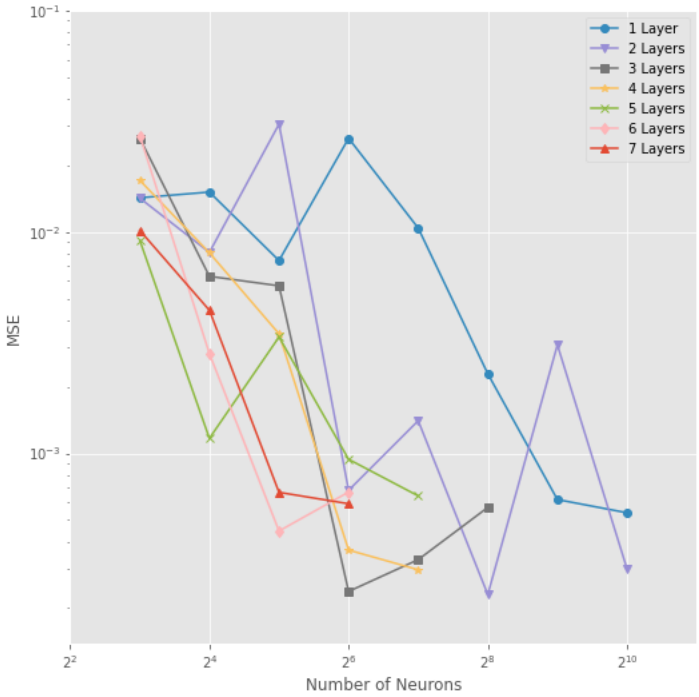


Figure 5.2: Mean Squared Error for step size 1×10^{-4}

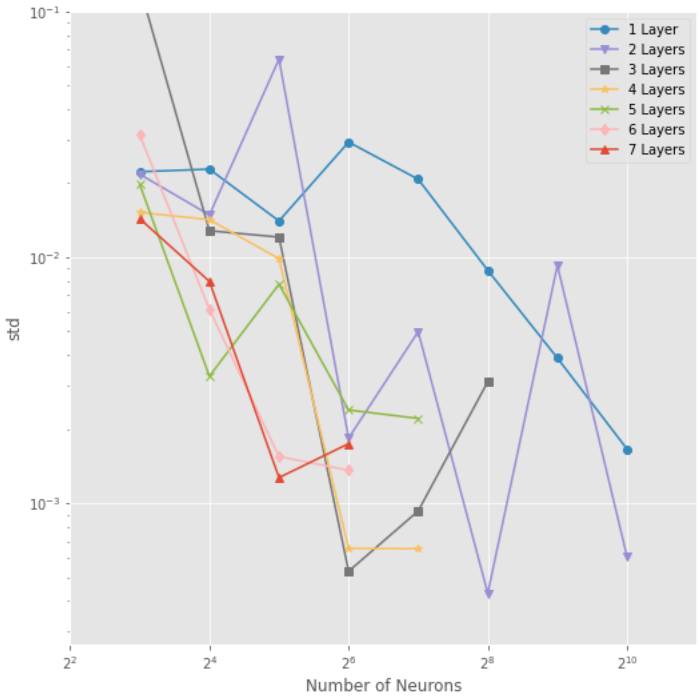


Figure 5.3: Error deviation for step size 1×10^{-4}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

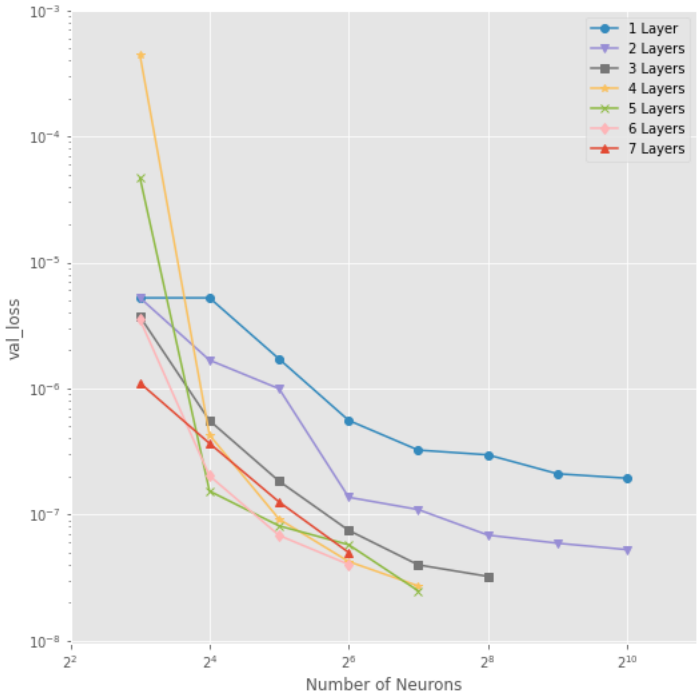


Figure 5.4: Validation Loss for step size 1×10^{-4}

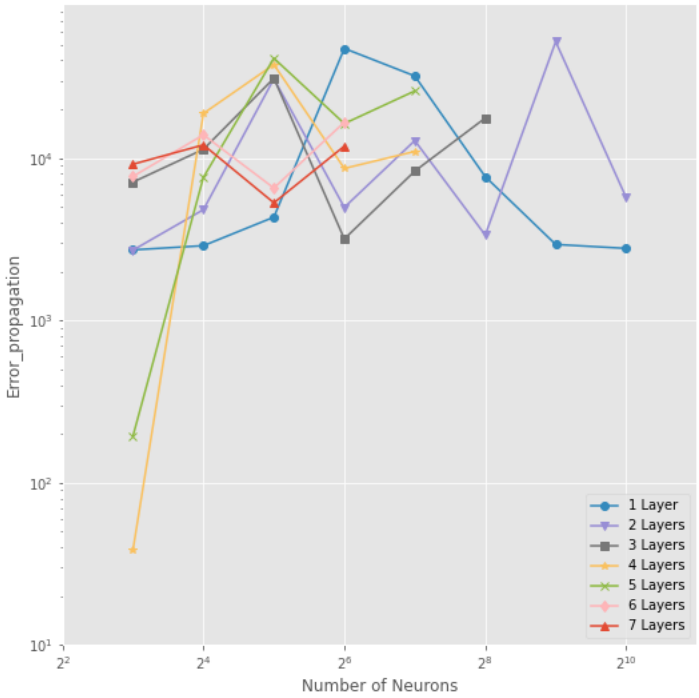


Figure 5.5: Error growth for step size 1×10^{-4}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The losses, or MSE obtained from the evaluation of the validation set can be seen in Figure 5.4. The loss appears to diminish inversely to the number of neurons and layers. An interesting observation when compared to Figure 5.2 and Figure 5.3 shows that in general a smaller validation loss does not imply smaller MSE and STD.

Figure 5.5 is useful to visualize the relation between MSE and validation loss, and thus the way error grows. With the exception of two points for networs with 4 and 5 layers with 8 neurons (which are outliers caused by high validation losses), one can see that the error can grow upwards up 10.000 times. An open problem would be on how to reduce MSE while keeping validation losses in the same ranges, as one runs the risk of overgeneralizing when reducing the validation loss significantly.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Table 5.2: Training results for step size 1×10^{-4}

	Layers	Neurons	MSE	STD	Elapsed Time (s)	Solver Time (s)	Network Time (s)	Validation loss	Time percentage	Error propagation
0	1.0	8	0.014243	0.022270	585.524014	34.270482	5.873770	5.233123e-06	0.171394	2721.784168
1	1.0	16	0.015116	0.022812	648.476230	33.728686	6.155044	5.233388e-06	0.182487	2888.317756
2	1.0	32	0.007418	0.014043	680.745593	34.053096	6.548119	1.712794e-06	0.192291	4330.758397
3	1.0	64	0.026438	0.029384	672.506701	35.617251	7.240925	5.576720e-07	0.203298	47407.279457
4	1.0	128	0.010396	0.020751	678.978993	35.360059	7.619250	3.241313e-07	0.215476	32073.175693
5	1.0	256	0.002286	0.008845	673.455579	35.017744	8.495790	2.972427e-07	0.242614	7689.690968
6	1.0	512	0.000618	0.003890	671.932599	34.087510	10.017442	2.103558e-07	0.293874	2939.716827
7	1.0	1024	0.000539	0.001660	680.149917	36.714068	13.303500	1.938990e-07	0.362354	2779.834030
8	1.5	7	0.035791	0.025844	587.277004	37.641343	5.292718	5.392394e-06	0.140609	6637.344959
9	2.0	8	0.014192	0.021747	672.401893	36.538600	8.281658	5.233036e-06	0.226655	2712.035705
10	2.0	16	0.008076	0.014862	714.862072	34.240389	8.171269	1.676756e-06	0.238644	4816.485951
11	2.0	32	0.030627	0.063882	727.275179	37.213857	9.373306	9.945479e-07	0.251877	30795.003413
12	2.0	64	0.000681	0.001838	752.968507	37.674974	10.665823	1.369155e-07	0.283101	4972.395326
13	2.0	128	0.001403	0.004981	754.046544	37.905755	13.903596	1.093845e-07	0.366794	12829.109112
14	2.0	256	0.000231	0.000430	767.976143	37.061304	23.655054	6.864069e-08	0.638268	3359.120850
15	2.0	512	0.003086	0.009267	815.472697	37.084699	59.281145	5.915942e-08	1.598534	52156.803363
16	2.0	1024	0.000301	0.000607	1149.116247	35.752486	514.990106	5.259907e-08	14.404316	5724.570802
17	3.0	8	0.026447	0.125667	774.900533	33.496305	9.450424	3.714979e-06	0.282133	7119.076575
18	3.0	16	0.006298	0.012815	813.026883	33.617709	10.120915	5.555101e-07	0.301059	11336.683771
19	3.0	32	0.005704	0.012075	806.992874	33.283093	10.841054	1.836608e-07	0.325723	31054.831407
20	3.0	64	0.000238	0.000528	807.775104	33.562359	12.354221	7.481066e-08	0.368098	3180.300690
21	3.0	128	0.000331	0.000929	808.589356	33.771101	17.609504	3.982012e-08	0.521437	8316.197828
22	3.0	256	0.000570	0.003117	835.025112	37.452276	39.217352	3.226583e-08	1.047129	17659.683737
23	4.0	8	0.017078	0.015204	808.570298	35.660258	11.531768	4.420500e-04	0.323379	38.634106
24	4.0	16	0.008039	0.014221	853.837950	34.933336	12.236397	4.264512e-07	0.350279	18850.838121
25	4.0	32	0.003464	0.009862	849.883182	35.978666	14.088842	9.185314e-08	0.391589	37710.409865
26	4.0	64	0.000366	0.000655	888.549038	34.492001	15.517962	4.231713e-08	0.449900	8642.947640
27	4.0	128	0.000298	0.000652	902.969200	36.460056	24.940570	2.711743e-08	0.684052	10995.261276
28	5.0	8	0.009142	0.019737	882.807317	37.440020	15.020239	4.706577e-05	0.401181	194.241984
29	5.0	16	0.001169	0.003282	925.178757	36.368275	15.712414	1.533102e-07	0.432036	7621.897807
30	5.0	32	0.003350	0.007768	954.114815	34.707282	16.133080	8.119441e-08	0.464833	41257.836727
31	5.0	64	0.000940	0.002392	953.989837	35.205379	19.353359	5.755646e-08	0.549727	16327.779681
32	5.0	128	0.000642	0.002210	971.576233	37.816620	31.546283	2.477408e-08	0.834191	25929.000552
33	6.0	8	0.027170	0.031408	1044.380768	34.892210	15.213702	3.520345e-06	0.436020	7718.117851
34	6.0	16	0.002818	0.006135	1001.024972	36.763042	17.404661	2.026612e-07	0.473428	13906.057117
35	6.0	32	0.000446	0.001547	978.492533	34.572075	17.869330	6.817046e-08	0.516872	6542.180728
36	6.0	64	0.000668	0.001361	971.825125	36.767126	22.315134	4.015456e-08	0.606932	16648.102911
37	7.0	8	0.010143	0.014334	1067.558576	34.983745	17.913267	1.105581e-06	0.512045	9174.553588
38	7.0	16	0.004423	0.007974	1084.060516	36.566014	20.205592	3.664188e-07	0.552578	12072.219672
39	7.0	32	0.000667	0.001270	1085.089096	34.803840	20.672726	1.256565e-07	0.593978	5305.426574
40	7.0	64	0.000592	0.001745	1092.655879	37.901155	26.788417	4.993414e-08	0.706797	11856.849123

5.1.2 Step size: 1×10^{-5}

The following are training results of networks intended to replace an integrator of step-size 1×10^{-5}.

Figure 5.6 shows the computing time of several networks as a fraction of the time taken by the VODE solver. It is possible to see that as the number and layers are bounded by the same constraints as shown for network of step size 1×10^{-4} (see Figure 5.1).

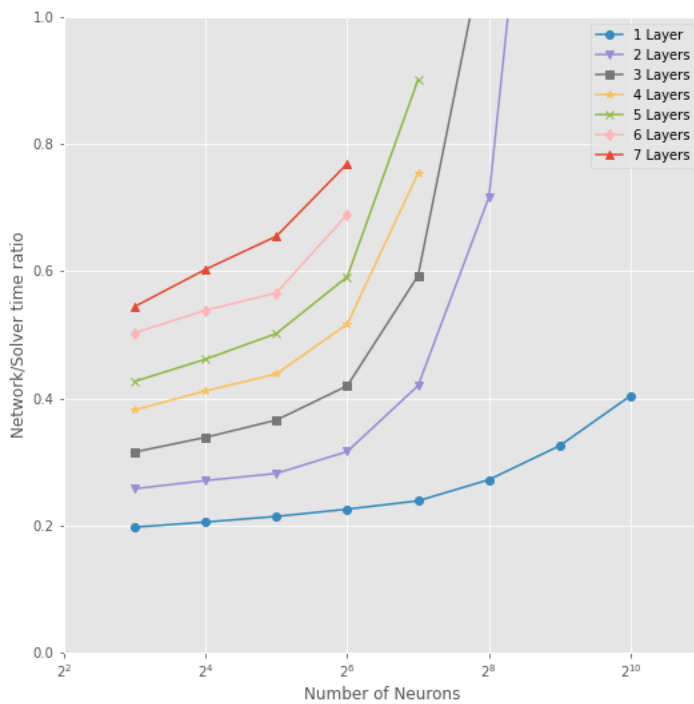


Figure 5.6: Speedup for step size 1×10^{-5}

Figure 5.7 and Figure 5.8 show MSE and STD over 1000 samples. Again, their behavior follows a similar pattern to the networks trained for step size 1×10^{-4}, with the noted difference that MSE and STD for most architectures seem to have decreased. Validation losses in Figure 5.9 seem to have in general also decreased. These phenomena might be explained by the fact that a smaller time-step provides a sort of higher "resolution" of the species' evolution, and/or that it might be easier for a network to learn the proper gradients for shorter time distances.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The error growth in Figure 5.10 seems to have increased, unlike MSE , STD, and validation loss. It could be related to MSE and validation losses decreasing at different rates.

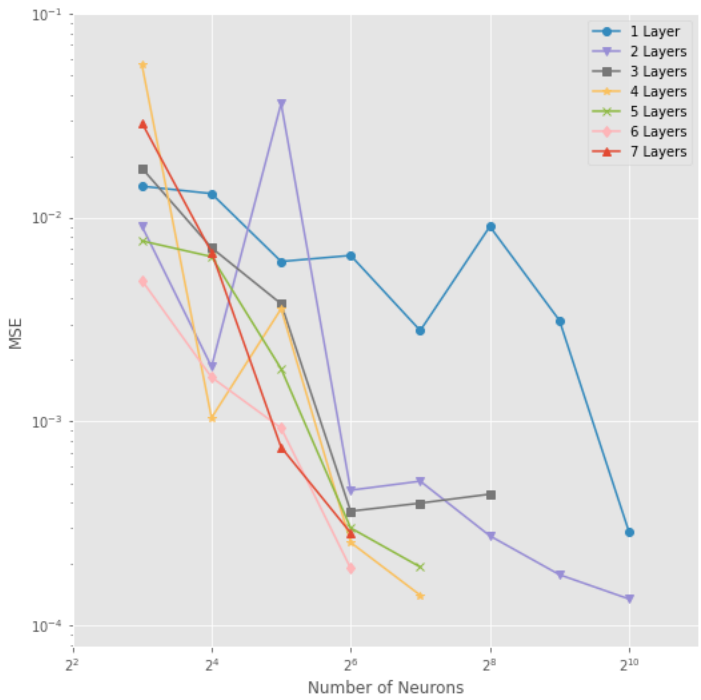


Figure 5.7: Mean Squared Error for step size 1×10^{-5}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

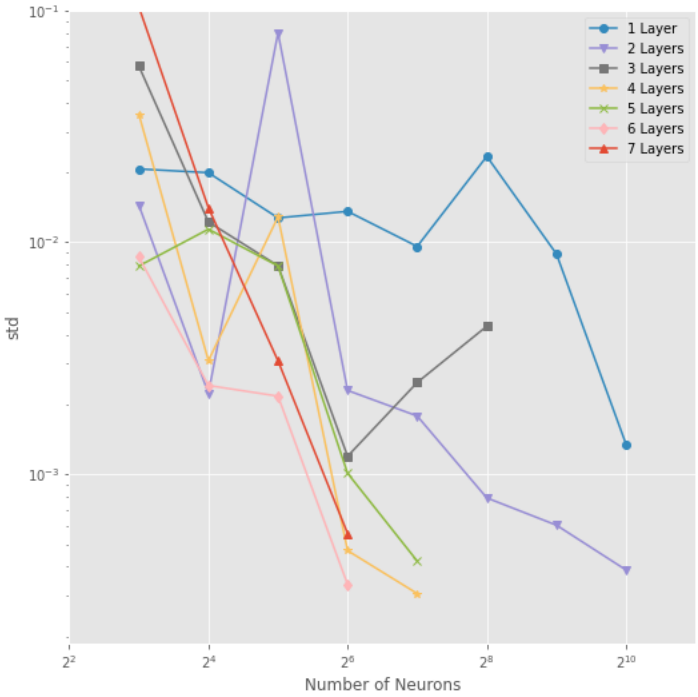


Figure 5.8: Error deviation for step size 1×10^{-5}

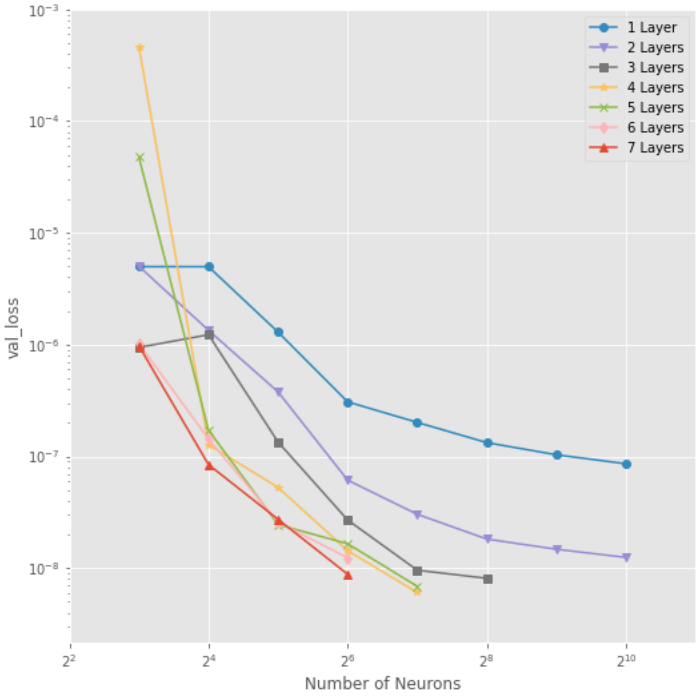


Figure 5.9: Validation Loss for step size 1×10^{-5}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

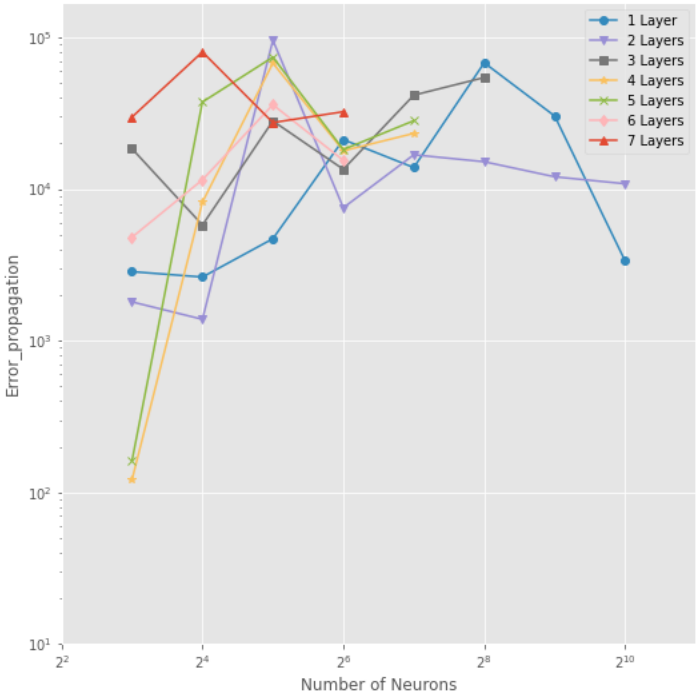


Figure 5.10: Error growth for step size 1×10^{-5}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Table 5.3: Training results for step size 1×10^{-5}

	Layers	Neurons	MSE	STD	Elapsed Time (s)	Solver Time (s)	Network Time (s)	Validation loss	Time percentage	Error propagation
0	1.0	8	0.014220	0.020707	734.413373	37.175095	7.355396	4.983819e-06	0.197858	2853.209056
1	1.0	16	0.013086	0.019955	762.917507	35.745976	7.362022	4.976484e-06	0.205954	2629.666196
2	1.0	32	0.006084	0.012740	785.435450	36.148113	7.763343	1.293580e-06	0.214765	4703.347526
3	1.0	64	0.006513	0.013595	802.726305	38.685339	8.750138	3.075153e-07	0.226187	21178.646298
4	1.0	128	0.002792	0.009600	805.150358	38.306903	9.166129	2.014558e-07	0.239281	13858.295616
5	1.0	256	0.009029	0.023467	804.342744	37.905159	10.328977	1.328131e-07	0.272495	67983.313800
6	1.0	512	0.003131	0.008917	802.316788	38.722693	12.621827	1.033648e-07	0.325954	30295.329365
7	1.0	1024	0.000288	0.001350	807.279842	36.853432	14.911884	8.588039e-08	0.404627	3355.778130
8	2.0	8	0.009054	0.014467	835.202468	36.643590	9.463626	5.020713e-06	0.258261	1803.349301
9	2.0	16	0.001858	0.002234	871.615039	38.979813	10.567852	1.342541e-06	0.271111	1384.026790
10	2.0	32	0.036144	0.079932	868.282271	37.307853	10.533530	3.774351e-07	0.282341	95761.732493
11	2.0	64	0.000461	0.002299	868.546416	37.068437	11.742358	6.126177e-08	0.316775	7527.034989
12	2.0	128	0.000511	0.001784	869.889959	35.466694	14.908819	3.034669e-08	0.420361	16822.815444
13	2.0	256	0.000276	0.000791	876.560965	37.683082	27.003347	1.820694e-08	0.716591	15135.889300
14	2.0	512	0.000178	0.000604	927.958166	38.180507	67.988840	1.476615e-08	1.780721	12041.819445
15	2.0	1024	0.000135	0.000387	1314.685730	36.605292	570.056680	1.247310e-08	15.573068	10836.492018
16	3.0	8	0.017423	0.057521	868.697313	36.834870	11.639429	9.440297e-07	0.315989	18456.506910
17	3.0	16	0.007103	0.012239	907.945911	40.691979	13.794150	1.226327e-06	0.338989	5792.144792
18	3.0	32	0.003776	0.007922	923.002133	40.496420	14.827261	1.343372e-07	0.366138	28110.085799
19	3.0	64	0.000364	0.001196	922.453194	36.458491	15.310863	2.697177e-08	0.419953	13493.127819
20	3.0	128	0.000399	0.002488	928.400703	40.867974	24.239399	9.585759e-09	0.593115	41572.108007
21	3.0	256	0.000441	0.004348	953.626123	37.398967	42.910742	8.114051e-09	1.147378	54354.099954
22	4.0	8	0.055971	0.035667	997.357052	38.184589	14.582781	4.620141e-04	0.381902	121.145146
23	4.0	16	0.001043	0.003092	1020.352923	39.787316	16.397327	1.272558e-07	0.412124	8194.839545
24	4.0	32	0.003568	0.012864	1041.137260	38.620322	16.940922	5.274864e-08	0.438653	67632.872655
25	4.0	64	0.000256	0.000469	1035.570870	37.830861	19.555648	1.431545e-08	0.516923	17865.229635
26	4.0	128	0.000141	0.000306	1031.302001	38.840197	29.290354	6.060972e-09	0.754125	23272.428147
27	5.0	8	0.007677	0.007903	1004.884643	35.967962	15.356913	4.751249e-05	0.426961	161.574054
28	5.0	16	0.006425	0.011390	1056.902538	39.110623	18.061716	1.714955e-07	0.461811	37464.942161
29	5.0	32	0.001815	0.007908	1083.987463	38.120286	19.141870	2.450275e-08	0.502144	74058.584878
30	5.0	64	0.000301	0.001010	1118.588767	37.981278	22.457180	1.653425e-08	0.591270	18215.317942
31	5.0	128	0.000194	0.000423	1132.072543	37.184725	33.509184	6.872760e-09	0.901155	28209.860938
32	6.0	8	0.004911	0.008642	1093.599316	37.996800	19.118926	1.021607e-06	0.503172	4807.263419
33	6.0	16	0.001648	0.002415	1049.262628	36.179746	19.490797	1.432225e-07	0.538721	11506.742450
34	6.0	32	0.000930	0.002172	1073.090730	36.949662	20.914077	2.585461e-08	0.566015	35951.890243
35	6.0	64	0.000191	0.000335	1078.962816	38.601445	26.605055	1.236363e-08	0.689224	15455.746774
36	7.0	8	0.029028	0.103579	1188.099535	38.484392	20.954034	9.738541e-07	0.544481	29807.196089
37	7.0	16	0.006760	0.014091	1342.888352	39.473755	23.800697	8.439203e-08	0.602950	80097.793589
38	7.0	32	0.000749	0.003084	1292.176548	38.856017	25.452476	2.731475e-08	0.655046	27416.354519
39	7.0	64	0.000285	0.000556	1329.204203	38.242596	29.407241	8.815478e-09	0.768966	32344.100555
40	1.5	7	0.029258	0.034046	686.425744	36.039533	5.709875	5.117514e-06	0.158434	5717.190951

5.1.3 Step size: 1×10^{-6}

The following are training results of networks intended to replace an integrator of step-size 1×10^{-6}:

Figure 5.12 and Figure 5.13 show mean squared error MSE and error deviation STD over 1000 samples. Again, their behavior follows a similar pattern to the networks trained for step size 1×10^{-5}, with similar decrease in MSE and STD with respect to step 1×10^{-4}. Validation losses in Figure 5.14 seem to decrease with respect to networks of step 1×10^{-4} with the same reductions as in Figure 5.9. The explanation for these phenomena given for those networks with step size 1×10^{-4} holds.

The error growth in Figure 5.15 seems to have increased a bit more however, when in comparison with Figure 5.10.

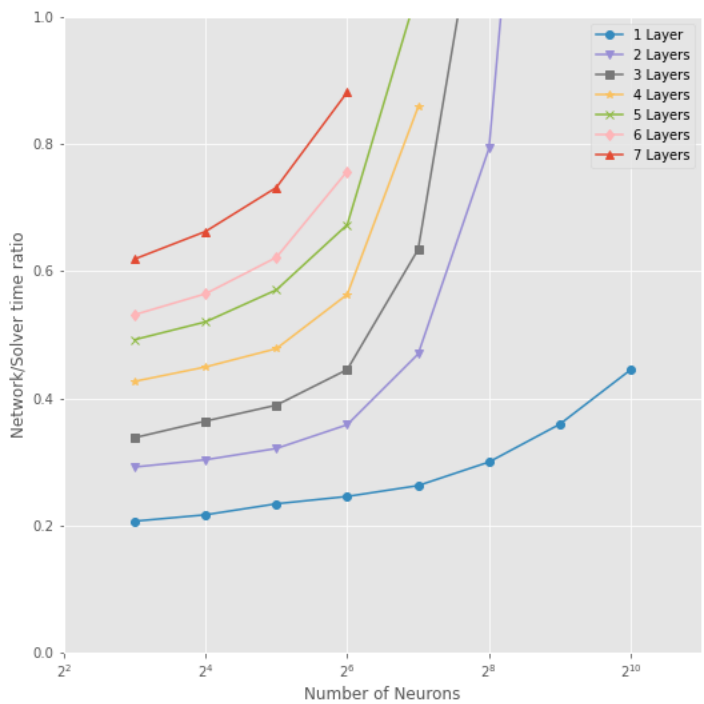


Figure 5.11: Speedup for step size 1×10^{-6}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

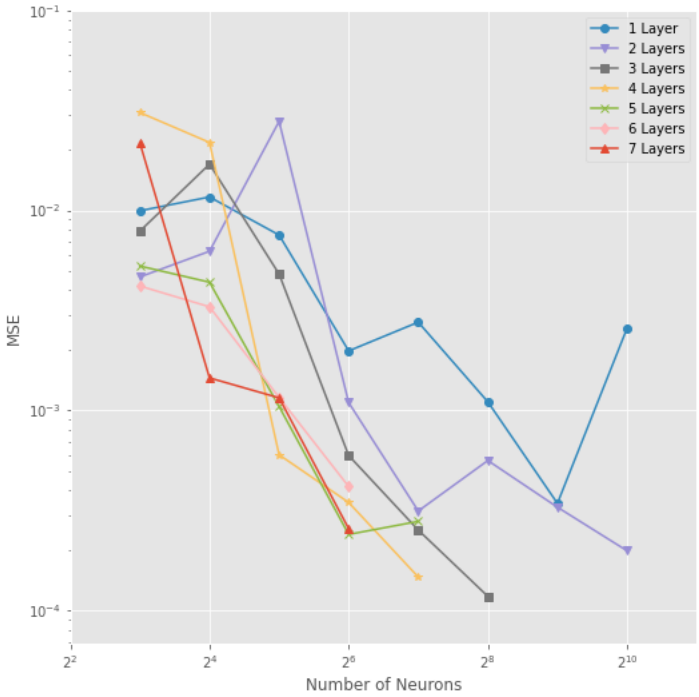


Figure 5.12: Mean Squared Error for step size 1×10^{-6}

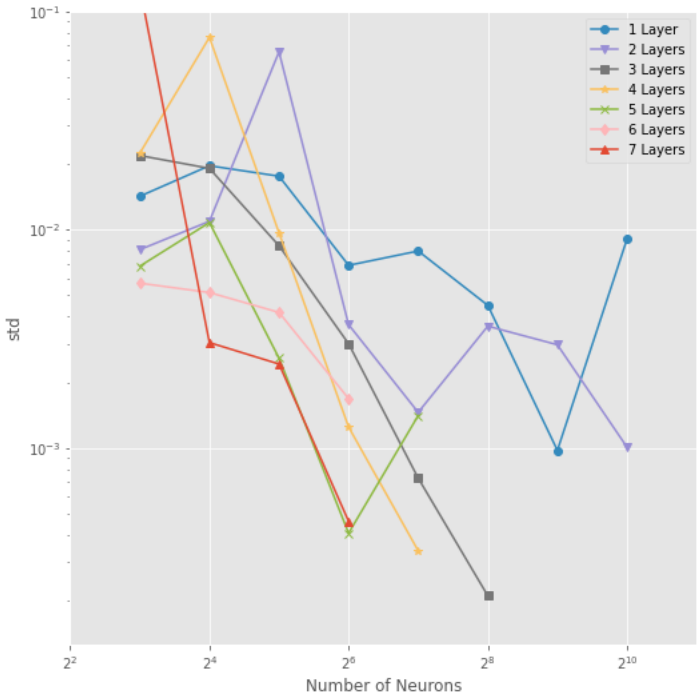


Figure 5.13: Error deviation for step size 1×10^{-6}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

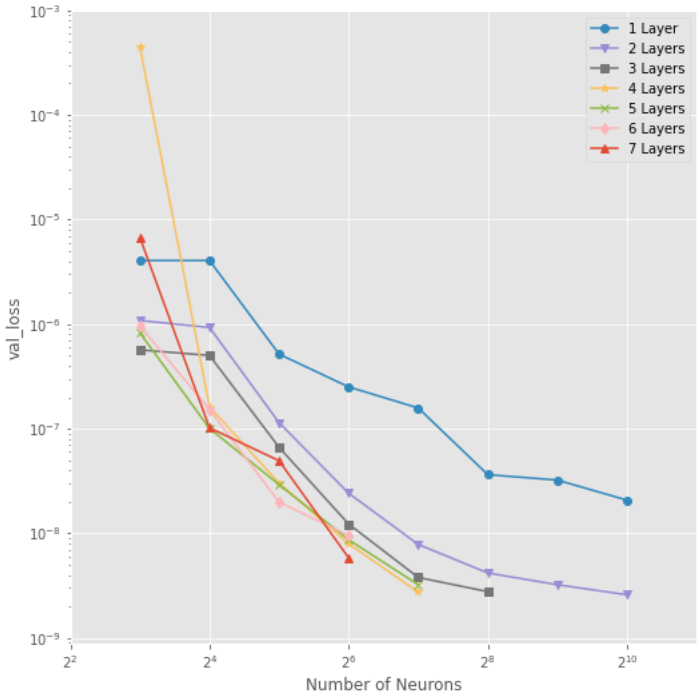


Figure 5.14: Validation Loss for step size 1×10^{-6}

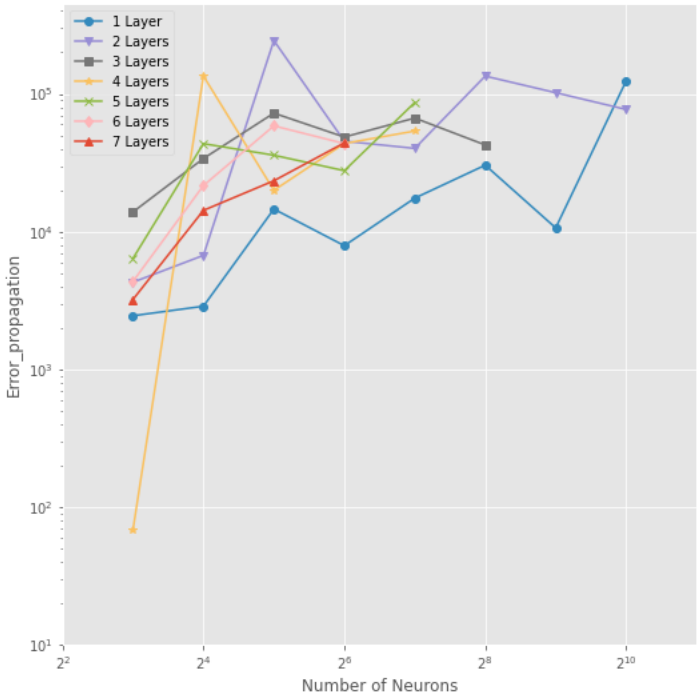


Figure 5.15: Error growth for step size 1×10^{-6}

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Table 5.4: Training results for step size 1×10^{-6}

	Layers	Neurons	MSE	STD	Elapsed Time (s)	Solver Time (s)	Network Time (s)	Validation loss	Time percentage	Error propagation
0	1.0	8	0.009957	0.014216	822.972898	42.128127	8.734096	4.056472e-06	0.207322	2454.614902
1	1.0	16	0.011675	0.019622	869.154863	38.741041	8.420566	4.056536e-06	0.217355	2878.136791
2	1.0	32	0.007535	0.017572	898.590212	41.909103	9.833926	5.163186e-07	0.234649	14593.702882
3	1.0	64	0.001986	0.006863	919.848721	39.439764	9.709080	2.509611e-07	0.246175	7915.431564
4	1.0	128	0.002765	0.007984	917.814224	39.207222	10.320991	1.578476e-07	0.263242	17515.881591
5	1.0	256	0.001105	0.004500	920.331415	40.470731	12.153204	3.637714e-08	0.300296	30389.850343
6	1.0	512	0.000344	0.000976	919.971878	42.104976	15.148365	3.228908e-08	0.359776	10652.318677
7	1.0	1024	0.002575	0.009048	929.490527	42.581913	18.964621	2.075454e-08	0.445368	124058.450641
8	1.5	7	0.987629	0.170429	797.053945	43.822737	7.636998	4.237914e-06	0.174270	233045.991445
9	2.0	8	0.004654	0.008096	1013.469139	43.551982	12.734756	1.082566e-06	0.292404	4298.841366
10	2.0	16	0.006252	0.010920	1004.410030	41.496525	12.607106	9.290321e-07	0.303811	6729.079405
11	2.0	32	0.027860	0.065183	1083.202228	44.069190	14.169793	1.142656e-07	0.321535	243817.632889
12	2.0	64	0.001097	0.003672	1075.976418	41.041997	14.730006	2.416883e-08	0.358901	45370.892917
13	2.0	128	0.000315	0.001456	1069.321846	41.953108	19.714694	7.828053e-09	0.469922	40231.487640
14	2.0	256	0.000563	0.003614	1091.910169	42.030105	33.330348	4.180666e-09	0.793011	134604.344655
15	2.0	512	0.000329	0.002970	1168.580735	42.207222	86.021226	3.225197e-09	2.038069	102006.183841
16	2.0	1024	0.000199	0.001009	1595.123912	41.724073	774.208409	2.586351e-09	18.555437	77067.174674
17	3.0	8	0.007884	0.021845	1001.157622	42.383364	14.346676	5.671173e-07	0.338498	13901.407160
18	3.0	16	0.017035	0.019083	1066.314767	41.488017	15.125783	5.016657e-07	0.364582	33956.340784
19	3.0	32	0.004811	0.008438	1066.850443	42.776007	16.666508	6.636036e-08	0.389623	72500.152338
20	3.0	64	0.000596	0.003002	1068.431833	41.976041	18.702330	1.225958e-08	0.445548	48625.153253
21	3.0	128	0.000253	0.000728	1075.168825	43.122412	27.347754	3.787248e-09	0.634189	66773.703355
22	3.0	256	0.000118	0.000211	1101.158669	43.210759	55.632455	2.769649e-09	1.287468	42592.696256
23	4.0	8	0.030833	0.022388	1162.969315	38.506174	16.443529	4.488772e-04	0.427036	68.689739
24	4.0	16	0.021866	0.075948	1245.610548	39.682076	17.843470	1.621887e-07	0.449661	134819.296396
25	4.0	32	0.000601	0.009670	1216.963900	40.998745	19.616314	3.012086e-08	0.478461	19964.528420
26	4.0	64	0.000347	0.001247	1226.753057	41.186149	23.212915	7.957319e-09	0.563610	43653.713923
27	4.0	128	0.000147	0.000338	1335.853916	43.004793	36.900799	2.740922e-09	0.858062	53772.795234
28	5.0	8	0.005273	0.006802	1169.946556	39.568910	19.488781	8.238481e-07	0.492528	6400.736382
29	5.0	16	0.004378	0.010778	1208.805355	42.123413	21.925069	1.005857e-07	0.520496	43528.024880
30	5.0	32	0.001048	0.002580	1249.742125	41.971150	23.955127	2.920356e-08	0.570752	35883.364116
31	5.0	64	0.000241	0.000407	1308.744916	41.473729	27.918161	8.680690e-09	0.673153	27708.376632
32	5.0	128	0.000279	0.001407	1329.575589	44.310164	46.225356	3.228559e-09	1.043222	86558.857635
33	6.0	8	0.004189	0.005687	1258.605460	38.975641	20.727274	9.617514e-07	0.531801	4356.052809
34	6.0	16	0.003299	0.005152	1278.384311	41.759673	23.580172	1.526706e-07	0.564664	21611.873616
35	6.0	32	0.001162	0.004178	1278.646999	42.359886	26.340665	1.982525e-08	0.621830	58615.664824
36	6.0	64	0.000417	0.001682	1283.071386	39.056370	29.572839	9.572887e-09	0.757183	43583.263716
37	7.0	8	0.021702	0.128605	1477.106700	39.940619	24.733632	6.734206e-06	0.619260	3222.600425
38	7.0	16	0.001456	0.003039	1522.671809	41.367185	27.407478	1.020004e-07	0.662542	14274.632000
39	7.0	32	0.001158	0.002424	1401.761922	40.474351	29.614590	4.943262e-08	0.731688	23419.088043
40	7.0	64	0.000257	0.000462	1441.225879	40.507993	35.707875	5.827824e-09	0.881502	44109.379510

5.2 Selected Architectures

The top performing neurons trained for the different time steps were later retrained with a batch size of 250 and doubling the number of epochs to 50. The best networks for step sizes 1×10^{-4} and 1×10^{-5} are explored here, as these are the ones that will be coupled to the combustion simulation.

As mentioned in Subsection 4.4.2, some fixed initial conditions (Table 4.7) were also solved for to allow for a visual inspection.

5.2.1 Network for step 1×10^{-4}

This network posseses the following parameters:

Table 5.5: Parameters of Neural Network for Step 1×10^{-4}

Parameter	Value
Layers	4
Neurons	64
MSE	0.00006808
STD	0.00014498
Elapsed Time (s)	3191.4328
Solver Time (s)	37.9383
Network Time (s)	16.7775
Speedup	0.4422

Figure 5.16 and Figure 5.17 show the time evolution of initial condition Y_1 (see Table 4.7) obtained by the VODE solver and the network trained for step size 1×10^{-4}. Two graphs are shown as this initial condition evolves very fast due to the high initial temperature. Figure 5.16 shows both solvers produce almost the same values, but this could be explained by the network only having to predict over one time step.

The extended plots shown in Figure 5.17 show a good approximation by the neural network, however the steady state values for CO_2 and CO are slightly moved. The

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

gradients for theses same fractions seem to invert their sign, which is an undesired behavior.

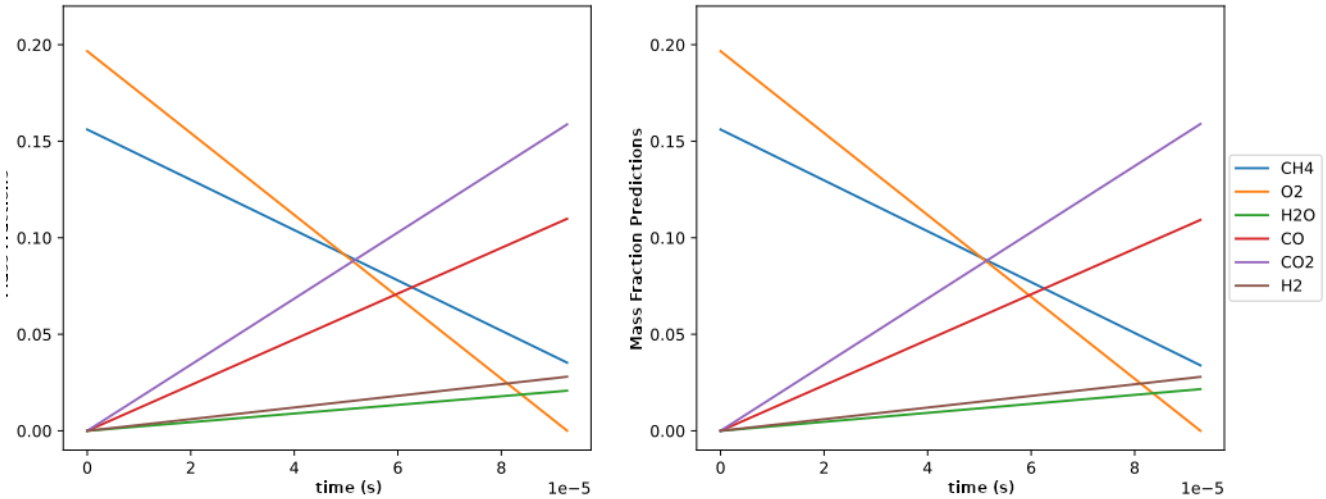


Figure 5.16: Initial condition I evaluation

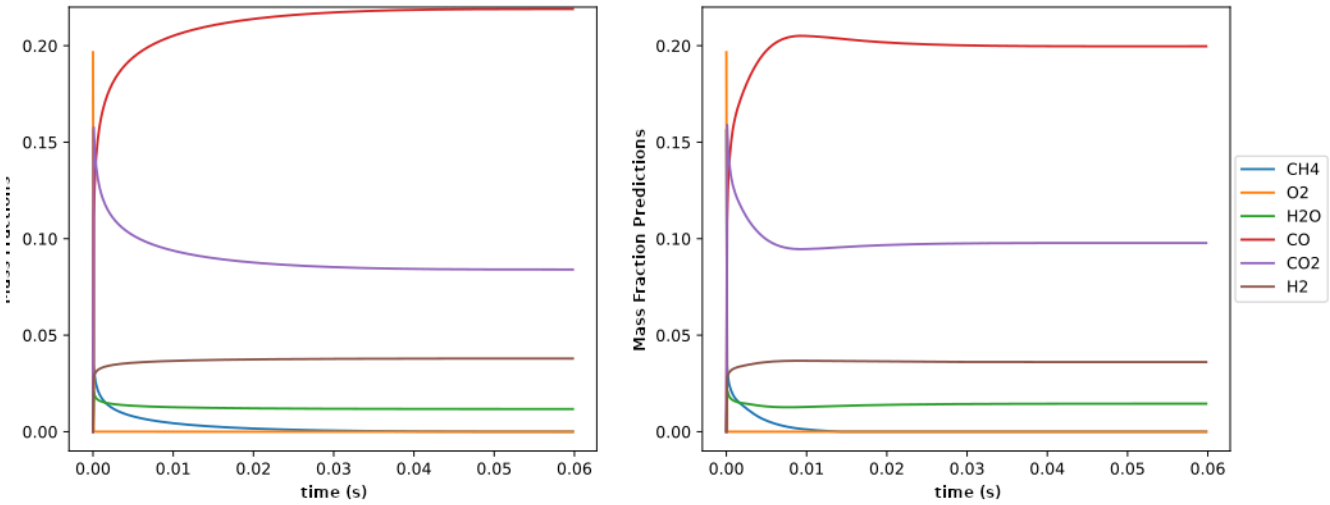


Figure 5.17: Initial condition I evaluation (Extended)

Figure 5.18 shows the evolution of condition Y_3. Here both mass fractions and their respective gradients seem to be correct. The initial straight horizontal lines are caused by a slow reaction due to a low starting temperature. The evolution of initial condition Y_2, is not shown as it reaches steady state in a time frame smaller than the allowed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

time step for this network.

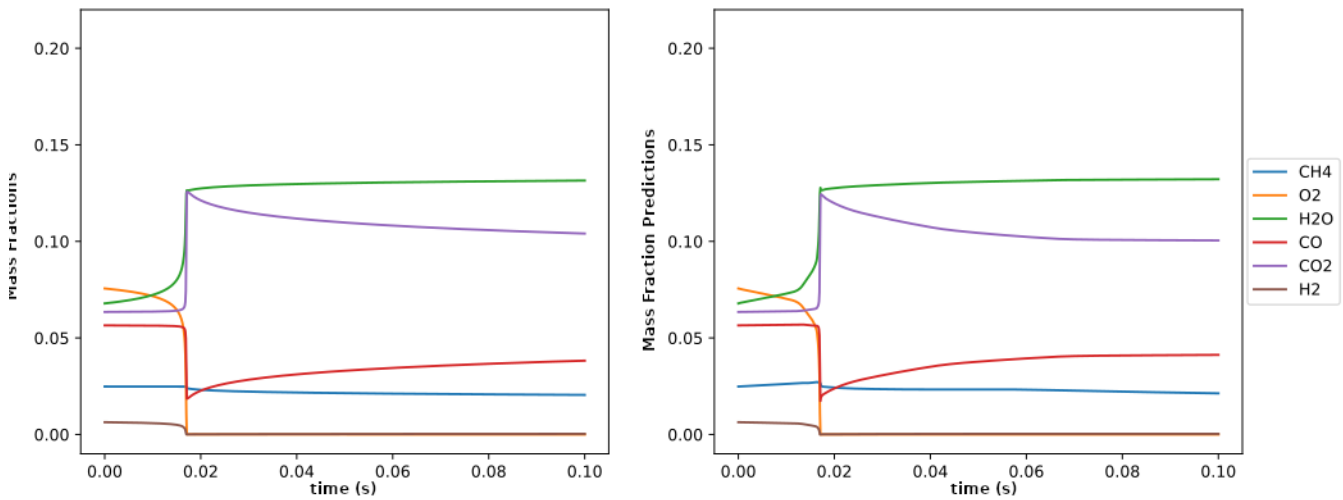


Figure 5.18: Initial condition III evaluation

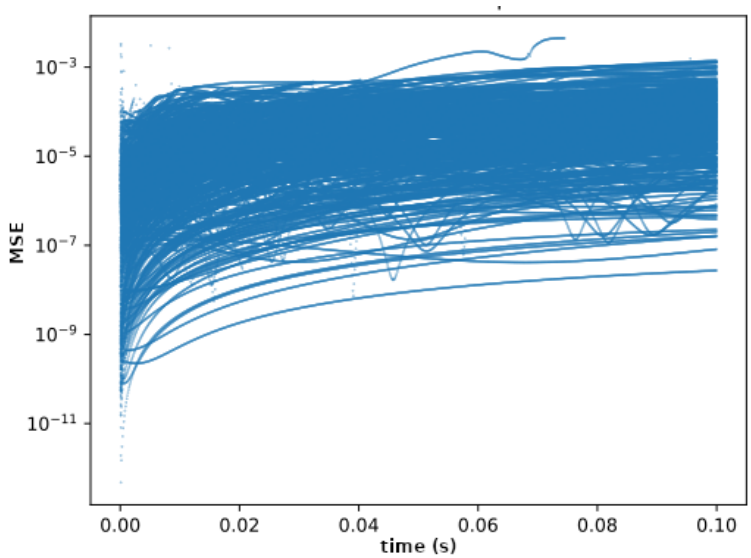


Figure 5.19: Error propagation for 1000 samples

In Figure 5.19 the MSE error across a thousand evaluated samples appears to have an upper bound of the order of 10^{-3}.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

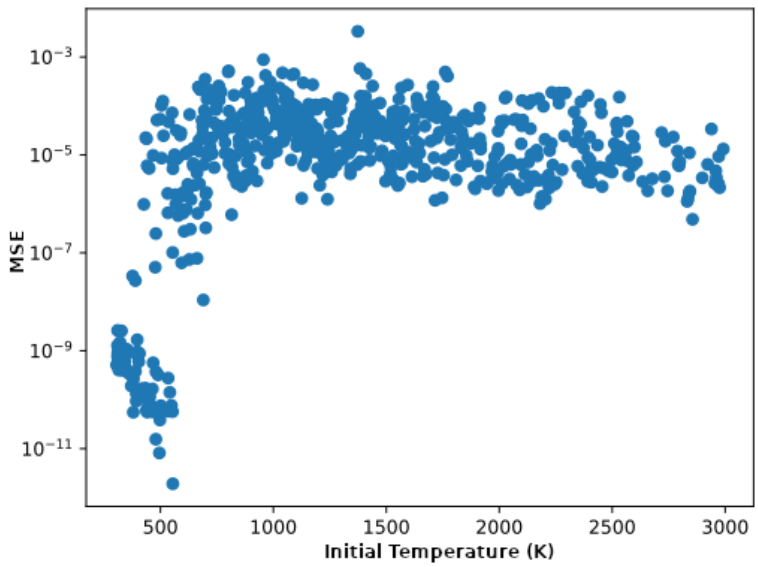


Figure 5.20: MSE vs Initial Temperature for 1000 samples

Figure 5.20 shows the error accross a developed initial condition compared to its initial temperature. As can be seen the error for initial temperatures in the range between 1000 and 3000 K is quite uniform, with smaller errors beign shown for those temperatures with less than 1000 K. The reason for why this might be ocurring is that species subjected to smaller temperatures show less variability than those for higher ones, and thus its behavior is easier to capture by the ANN

5.2.2 Network for step 1×10^{-5}

This network posseses the following parameters:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Table 5.6: Parameters of Neural Network for Step 1×10^{-5}

Parameter	Value
Layers	6
Neurons	64
MSE	0.00007705
STD	0.00035005
Elapsed Time (s)	4375.4149
Solver Time (s)	38.8137
Network Time (s)	26.3046
Speedup	0.6777

Figure 5.21 and Figure 5.22 show the time evolution of initial condition Y_1 (see Table 4.7) obtained by the VODE solver and the network trained for step size 1×10^{-5}. This initial condition evolves very fast due to the high initial temperature. Figure 5.21 shows both solver produce almost the same values and the correct gradients in the extended view of a small time interval.

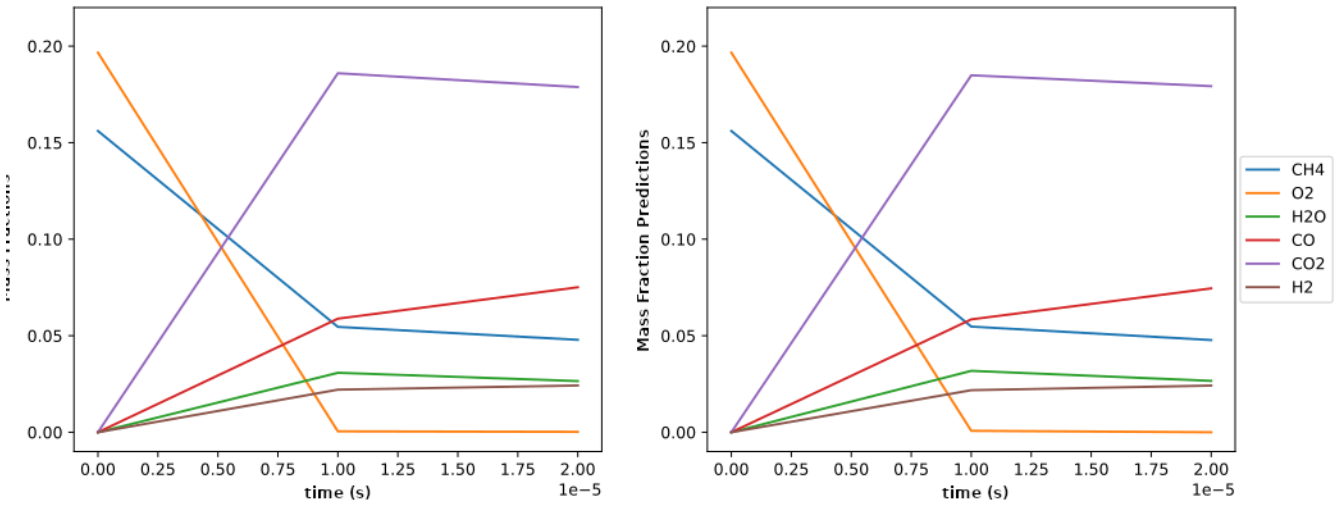


Figure 5.21: Initial condition I evaluation

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

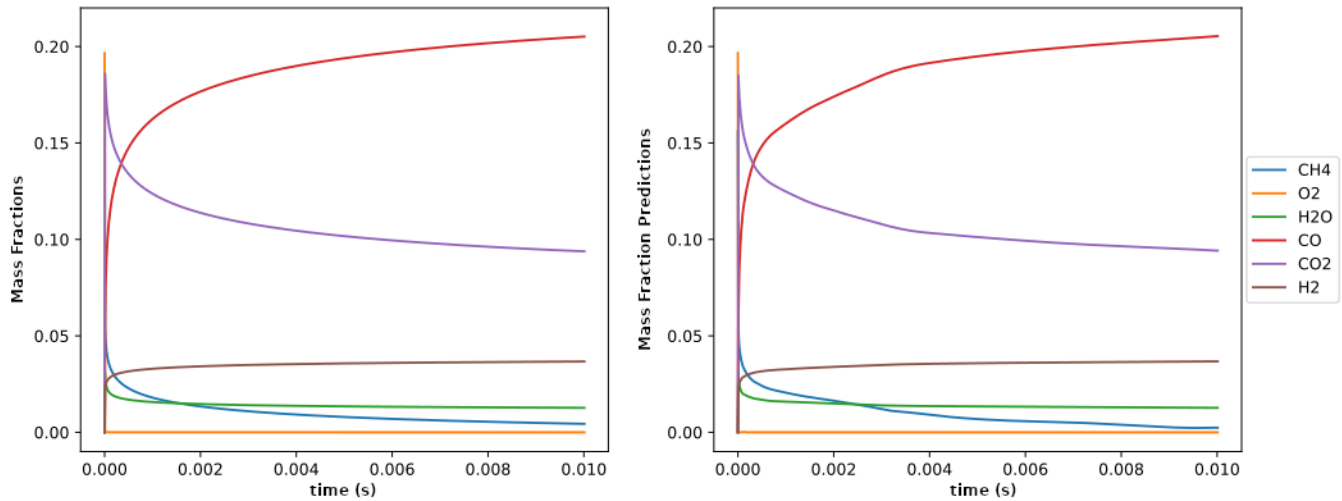


Figure 5.22: Initial condition I evaluation (Extended)

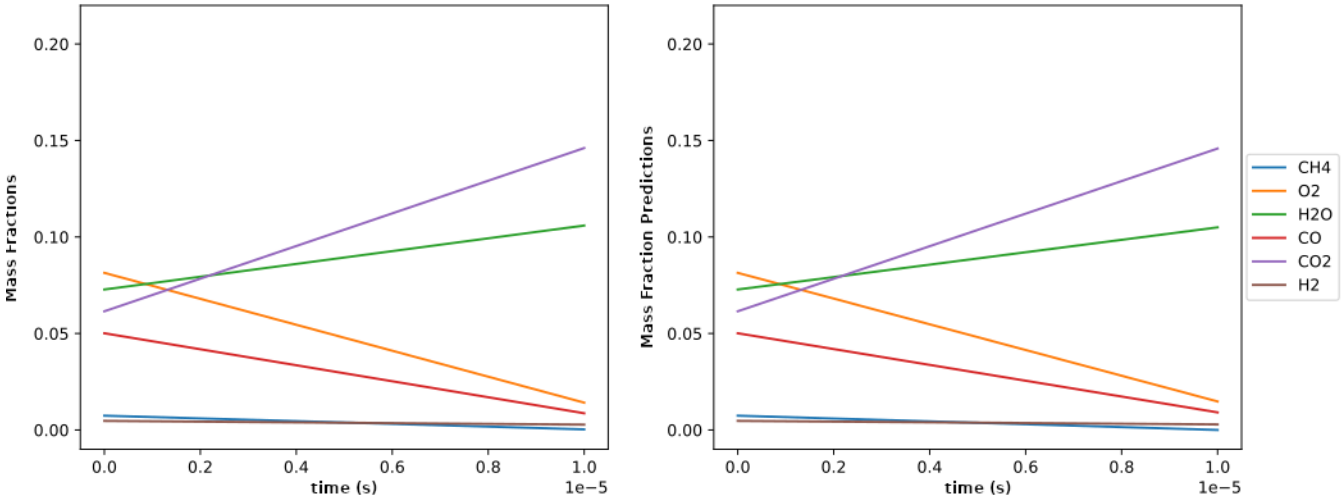


Figure 5.23: Initial condition II evaluation

The extended plots shown in Figure 5.22 show a good approximation by the neural network as well as a correct behavior for the gradients.

Figure 5.23 shows the accuracy in the approximation of Y_2 over a single time step. Figure 5.24 is also an extremely good approximation, although the O_2 fraction seems to decrease at a slightly faster rate.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

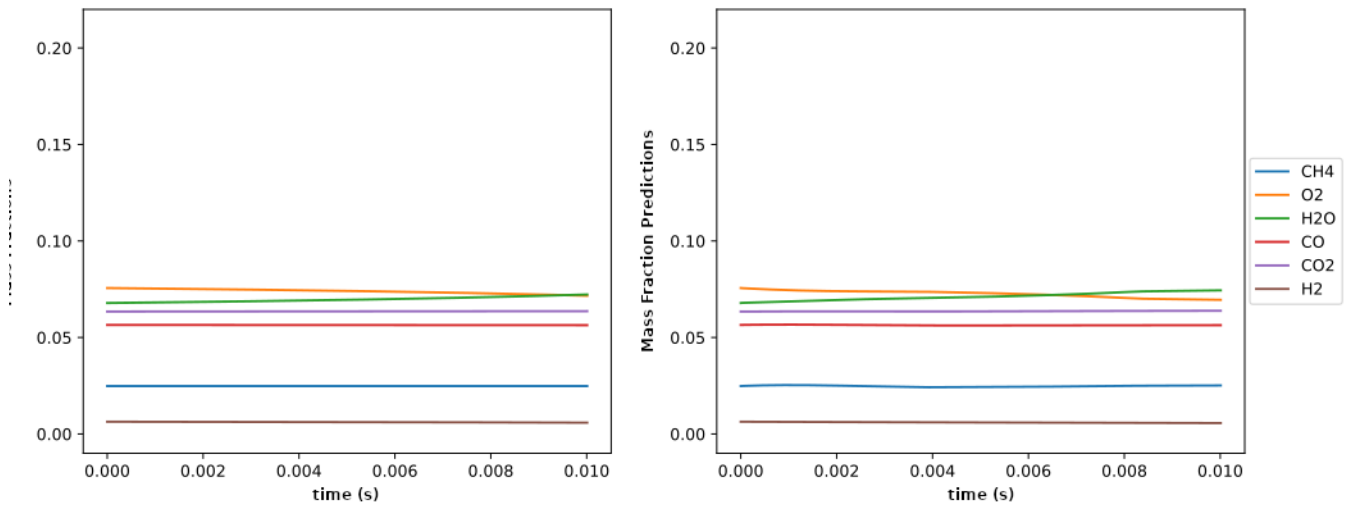


Figure 5.24: Initial condition III evaluation

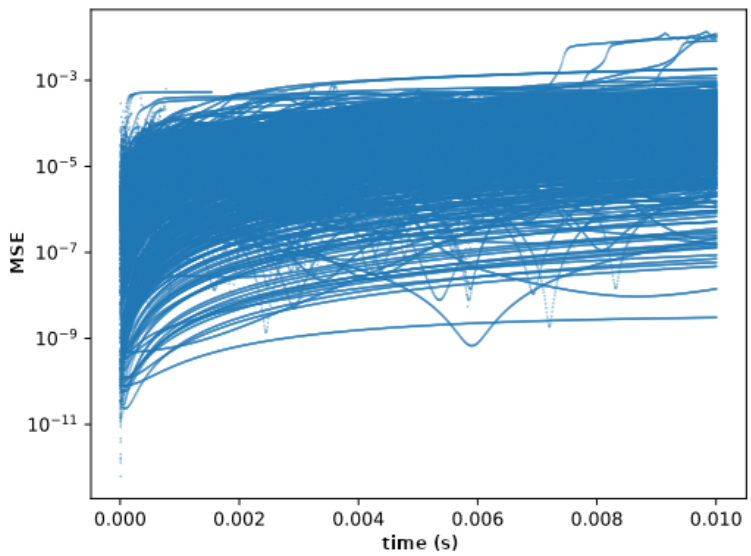


Figure 5.25: Error propagation for 1000 samples

Figure 5.25 shows also that the accumulated MSE error across a thousand evaluated samples seems to have an upper bound of the order of 10^{-3}.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

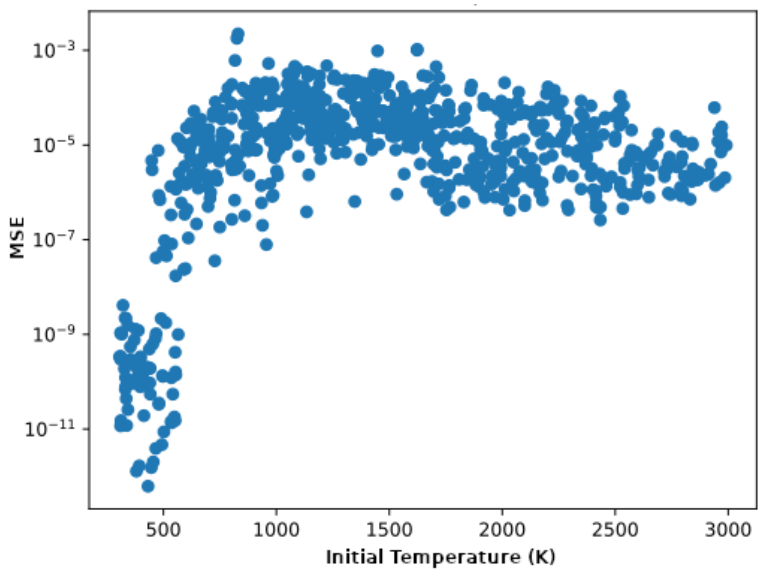


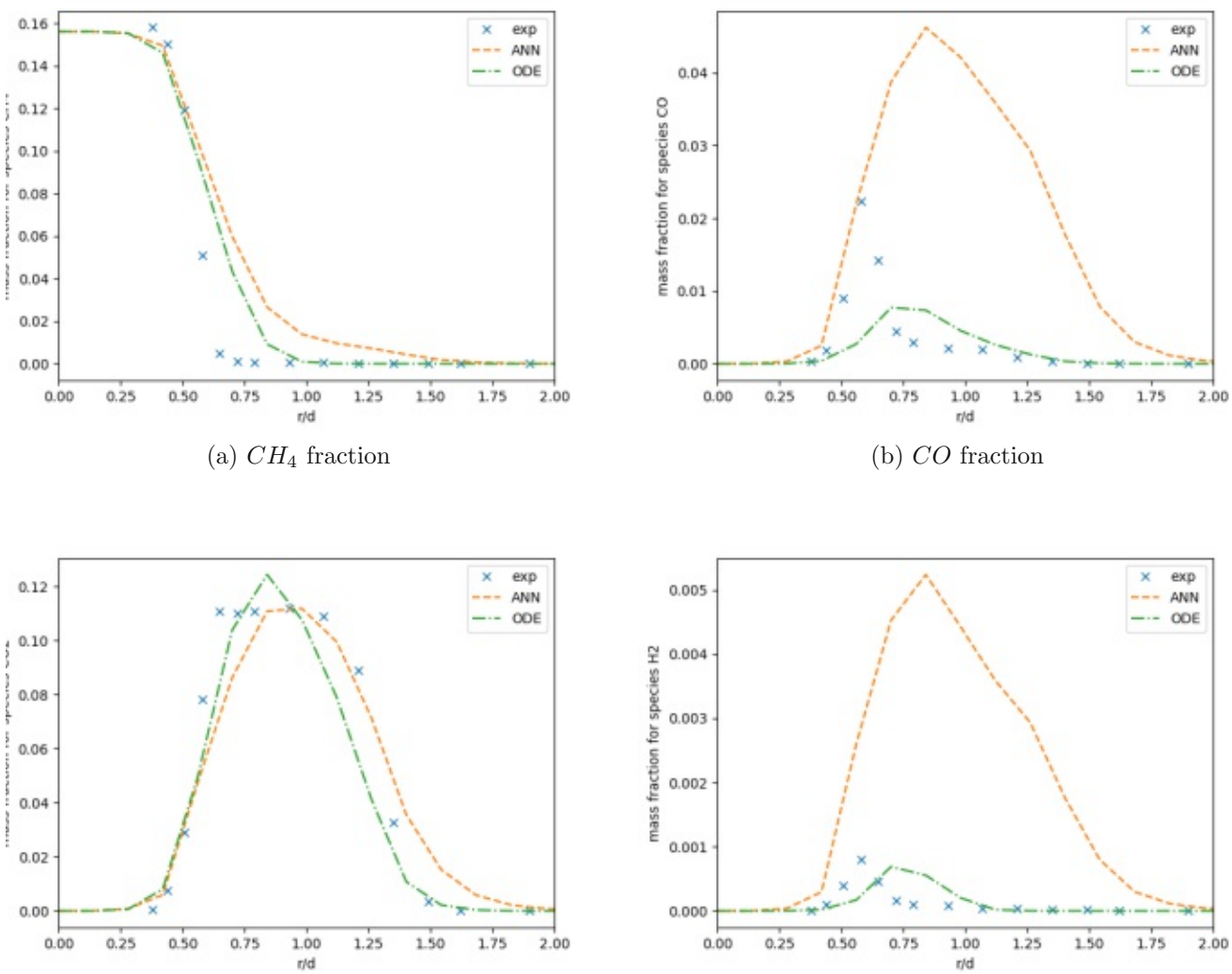
Figure 5.26: MSE vs Initial Temperature for 1000 samples

Figure 5.26 shows the error accross a developed initial condition compared to its initial temperature. As can be seen the error for initial temperatures with less than 1000 K seems to be smaller, as was the case for Figure 5.20. The same reasons for this behavior apply.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

5.3 Simulation Results

5.3.1 Radial profiles at x/d=1



(a) CH_4 fraction

(b) CO fraction

(c) CO_2 fraction

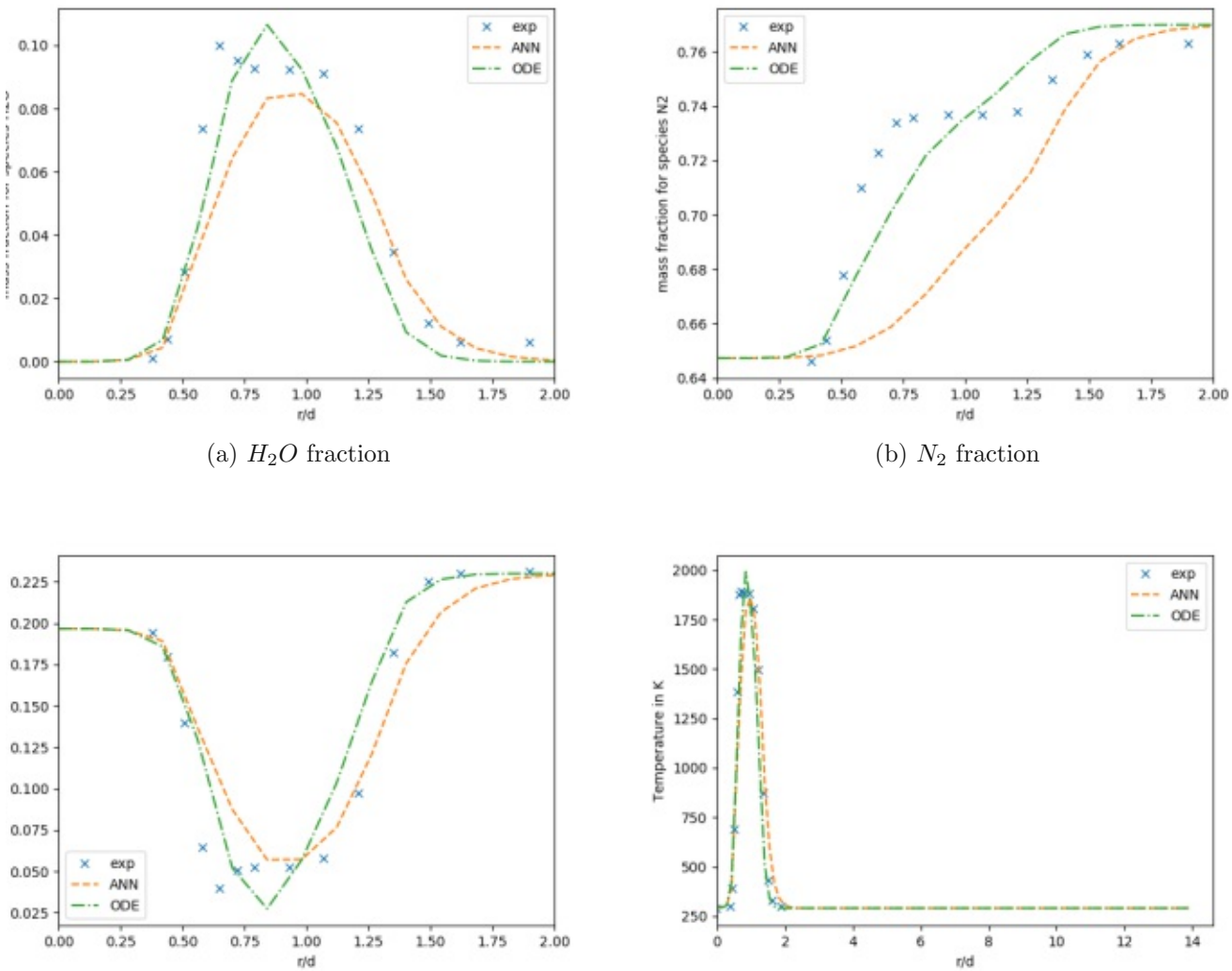
(d) H_2 fraction

Figure 5.27: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 1$

Figure 5.27 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 1$. The predictions for CH_4 and CO_2 show quite a degree of accuracy, with methane fractions being less accurate when r/d is in the range betwen around 0.60 and

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

1. CO and H_2 predictions seem somewhat accurate until $r/d = 0.60$ after which they diverge completely.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

(d) Temperature

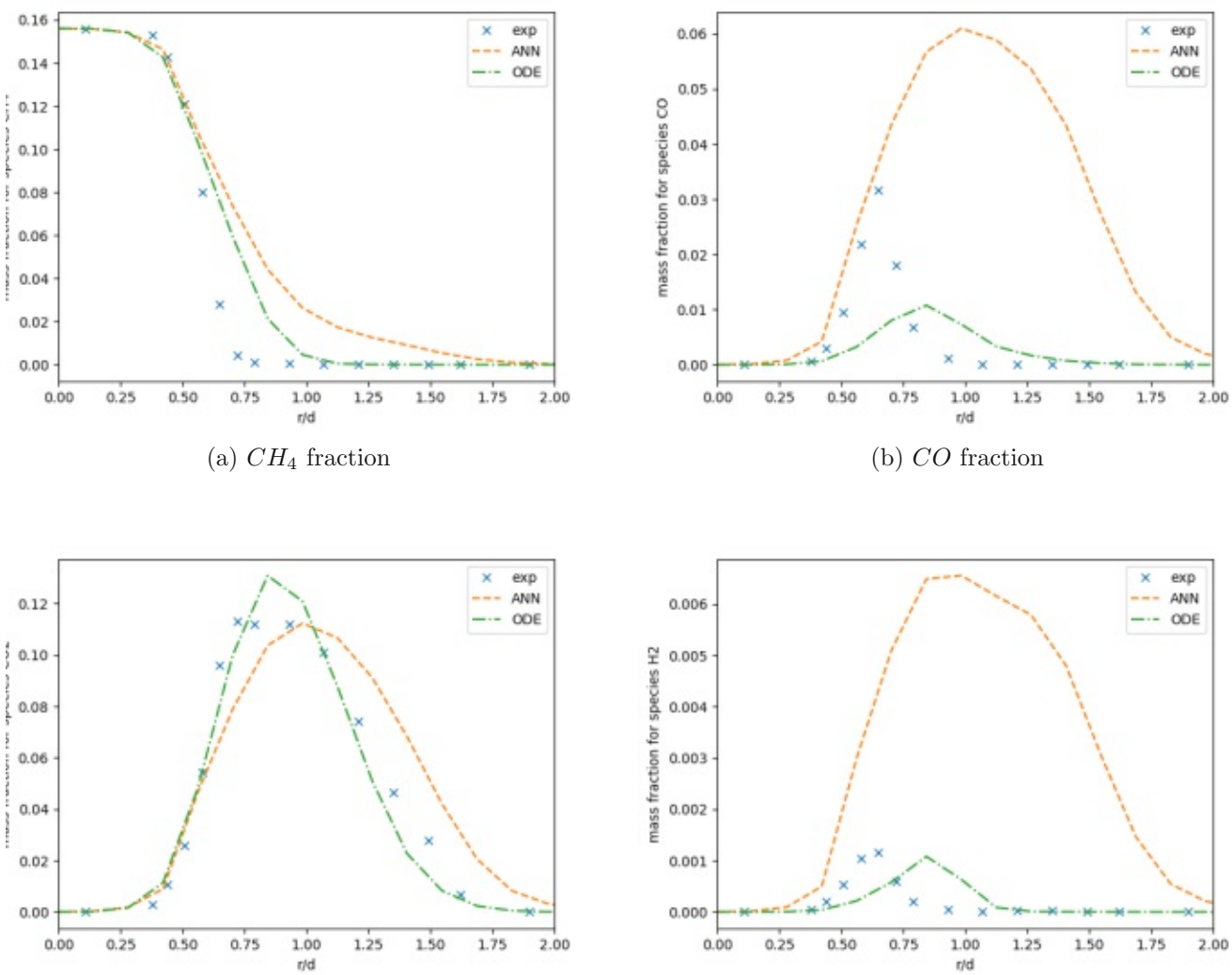
Figure 5.28: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 1$

Figure 5.28 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 1$. The predictions for H_2O, O_2 and temperature show good results, however N_2 fractions apparently evolve a slower rate when r/d is in the range betwen around 0.50 and 1.25. From the temperature profile, it is possible to infer that pred-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

diction accuracy seem to decrease when temperature is higher than 1250K.

5.3.2 Radial profiles at x/d=2



(a) CH_4 fraction

(b) CO fraction

(c) CO_2 fraction

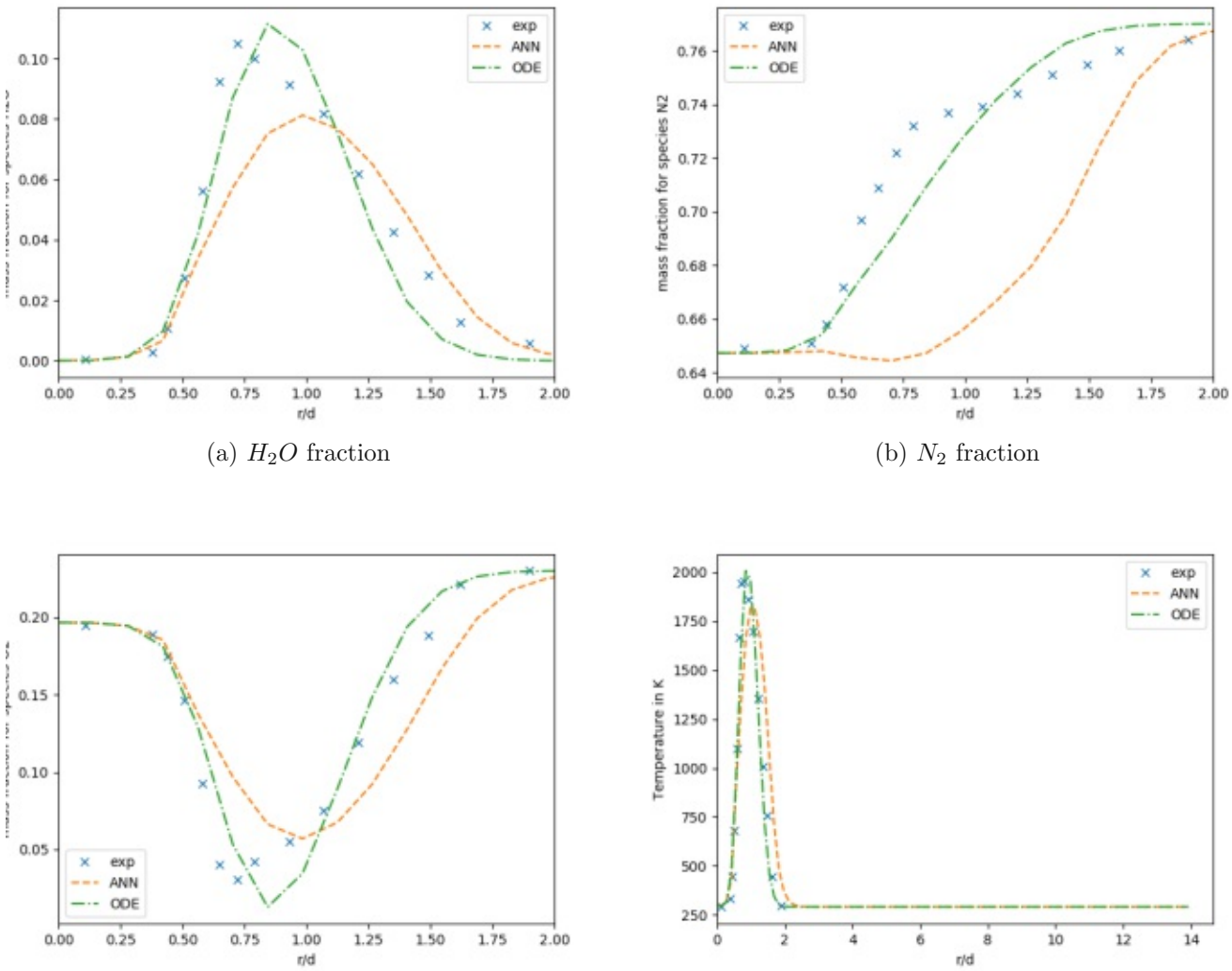
(d) H_2 fraction

Figure 5.29: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 2$

Figure 5.29 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 2$. The predictions for CH_4 and CO_2 show good degree of accuracy, with

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

methane fractions being less accurate when r/d is in the range betwen around 0.60 and 1. CO and H_2 predictions seem somewhat accurate until $x/d = 0.60$ after which they diverge completely. The overall behavior is similar to that shown when $x/d = 1$.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

(d) Temperature

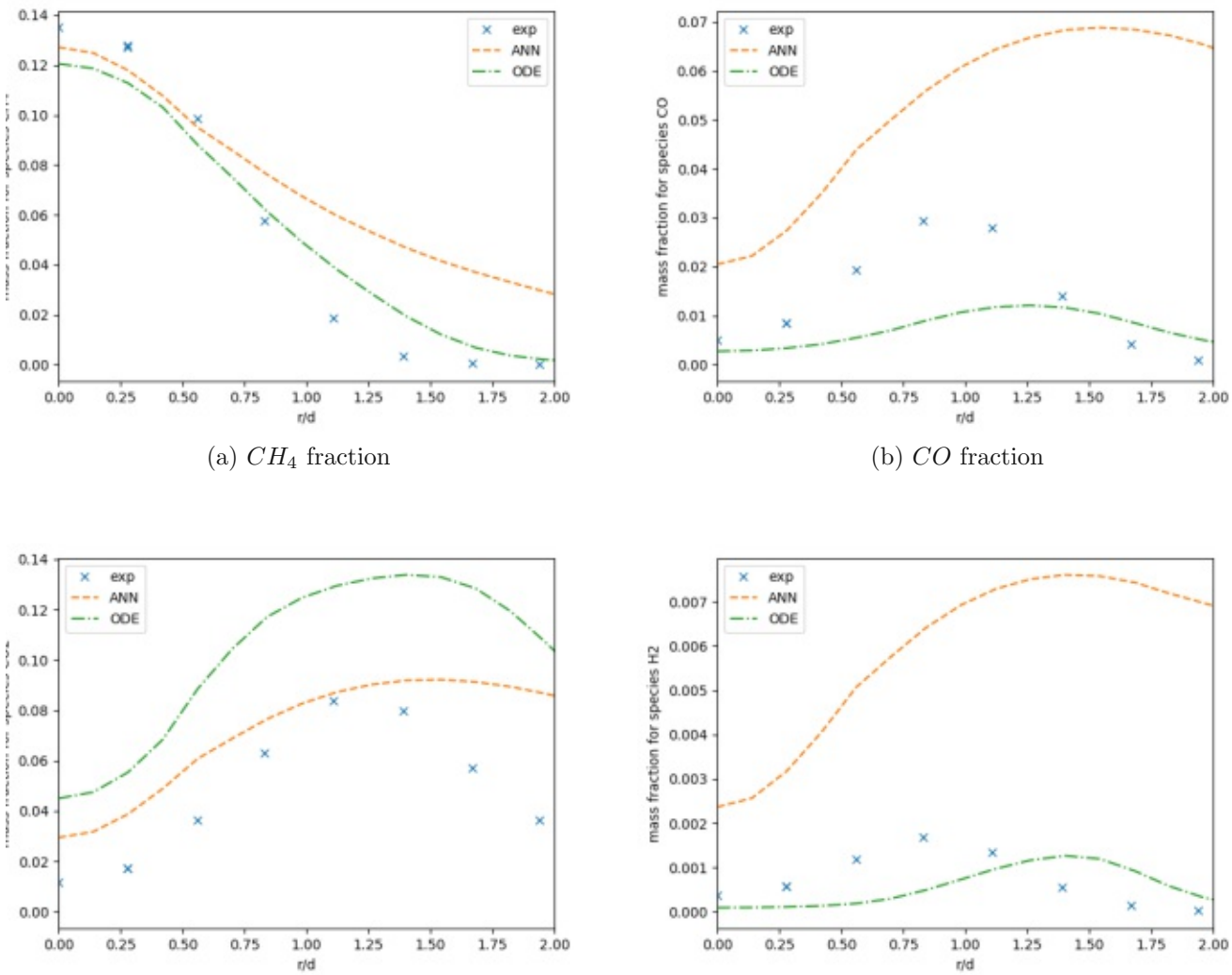
Figure 5.30: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 2$

Figure 5.30 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature at $x/d = 2$. The predictions for H_2O, O_2 and temperature show good results as in the case for $x/d = 1$. N_2 fractions apparently evolve a at an even slower rate when

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

r/d is in the range betwen around 0.50 and 1.25 than the case when $x/d = 1$. From the temperature profile, it is possible to infer that prediction accuracy seems to decrease when temperature is higher than around 1100K. There appears to be an slight decrease in the fraction of N_2 which should not be possible.

5.3.3 Radial profiles at x/d=15



(a) CH_4 fraction

(b) CO fraction

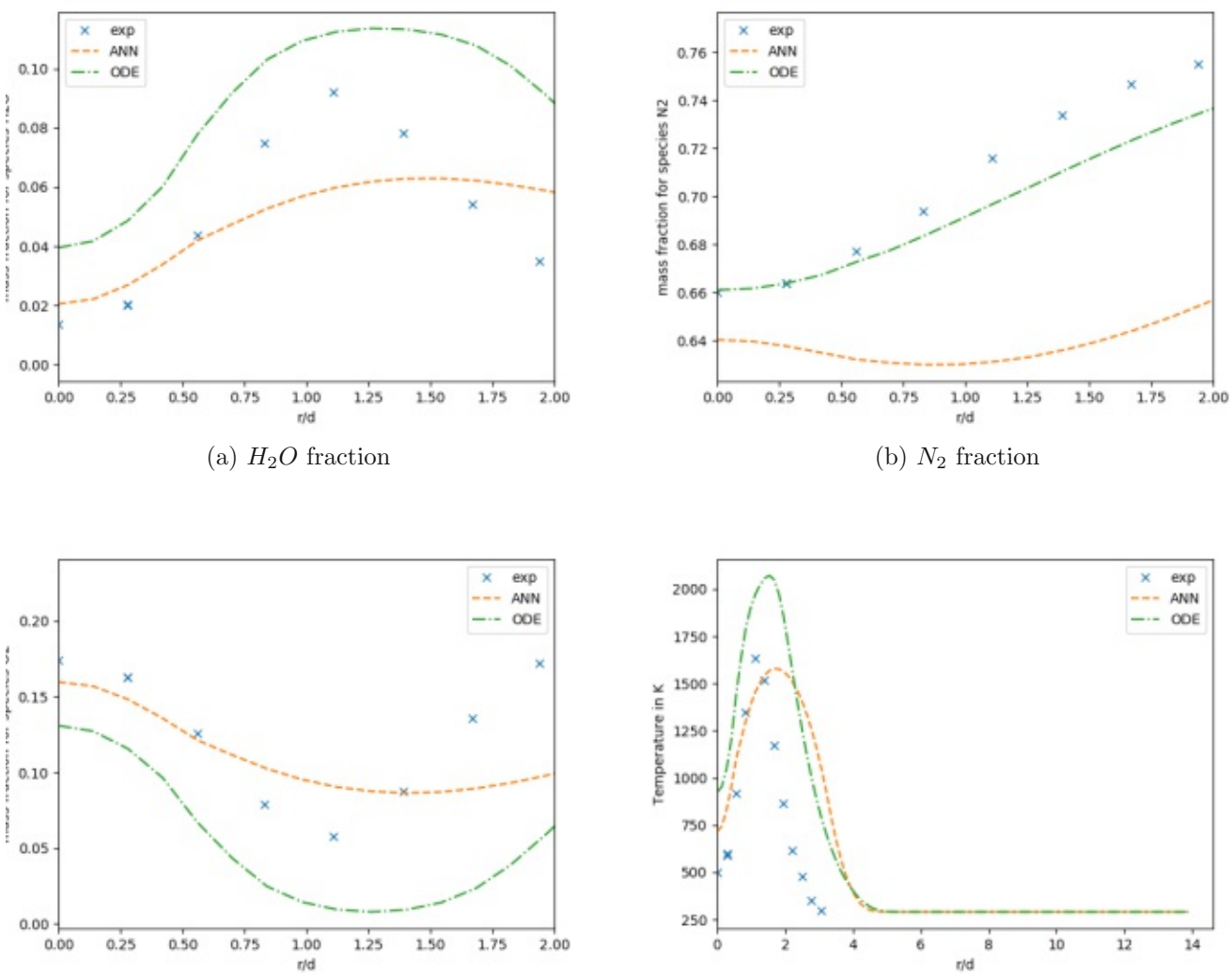
(c) CO_2 fraction

(d) H_2 fraction

Figure 5.31: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 15$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figure 5.31 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 15$. The predictions for CH_4 and CO_2 are still good, though a bit further than the experimetal data, with CO_2 predictions being actually closer than the ones calculated with the ODE integratoe. CO and H_2 innacuracies after $r/d = 0.60$ have increased more. They approximate the right shape in this range, but offset vertically from the correct data.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

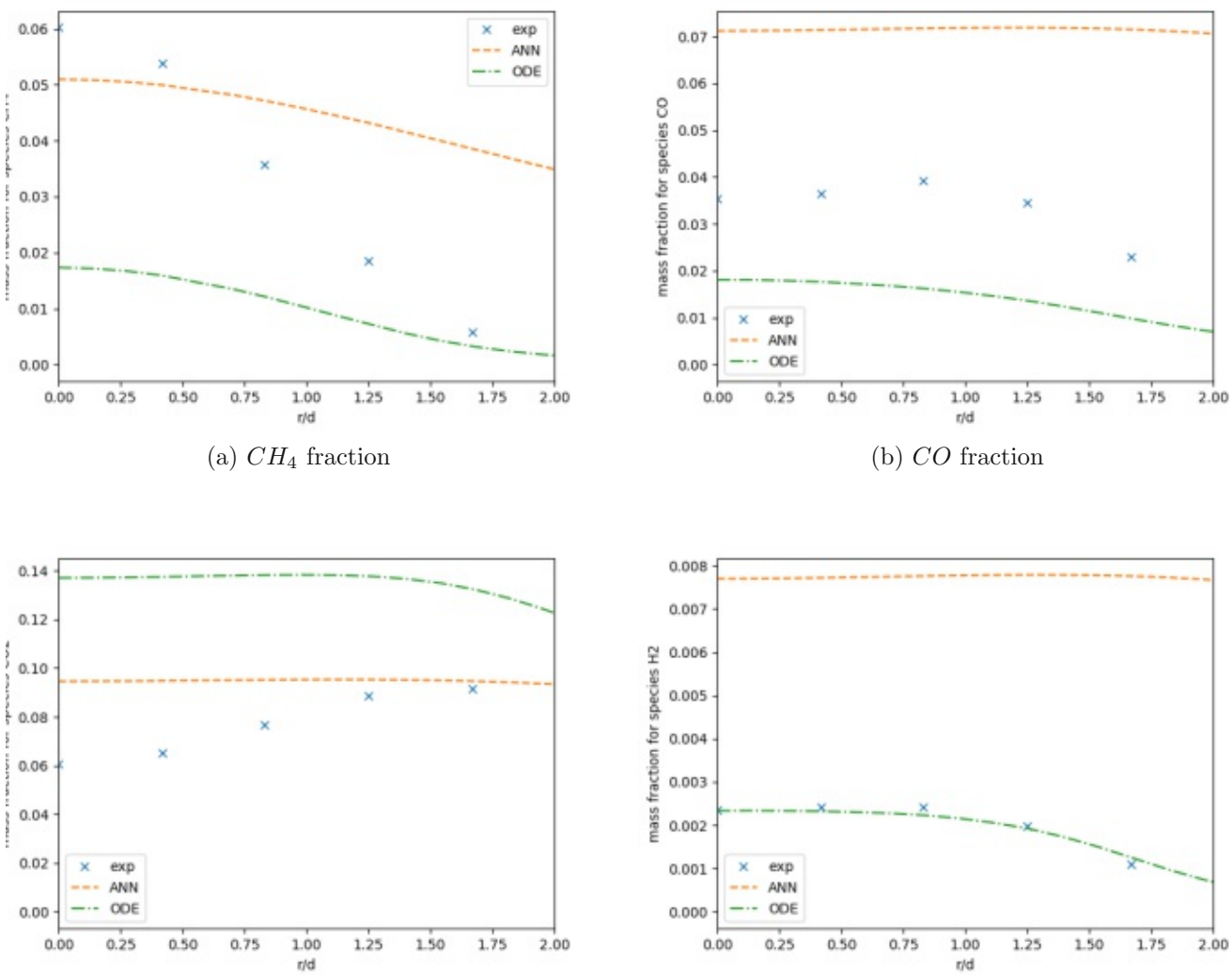
(d) Temperature

Figure 5.32: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 15$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figure 5.32 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 15$. The predictions for H_2O, O_2 seem to be accurate and closer to the data than the ones computed through the ODE solver. N_2 error is still increasing, and temperature values have grown more distant from the experimental data.

5.3.4 Radial profiles at x/d=30



(a) CH_4 fraction

(b) CO fraction

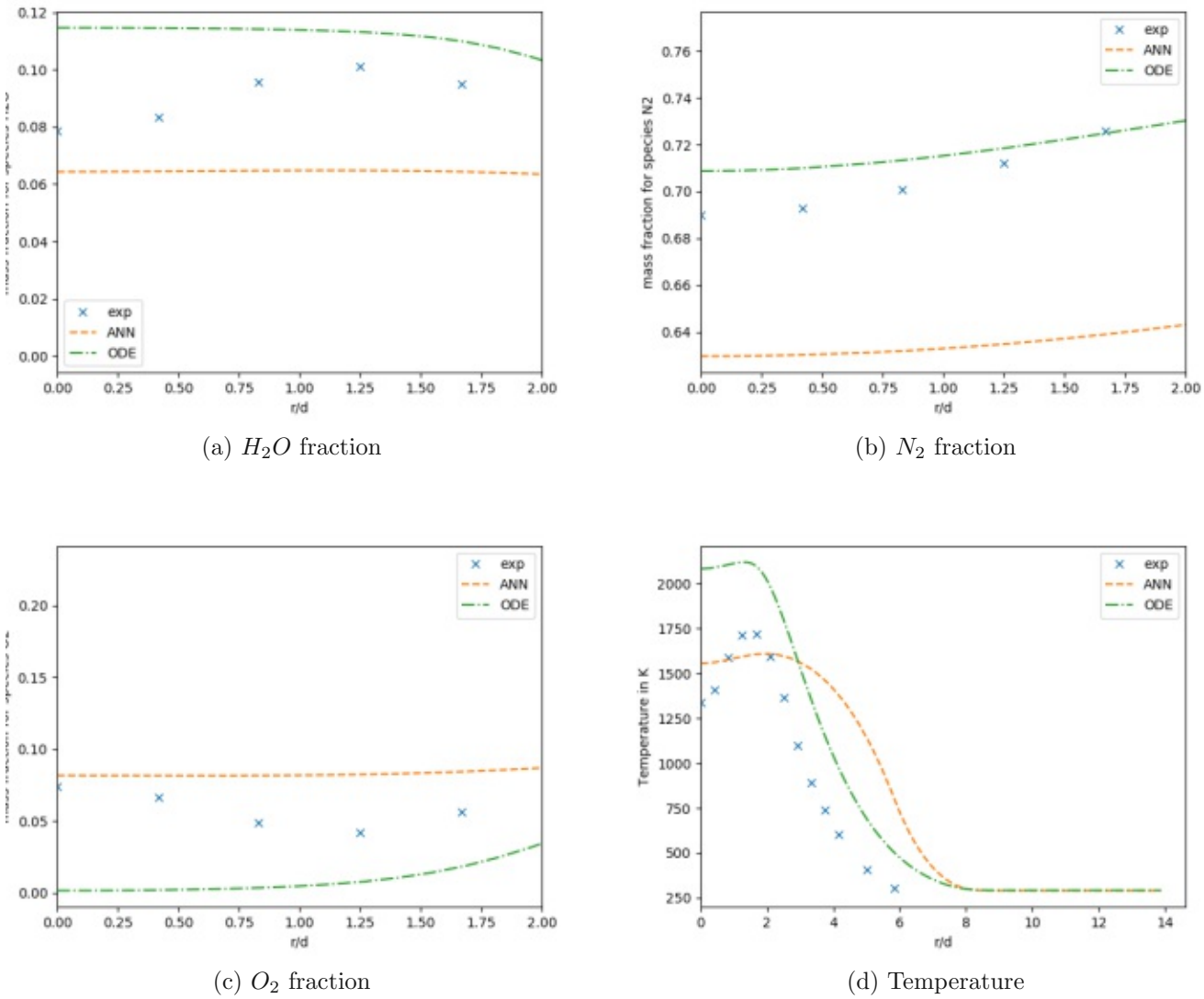
(c) CO_2 fraction

(d) H_2 fraction

Figure 5.33: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 30$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figure 5.33 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 30$. The predictions for CH_4 and CO_2 have increased in error, but seem to be closer to the experimental data than the ones from ODE integrator. CO and H_2 have stopped approximating the experimental data, but curiously seem to follow the shape of the ODE integrator curves.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

(d) Temperature

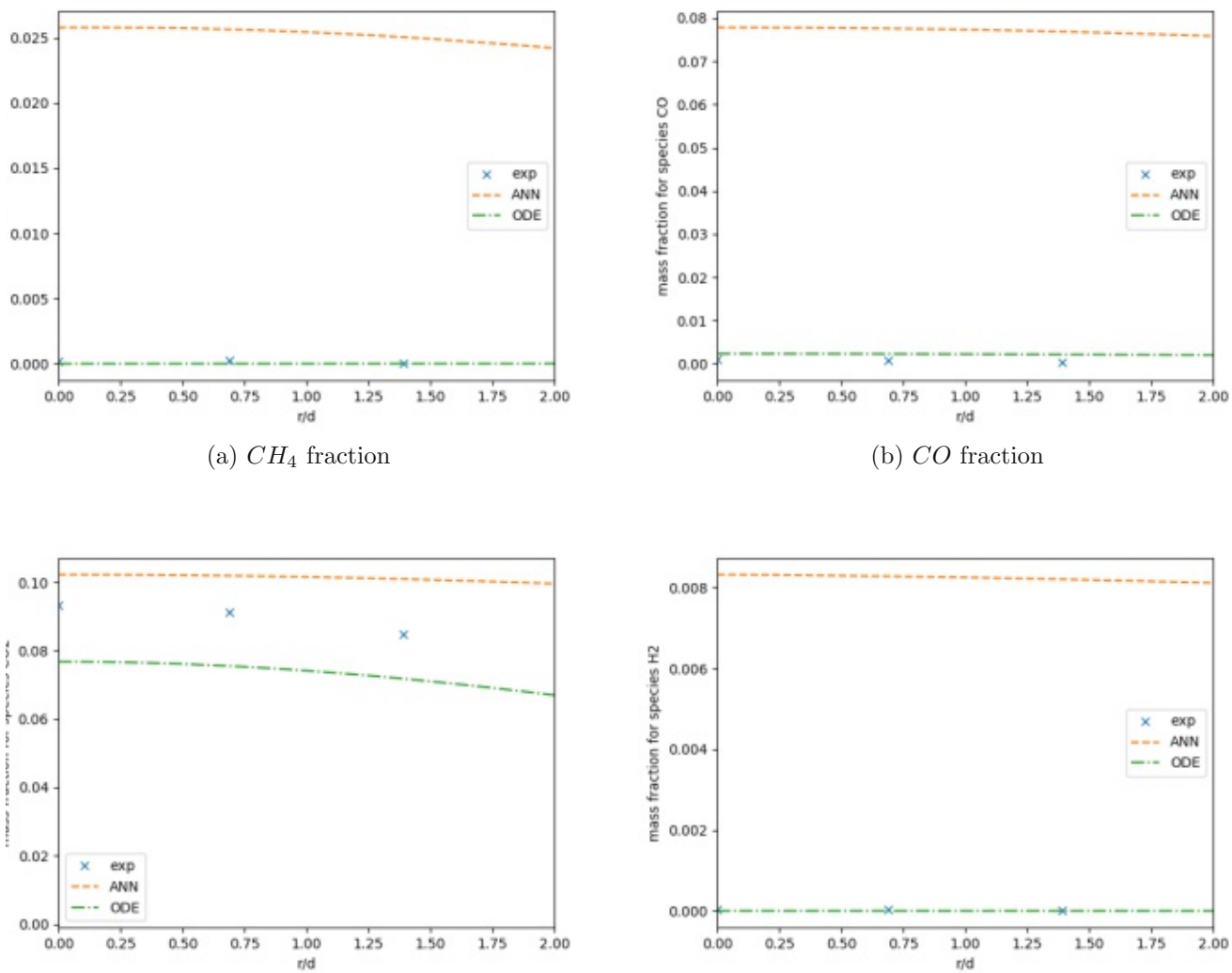
Figure 5.34: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 30$

Figure 5.34 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temper-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

ature with $x/d = 30$. The predictions for H_2O, O_2 seem to be accurate and closer to the data than the ones computed through the ODE solver. N_2 error is still increasing, and temperature values have grown more distant from the experimental data.

5.3.5 Radial profiles at x/d=60



(a) CH_4 fraction

(b) CO fraction

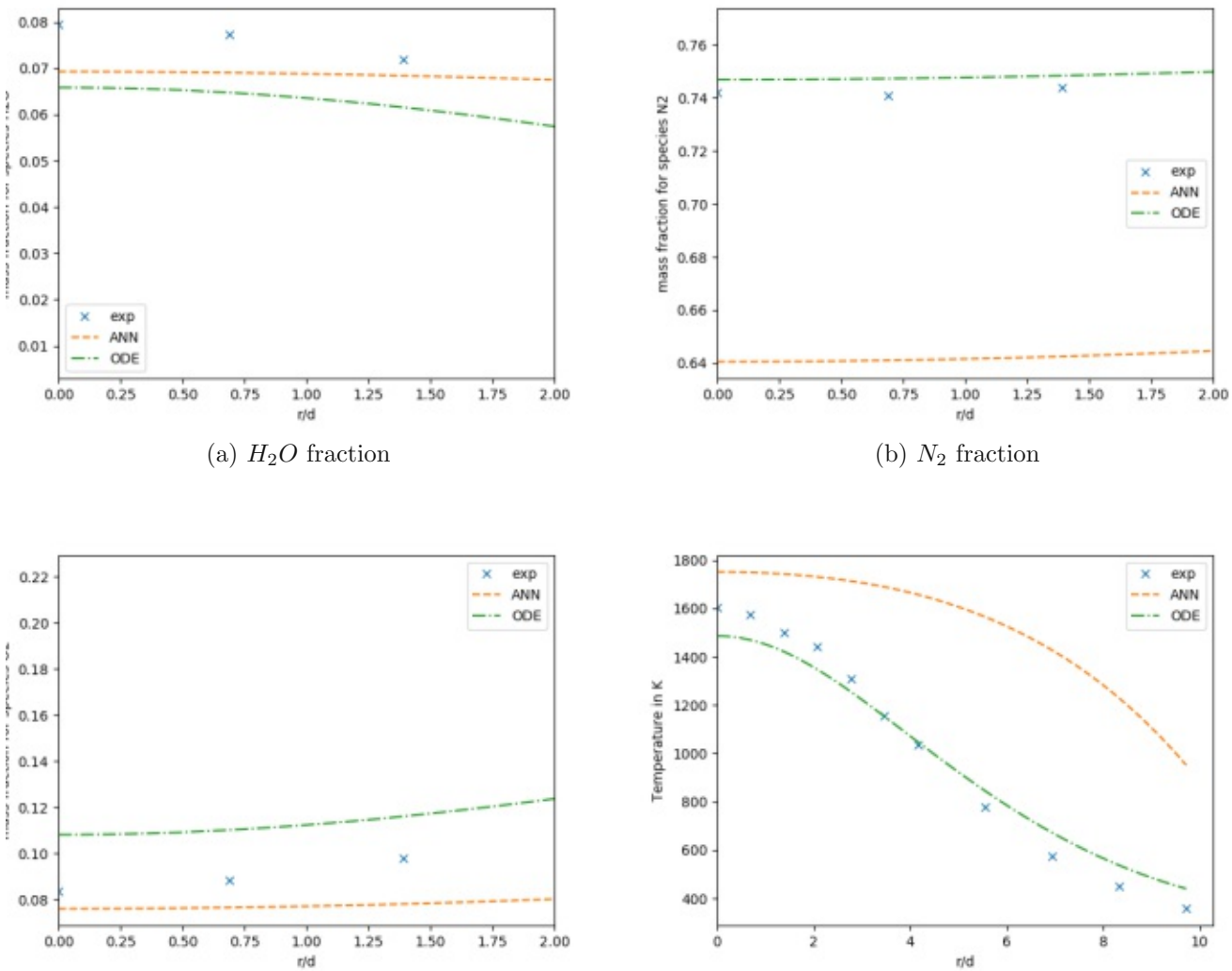
(c) CO_2 fraction

(d) H_2 fraction

Figure 5.35: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 60$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figure 5.35 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 60$. Only the pedictions for CO_2 are show some degree of accuracy.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

(d) Temperature

Figure 5.36: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 60$

Figure 5.36 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 60$. Water and oxygen predictions seem to be somewhat close. Temperature results have diverged notably.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

6 Conclusions and Outlook

6.1 Conclusions

- Over a thousand combinations of ANN architectures, training data, activation functions, and hyperparameters were tested to generate an accurate model capable of replacing the numerical integrator used when simulating reacting flows via the EDC. The results of the 40 best performing networks trained to work with an integration time step of 1×10^{-4},1×10^{-5}, and 1×10^{-6} are reported.

- Data was generated in order to train different ANN's for time steps of 1×10^{-4}, 1×10^{-5}, 1×10^{-6} and 1×10^{-7} and a reaction mechanism involving seven chemical species and five elmentary reactions. Initial temperature were chosen randomly from an uniform distribution between 300 and 3000K. Initial species mass fractions were created via a procedure in which an undetermined system of question is set with the use of a random mixture fraction (between 0.1 and 1), elemental mass fractions of a fuel and oxydizer streams, and Bilger's definition, and then solved with a combination of LS and NNLS.

- An ODE system for a PFR was discretized with BDF , and solved for 10000 initial conditions with VODE for each of the aforementioned time steps until steady state was reached (1×10^{-9}). Ten percent of the data was separated to use as a validation set.

- Experimental limits in speedup with respect to the VODE solver were found for the number of layers and neurons in a feedforward ANN. Networks with 5, 6 and 7 hidden layers must have a number of neurons smaller than 128 in order to be faster. The number of neurons for networks of 2, 3 and 4 layers must be smaller than 256. These limits could be further reduced with more experiments. Tests to find the limits for the number of neurons in networks with a single hidden layer, and with more than 7 hidden layers could also be tried.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

- A thousand initial conditions were developed with the ANN integrator with each of the trained networks and the MSE was reported. There is a general tendency for the MSE to decrease as the number of layers and neurons increases, however the number of these is limited by the increase in computing complexity.

  ANN's trained for shorter time steps were shown to have a smaller MSE thant those with bigger ones. Graphically, it was observed that ANN's trained for bigger time steps had more difficulties capturing the correct gradients of the chemical species evolution, even if the predictions for the mass fractions were accurate. This is an issue, as the correct behavior is needed in order to obtain accurate reaction rates. Possible solutions include increasing the number of samples in the data set, and lowering the batch size during training.

  An investigation of how the error propagates was also carried out. During training, single step predictions had a validation MSE of the order of 1×10^{-9} to 1×10^{-8}, while using the ANN as an integrator for an initial condition until steady state lead to errors up to the order of 1×10^{-3}. For the purpose of this work the error propagation was considered as the ratio between the MSE after steady state and the MSE after a single prediction (as in the error of the validation set). It was shown that this ratio is between the orders of 1×10^{3} to 1×10^{5}, meaning ideally one would like to decrease the error of a single prediction in order to achieve an error reduction over the whole integration.

- Trying to enforce the physical conditions of non-negativity and summing up to one of the species mass fractions as part of the neural network proved difficult. Modifying the loss function, so that the sum of all components added up to one caused a situation where the gradients exploded during training. On the other hand, implementing non-negativity with the use of ReLU -and other- functions resulted in cases of vanishing gradient, where the learning stopped. Further investigation is needed on how to best implement these conditions.

- The best ANN's obtained for step sizes 1×10^{-4} and 1×10^{-5} were selected to be

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

coupled with OpenFOAM® . Both are feedforward networks, with LeakyReLU activation functions in each layer, trained for 50 epochs with batch size of 250. The activation of the output layer is replaced by a ReLU during coupling with OpenFOAM® to ensure non-negativity of the mass fractions, and the output was normalized.

The ANN trained for a step size of 1×10^{-5} has 6 inner layers with 64 neurons each. For a 1000 samples, average MSE was 7.705×10^{-5} with a deviation of 3.5×10^{-4}. It was able to compute the thousand samples in 67.77% of the time required by the VODE solver.

The ANN trained for a step size of 1×10^{-4} has 4 inner layers with 64 neurons each. For a 1000 samples, average MSE was 6.808×10^{-5} with a deviation of 1.5×10^{-5}. It was able to compute the thousand samples in 44.22% of the time required by the VODE solver.

- ANN's coupled with OpenFOAM® showed promising results when predicting species mass fractions with a certain degree of precision. However, as could be seen from the radial profiles, the error in the predictions seems to increase for higher temperatures. This affects less the mass fractions of CH_4, CO_2, H_2O, and O_2, however the problem is still existent.

  Mass fractions for H_2 and CO showed the least degree of accuracy and much more sensitive to the effects of temperature. This could be in part caused by the fact that these mass fractions are much smaller than those of the other species involved, which could have caused the learning procedure to be dominated by the gradients of the other more dominant species. As was seen while reporting the error propagation for the different ANN's, the error can grow up to the order of 10^{-3}, which could be to big when compared to chemical species whose mass fraction ranges lie in the orders of 10^{-5} and 10^{-6}. A possible fix without modifying the current architecture framework, would be to implement a normalization procedure, similar to the one for the temperature shown in Equation 4.18.

  The error for the mass fractions of N_2 might be caused by the network itself, as it

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

might be failing to predict that N_2 fractions must remain constant. One possible solution, which if succesfull might solve also the requirement for all species mass fractions to add up to one, would be to reduce the output of the network to six values and avoid the prediction of nitrogen. Then, the nitrogen fractions could be computed as the difference between 1 and the sum of all the other predictions.

- Performance results are not reported for the coupling of the ANN's and OpenFOAM® . The reason for this is that not significant speed up was measured during the simulations of the Sandia Flame D, sometimes with the ANN's being slower than the ODE solver. It might be possible that the ODE solver is using a time step that is bigger than the ones the ANN's were trained for, and thus can resolve the chemical species evolution faster. Another reason could be that the one way the ANN's were implemented in OpenFOAM® is not fully optimized, and thus at a disadvantage.

6.2 Outlook

The following are some ideas and suggestions on how could the results presented in this work could be improved in future studies:

- As was mentioned in the conclussions, one possible way to improve the results and decrease the error in the ANN's is to decrease the batch size and increase the number of training samples (whether by evaluating more initial conditions or increasing the stopping times), however with a significant hit on the training speed. Another downside connected to a larger training set are the memory requirements, which was a limitation on the available hardware. This was solved by implementing a method to read data from a *.csv file and transfer it to the GPU memory, and is useful as long as the whole data set can fit into the computer's RAM. For larger datasets, it is possible to generate data on the fly with a significant cost in performance, as the whole data set would have to be generated for every epoch.

- A possible way to decrease the propagation error is to implement a method to

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

apply the forward pass recursively according to the number of time steps taken between an initial condition and a target before the backpropagation, which is the same basic principle used by recursive neural networks [126]. As a disadvantage, the number of inputs and ouputs of the ANN would have to be the same. It is expected that this would require only minimal modifications when coupling with OpenFOAM® .

- Similar to Blasco [13], the reaction space -i.e. the temperature, mixture and mass fraction ranges- can be divided in smaller subranges, with ANN's trained for each particular sub-domain. This could increase the accuracy of the predicitions, as the different networks would not have to be able to generalize as much. The main drawbacks is that the number of experiments would increase significantly. The results reported in this work for architecture complexity, could help narrow down and reduce the number of performed experiments.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

References

[1] Gregory P. Smith, David M. Golden, Michael Frenklach, Nigel W. Moriarty, Boris Eiteneer, Thomas Goldenberg, Mikhail C. Bowman, Ronald K. Hanson, Soonho Song, William C. Jr. Gardiner, Vitali V. Lissianski, and Zhiwei Qin. *GRI-MECH 3.0*. 2000. URL: http://www.me.berkeley.edu/gri%7B%5C_%7Dmech.

[2] Matheus S. Serpa, Francis B. Moreira, Philippe O.A. Navaux, Eduardo H.M. Cruz, Matthias Diener, Dalvan Griebler, and Luiz Gustavo Fernandes. "Memory Performance and Bottlenecks in Multicore and GPU Architectures". In: *Proceedings - 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2019* (2019), pp. 233–236. DOI: 10.1109/EMPDP.2019.8671628.

[3] Herb Sutter. *The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software*. 2005. URL: http://www.gotw.ca/publications/concurrency-ddj.htm.

[4] Hesheng Bao. "Development and Validation of a New Eddy Dissipation Concept (EDC) Model for MILD Combustion". In: (2017). ISSN: 00349712.

[5] James W. Deardorff. "A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers". In: *Journal of Fluid Mechanics* 41.2 (1970), pp. 453–480. ISSN: 14697645. DOI: 10.1017/S0022112070000691.

[6] Baris A Sen. "Artificial neural networks based subgrid chemistry model for turbulent reactive flow simulations". PhD thesis. 2009, p. 243. ISBN: 9781109627510.

[7] Bjørn F Magnussen. "The Eddy Dissipation Concept a bridge between science and technology". In: *Eccomas* (2005).

[8] B. Magnussen. *On the structure of turbulence and a generalized eddy dissipation concept for chemical reaction in turbulent flow*. 1981. DOI: 10.2514/6.1981-42.

[9] F. C. Christo, A. R. Masri, E. M. Nebot, and S. B. Pope. "An integrated PDF/neural network approach for simulating turbulent reacting systems". In: *Symposium (International) on Combustion* 26.1 (1996), pp. 43–48. ISSN: 00820784. DOI: 10.1016/S0082-0784(96)80198-6.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[10] Boris A. Khesin and Serge. Tabachnikov. *Arnold : swimming against the tide*. American Mathematical Society, 2014, p. 165. ISBN: 9781470416997.

[11] G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function*". In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–3014. ISSN: 10009221. DOI: https://doi.org/10.1007/BF02551274.

[12] Kjetil O. Lye, Siddhartha Mishra, and Deep Ray. "Deep learning observables in computational fluid dynamics". In: *Journal of Computational Physics* 410 (2020), p. 109339. ISSN: 00219991. DOI: 10.1016/j.jcp.2020.109339. arXiv: 1903.03040.

[13] J. A. Blasco, N. Fueyo, C. Dopazo, and J. Ballester. "Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network". In: *Combustion and Flame* 113.1-2 (1998), pp. 38–52. ISSN: 00102180. DOI: 10.1016/S0010-2180(97)00211-3.

[14] Baris A. Sen and Suresh Menon. "Turbulent premixed flame modeling using artificial neural networks based chemical kinetics". In: *Proceedings of the Combustion Institute* 32 I.1 (2009), pp. 1605–1611. ISSN: 15407489. DOI: 10.1016/j.proci.2008.05.077.

[15] A. Kempf, F. Flemming, and J. Janicka. "Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by les". In: *Proceedings of the Combustion Institute* 30.1 (2005), pp. 557–565. ISSN: 15407489. DOI: 10.1016/j.proci.2004.08.182.

[16] M. D. Emami and A. Eshghinejad Fard. "Laminar flamelet modeling of a turbulent CH 4/H 2/N 2 jet diffusion flame using artificial neural networks". In: *Applied Mathematical Modelling* 36.5 (2012), pp. 2082–2093. ISSN: 0307904X. DOI: 10.1016/j.apm.2011.08.012.

[17] Ionut Porumbel, Andreea Cristina Petcu, Florin Gabriel Florean, and Constantin Eusebiu Hritcu. "Artificial neural networks for modeling of chemical source terms in CFD simulations of turbulent reactive flows". In: *Applied Mechanics and Materials* 555.June (2014), pp. 395–400. ISSN: 16627482. DOI: 10.4028/www.scientific.net/AMM.555.395.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[18] Ryno Laubscher. "Utilization of artificial neural networks to resolve chemical kinetics in turbulent fine structures of an advanced CFD combustion model". PhD thesis. Stellenbosch University, 2017. URL: https://scholar.sun.ac.za/handle/10019.1/101104.

[19] Ryno Laubscher. "Utilization of Basic Multi-layer Perceptron Artificial Neural Networks to Resolve Turbulent Fine Structure Chemical Kinetics Applied to a CFD Model of a Methane/Air Piloted Jet Flame". In: *Journal of Thermal Engineering* 4.2 (2018), pp. 1828–1846. ISSN: 2148-7847. DOI: 10.18186/journal-of-thermal-engineering.381838.

[20] Rishikesh Ranade and Tarek Echekki. "A framework for data-based turbulent combustion closure: A posteriori validation". In: *Combustion and Flame* 210 (2019), pp. 279–291. ISSN: 15562921. DOI: 10.1016/j.combustflame.2019.08.039.

[21] Corentin J. Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, and Thierry Poinsot. "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates". In: *Combustion and Flame* 203 (May 2019), pp. 255–264. ISSN: 15562921. DOI: 10.1016/j.combustflame.2019.02.019. arXiv: 1810.03691.

[22] Yan Zhang, Shijie Xu, Shenghui Zhong, Xue-Song Bai, Hu Wang, and Mingfa Yao. "Large eddy simulation of spray combustion using flamelet generated manifolds combined with artificial neural networks". In: *Energy and AI* 2 (Nov. 2020), p. 100021. ISSN: 26665468. DOI: 10.1016/j.egyai.2020.100021.

[23] Jian An, Hanyi Wang, Bing Liu, Kai Hong Luo, Fei Qin, and Guo Qiang He. "A deep learning framework for hydrogen-fueled turbulent combustion simulation". In: *International Journal of Hydrogen Energy* 45.35 (July 2020), pp. 17992–18000. ISSN: 03603199. DOI: 10.1016/j.ijhydene.2020.04.286.

[24] Joel H. Ferziger and Milovan Peric. *Computational Methods For Fluid Dynamics*. 3rd ed. Springer-Verlag Berlin Heidelberg, 2002, p. 426. ISBN: 978-3-642-56026-2.

[25] Robert J. Kee, Michael E. Coltrin, and Peter Glarborg. *Chemically Reacting Flow: Theory & Practice*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2003, p. 848. ISBN: 9780471461296. DOI: 10.1002/0471461296.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[26] Olivier Darrigol. "Between Hydrodynamics and Elasticity Theory: The First Five Births of the Navier-Stokes Equation". In: *Archive for History of Exact Sciences* 56.2 (2002), pp. 95–150. ISSN: 1432-0657. DOI: 10.1007/s004070200000.

[27] Roger Temam. *Navier–Stokes Equations and Nonlinear Functional Analysis*. 1995. ISBN: 0898713404. DOI: 10.1137/1.9781611970050.

[28] F M White. *Viscous Fluid Flow*. McGraw-Hill international edition. McGraw-Hill, 2006. ISBN: 9780071244930.

[29] T Poinsot and D Veynante. *Theoretical and Numerical Combustion*. T. Poinsot, D. Veynante, 2011. ISBN: 9782746639904.

[30] O. C. Zienkiewicz, R. L. Taylor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics: Seventh Edition*. 2013, pp. 1–544. ISBN: 9781856176354. DOI: 10.1016/C2009-0-26328-8.

[31] V. Penaz. *The compressibility of liquids*. Tech. rep. 12 , 1982. United States Geological Survey, 1982, pp. 917–920. DOI: 10.3133/b92.

[32] Jamal H. Abou-Kassem, S.M. Farouq Ali, and M. Rafiq Islam. "Single-Phase Flow Equation for Various Fluids". In: *Petroleum Reservoir Simulations*. Elsevier, 2006, pp. 207–282. DOI: 10.1016/b978-0-9765113-6-6.50013-7.

[33] John D. Jr. Anderson. *Fundamentals of Aerodynamics*. 5th ed. McGraw-Hill Education, 2010, p. 1106. ISBN: 978-0073398105.

[34] Malte Braack and Rolf Rannacher. *Adaptive Finite Element Methods for Low-Mach-Number Flows with Chemical Reactions*. Tech. rep. Institute for Applied Mathematics, University of Heidelberg, 1999.

[35] W. P. Jones and J. H. Whitelaw. "Calculation methods for reacting turbulent flows: A review". In: *Combustion and Flame* 48.C (Jan. 1982), pp. 1–26. ISSN: 00102180. DOI: 10.1016/0010-2180(82)90112-2.

[36] H K, Versteeg and Weeratunge Malalasekera. *Introduction: Computational Fluid Dynamics The Finite Volume Method*. Vol. 44. 2. 2006, pp. 193–193. ISBN: 9780131274983. DOI: 10.2514/1.22547.

[37] R.I Issa. "Solution of the implicitly discretised fluid flow equations by operator-splitting". In: *Numerical Heat Transfer; Part A: Applications* 62.1 (1986), pp. 40–65. ISSN: 15210634. DOI: 10.1080/10407782.2016.1173467.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[38] R. I. Issa, A. D. Gosman, and A. P. Watkins. "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme". In: *Journal of Computational Physics* 62.1 (1986), pp. 66–82. ISSN: 10902716. DOI: 10.1016/0021-9991(86)90100-2.

[39] Arnold Sommerfeld. "Ein Beitrag zur hydrodynamischen Erklärung der turbulenten Flüssigkeitsbewegungen". In: *Atti del IV Congresso Internazionale dei Matematici* 3.1883 (1908), pp. 116–124.

[40] P. Lanasa and L. Upp. *Fluid Flow Measurement: A Practical Guide to Accurate Flow Measurement Third edition*. Butterworth-Heinemann, 2014. ISBN: 9780124095243.

[41] Y. Bartosiewicz and M. Duponcheel. *Large-eddy simulation: Application to liquid metal fluid flow and heat transfer*. Elsevier Ltd., 2018, pp. 245–271. ISBN: 9780081019801. DOI: 10.1016/B978-0-08-101980-1.00017-X.

[42] A Kolmogorov. "The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers". In: *Akademiia Nauk SSSR Doklady* 30 (1941), pp. 301–305.

[43] V M Canuto. "Compressible Turbulence". In: *SCORe '96: Solar Convection and Oscillations and their Relationship*. Ed. by F P Pijpers, J Christensen-Dalsgaard, and C S Rosenthal. Dordrecht: Springer Netherlands, 1997, pp. 105–125. ISBN: 978-94-011-5167-2. DOI: 10.1007/978-94-011-5167-2_10.

[44] A Favre. "Statistical equations of turbulent gases". In: *Problems of hydrodynamics and continuum mechanics* (1969), pp. 231–266.

[45] Joseph Boussinesq. *Theórie analytiche de la chaleur mise en harmonie avec la thermodynamique et avec la théorie mécanique de la lumière*. Gauthier-Villars, 1903. DOI: 10.1007/bf01706901.

[46] Brian Launder and Dudley Brian Spalding. "The numerical computation of turbulent flows". In: *Computer Methods in Applied Mechanics and Engineering* 3.2 (1974), pp. 269–289. DOI: 10.1016/0045-7825(74)90029-2.

[47] Changsi Chen, James J. Riley, and Patrick A. McMurtry. "A study of Favre averaging in turbulent flows with chemical reaction". In: *Combustion and Flame* 87.3-4 (1991), pp. 257–277. ISSN: 00102180. DOI: 10.1016/0010-2180(91)90112-0.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[48] Theodore L. Brown, H. Eugene Jr. LeMay, Bruce E. Bursten, Catherine J. Murphy, and Patrick M. Woodward. *Chemistry The Central Science*. 12th ed. Prentice Hall, 2012, p. 1995. ISBN: 0-321-69672-7.

[49] Luonan Chen, Ruiqi Wang, Chunguang Li, and Kayuzuki Aihara. *Modeling Biomolecular Networks in Cells: Structures and Dynamics*. Vol. 110. 9. Springer-Verlag London, 2010. ISBN: 9781849962131.

[50] Chiranjib Kumar. Gupta. *Chemical Metallurgy. Principles and Practice*. Wiley-VCH, 2003. ISBN: 9780408005678. DOI: 10.1016/B978-0-408-00567-8.50011-0.

[51] Robert W. Carr. *Modeling of Chemical Reactions*. 1st ed. Vol. 42. Elsevier Science. ISBN: 9780444513663.

[52] Svante Arrhenius. "Über die Dissociationswärme und den Einfluss der Temperatur auf den Dissociationsgrad der Elektrolyte". In: (1889). ISSN: 0942-9352. DOI: 10.1515/zpch-1889-0408.

[53] N Peters. "Fifteen Lectures on Laminar and Turbulent Combustion". In: *ERCOFTAC Summer School* April (1992), pp. 1–245.

[54] F A Lindemann, Svante Arrhenius, Irving Langmuir, N R Dhar, J Perrin, and W C McC. Lewis. "Discussion on "the radiation theory of chemical action"". In: *Trans. Faraday Soc.* 17.0 (1922), pp. 598–606. DOI: 10.1039/TF9221700598. URL: http://dx.doi.org/10.1039/TF9221700598.

[55] P. H. Stewart, C. W. Larson, and D. M. Golden. "Pressure and temperature dependence of reactions proceeding via a bound complex. 2. Application to 2CH3 → C2H5 + H". In: *Combustion and Flame* 75.1 (1989), pp. 25–31. ISSN: 00102180. DOI: 10.1016/0010-2180(89)90084-9.

[56] D. O. Allison. "Polynomial approximations of thermodynamic properties of arbitrary gas mixtures over wide pressure and density ranges". In: August (1972).

[57] Willi Gujer. "Systems Analysis for Water Technology". In: Springer-Verlag Berlin Heidelberg, 2008. ISBN: 978-3-540-77278-1. DOI: https://doi.org/10.1007/978-3-540-77278-1_6.

[58] David Rosa, Paulo Goes, and João Manzi. "Minimum Entropy Based PFR Analysis". In: *Computer Aided Chemical Engineering* 43 (2018), pp. 507–512. ISSN: 15707946. DOI: 10.1016/B978-0-444-64235-6.50090-5.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[59] Lanny D. Schmidt. *The Engineering of Chemical Reactions*. Oxford University Press, 2004. ISBN: 9780195169256.

[60] Stephen R. Turns. *An Introduction to Combustion Concepts and Applications*. McGraw-Hill, 2012. ISBN: 9780073380193. URL: www.mhhe.com.

[61] David S-K. Ting. "Chapter 4 - Turbulence Scales". In: *Basics of Engineering Turbulence*. Ed. by David S-K. Ting. Academic Press, 2016, pp. 69–98. ISBN: 978-0-12-803970-0. DOI: https://doi.org/10.1016/B978-0-12-803970-0.00004-0.

[62] Markus Bösenhofer, Eva Maria Wartha, Christian Jordan, and Michael Harasek. "The eddy dissipation concept-analysis of different fine structure treatments for classical combustion". In: *Energies* 11.7 (2018), pp. 1–21. ISSN: 19961073. DOI: 10.3390/en11071902.

[63] Christopher P. Stone and Fabrizio Bisetti. "Comparison of ODE solvers for chemical kinetics and reactive CFD applications". In: *52nd Aerospace Sciences Meeting* January 2014 (2014), pp. 1–19. DOI: 10.2514/6.2014-0822.

[64] S Sivaji Ganesh. *Lecture notes on Ordinary Differential Equations Plan of lectures*. Tech. rep.

[65] G Peano. "Démonstration de l'intégrabilité des équations différentielles ordinaires". In: *Mathematische Annalen* 37.2 (1890), pp. 182–228. ISSN: 1432-1807. DOI: 10.1007/BF01200235.

[66] Shuyu Sun and M. F. Wheeler. "Analysis of Discontinuous Galerkin Methods for Multicomponent Reactive Transport Problems". In: *Computers and Mathematics with Applications* 52.5 (Sept. 2006), pp. 637–650. ISSN: 08981221. DOI: 10.1016/j.camwa.2006.10.004.

[67] E. Lindelöf. "Sur l'application de la méthode des approximations successives aux équations différentielles ordinaires du premier ordre." In: *Comptes rendus hebdomadaires des séances de l'Académie des Sciences* 116 (1894), pp. 454–457.

[68] Dirk Praetorius and Michael Innerberger. "Lectures in Numerics of Differential Equations". In: (2019). URL: https://www.asc.tuwien.ac.at/%7B~%7Dpraetorius/?open=numodes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[69] M. N. Spijker. "Stiffness in numerical initial-value problems". In: *Journal of Computational and Applied Mathematics* 72.2 (1996), pp. 393–406. ISSN: 03770427. DOI: 10.1016/0377-0427(96)00009-X.

[70] Greg Fasshauer. "Lectures on Numerical Methods for Differential Equations/Computational Mathematics II". In: (2007). URL: http://www.math.iit.edu/%7B~%7Dfass/478%7B%5C_%7D578%7B%5C_%7Dhandouts.html.

[71] Germund G Dahlquist. "A special stability problem for linear multistep methods". In: *BIT Numerical Mathematics* 3.1 (1963), pp. 27–43. ISSN: 1572-9125. DOI: 10.1007/BF01963532.

[72] Peter N. Brown, George D. Byrne, and Alan C. Hindmarsh. "VODE: A Variable-Coefficient ODE Solver". In: *SIAM Journal on Scientific and Statistical Computing* 10.5 (Sept. 1989), pp. 1038–1051. ISSN: 0196-5204. DOI: 10.1137/0910062. URL: https://computing.llnl.gov/casc/nsde/pubs/207532.pdf.

[73] A L Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM J. Res. Dev.* 3 (1959), pp. 210–229.

[74] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019, p. 160. ISBN: 978-1999579500.

[75] Dan Simovici. *Mathematical Analysis For Machine Learning And Data Mining*. World Scientific Publishing, 2018. ISBN: 9789813229686.

[76] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2017, p. 800. ISBN: 9780262035613. URL: http://www.deeplearningbook.org.

[77] M. Foster and C.S. Sherrington. . *Textbook of Physiology, volume 3*. 7th ed. London: Macmillan, 1897, p. 929.

[78] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259.

[79] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 0033295X. DOI: 10.1037/h0042519.

[80] Patrick Henry Winston. *Artificial intelligence*. 3rd ed. Addison-Wesley, 1992, p. 737. ISBN: 0201533774.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[81] Eugenio Rossini. "Deep Learning and Nonlinear PDEs in High-Dimensional Spaces". In: (2018).

[82] Leonardo Ferreira Guilhoto. *An Overview Of Artificial Neural Networks for Mathematicians*. Tech. rep. University of Chicago, pp. 1–25. URL: http://math.uchicago.edu/%7B~%7Dmay/REU2018/REUPapers/Guilhoto.pdf.

[83] Catherine F. Higham and Desmond J. Higham. "Deep Learning: An Introduction for Applied Mathematicians". In: (Jan. 2018). arXiv: 1801.05894.

[84] *ReLu Definition — DeepAI*. URL: https://deepai.org/machine-learning-glossary-and-terms/relu (visited on 09/24/2020).

[85] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen. Diploma. Technische Universität München". In: *Diploma, Technische Universität München* 91.1 (1991).

[86] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. "Dying ReLU and Initialization: Theory and Numerical Examples". In: (2019). arXiv: 1903.06733.

[87] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[88] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: (2015). arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

[89] Sergios Theodoridis. "Stochastic Gradient Descent". In: *Machine Learning*. Elsevier, Jan. 2015, pp. 161–231. DOI: 10.1016/b978-0-12-801522-3.00005-7.

[90] K P Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series. MIT Press, 2012. ISBN: 9780262018029.

[91] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: (Sept. 2016). arXiv: 1609.04747.

[92] Zhangyang Wang. "Bi-Level Sparse Coding: A Hyperspectral Image Classification Example". In: *Deep Learning Through Sparse and Low-Rank Modeling*. Elsevier, Jan. 2019, pp. 9–29. DOI: 10.1016/b978-0-12-813659-1.00002-0.

[93] S Sra, S Nowozin, and S J Wright. *Optimization for Machine Learning*. Neural information processing series. MIT Press, 2012. ISBN: 9780262016469.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[94] Diederik P Kingma and Jimmy Lei Ba. *Adam: A Method for Stochastic Optimization*. Tech. rep. arXiv: 1412.6980v9.

[95] Ruoyu Sun. *Optimization for deep learning: theory and algorithms*. Tech. rep. 2019. arXiv: 1912.08957v1.

[96] Sylvain Gugger and Jeremy Howard. *AdamW and Super-convergence is now the fastest way to train neural nets · fast.ai*. 2018. URL: https://www.fast.ai/2018/07/02/adam-weight-decay/ (visited on 11/21/2020).

[97] Nitish Shirish Keskar and Richard Socher. *Improving Generalization Performance by Switching from Adam to SGD*. Tech. rep. arXiv: 1712.07628v1.

[98] Guillaume Carlier Mqef. "Mathematics for Machine Learning". In: *Quantitative literacy: Why numeracy matters for schools* c (2009), pp. 533–540. DOI: 10.1007/b97511.

[99] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0.

[100] Atılım Güne¸, Güne¸s Baydin, Barak A Pearlmutter, and Jeffrey Mark Siskind. "Automatic Differentiation in Machine Learning: a Survey". In: *Journal of Machine Learning Research* 18 (2018), pp. 1–43.

[101] F Girosi and T Poggio. "Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant". In: *Neural Computation* 1.4 (Dec. 1989), pp. 465–469. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.465.

[102] J Lin and R Unbehauen. "On the Realization of a Kolmogorov Network". In: *Neural Computation* 5.1 (Jan. 1993), pp. 18–20. ISSN: 0899-7667. DOI: 10.1162/neco.1993.5.1.18.

[103] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 08936080. DOI: 10.1016/0893-6080(91)90009-T.

[104] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867. ISSN: 08936080. DOI: 10.1016/S0893-6080(05)80131-5.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[105] Vugar E. Ismailov. "On the approximation by neural networks with bounded number of neurons in hidden layers". In: *Journal of Mathematical Analysis and Applications* 417.2 (Sept. 2014), pp. 963–969. ISSN: 10960813. DOI: 10.1016/j.jmaa.2014.03.092.

[106] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. "The expressive power of neural networks: A view from the width". In: *arXiv* Nips (2017), pp. 1–9.

[107] Patrick Kidger and Terry Lyons. *Universal Approximation with Deep Narrow Networks*. Tech. rep. 2020, pp. 1–22. arXiv: 1905.08539v2.

[108] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, 2009. ISBN: 1441412697. DOI: 10.5555/1593511.

[109] David G Goodwin, Raymond L Speth, Harry K Moffat, and Bryan W Weber. *Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes*. 2018. DOI: 10.5281/zenodo.1174508. URL: https://www.cantera.org.

[110] The Pandas development team. *pandas-dev/pandas: Pandas*. 2020. DOI: 10.5281/zenodo.3509134.

[111] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[112] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J van der Walt, Matthew Brett, Joshua Wilson, K Jarrod Millman, Nikolay Mayorov, Andrew R J Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, \.Ilhan Polat, Yu Feng, Eric W Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E A Quintero, Charles R Harris, Anne M Archibald, Antônio H Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. "{SciPy} 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[113] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H Wallach, H Larochelle, A Beygelzimer, F d\textquotesingle Alché-Buc, E Fox, and R Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[114] Ronan Collobert, Samy Bengio, and Johnny Marithoz. *Torch: A Modular Machine Learning Software Library*. 2002.

[115] *Comet – Build better models faster!* URL: https://www.comet.ml/site/ (visited on 11/02/2020).

[116] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques". In: *Computers in Physics* 12.6 (1998), p. 620. ISSN: 08941866. DOI: 10.1063/1.168744.

[117] Charles K. Westbrook and Frederick L. Dryer. "Chemical kinetic modeling of hydrocarbon combustion". In: *Progress in Energy and Combustion Science* 10.1 (1984), pp. 1–57. ISSN: 03601285. DOI: 10.1016/0360-1285(84)90118-7.

[118] R. W. Bilger. "Turbulent jet diffusion flames". In: *Progress in Energy and Combustion Science* 1.2-3 (1976), pp. 87–109. ISSN: 03601285. DOI: 10.1016/0360-1285(76)90022-8.

[119] Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra*. 2018. ISBN: 9781316518960. DOI: 10.1017/9781108583664.

[120] Donghui Chen and Robert J. Plemmons. "Nonnegativity constraints in numerical analysis". In: *The Birth of Numerical Analysis* (2009), pp. 109–140. DOI: 10.1142/9789812836267_0008.

[121] Robert Barlow and Jonathan Frank. *Piloted CH4/Air Flames C, D, E, and F; Release 2.1*. Tech. rep. June. 2007, pp. 1–12. URL: http://www.sandia.gov/TNF/DataArch/FlameD/SandiaPilotDoc21.pdf.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[122] Sandia National Laboratory. *Flame D*. URL: https://www.sandia.gov/TNF/DataArch/FlameD.html (visited on 11/17/2020).

[123] Ch Schneider, A. Dreizler, J. Janicka, and E. P. Hassel. "Flow field measurements of stable and locally extinguishing hydrocarbon-fuelled jet flames". In: *Combustion and Flame* 135.1-2 (2003), pp. 185–190. ISSN: 00102180. DOI: 10.1016/S0010-2180(03)00150-0.

[124] R. S. Barlow, J. H. Frank, A. N. Karpetis, and J. Y. Chen. "Piloted methane/air jet flames: Transport effects and aspects of scalar structure". In: *Combustion and Flame* 143.4 (2005), pp. 433–449. ISSN: 00102180. DOI: 10.1016/j.combustflame.2005.08.017.

[125] Eva-Maria Wartha. "A Study on New Developments and Chemical Time Scale Definitions for the Eddy Dissipation Concept". PhD thesis. Technical University of Vienna, 2017.

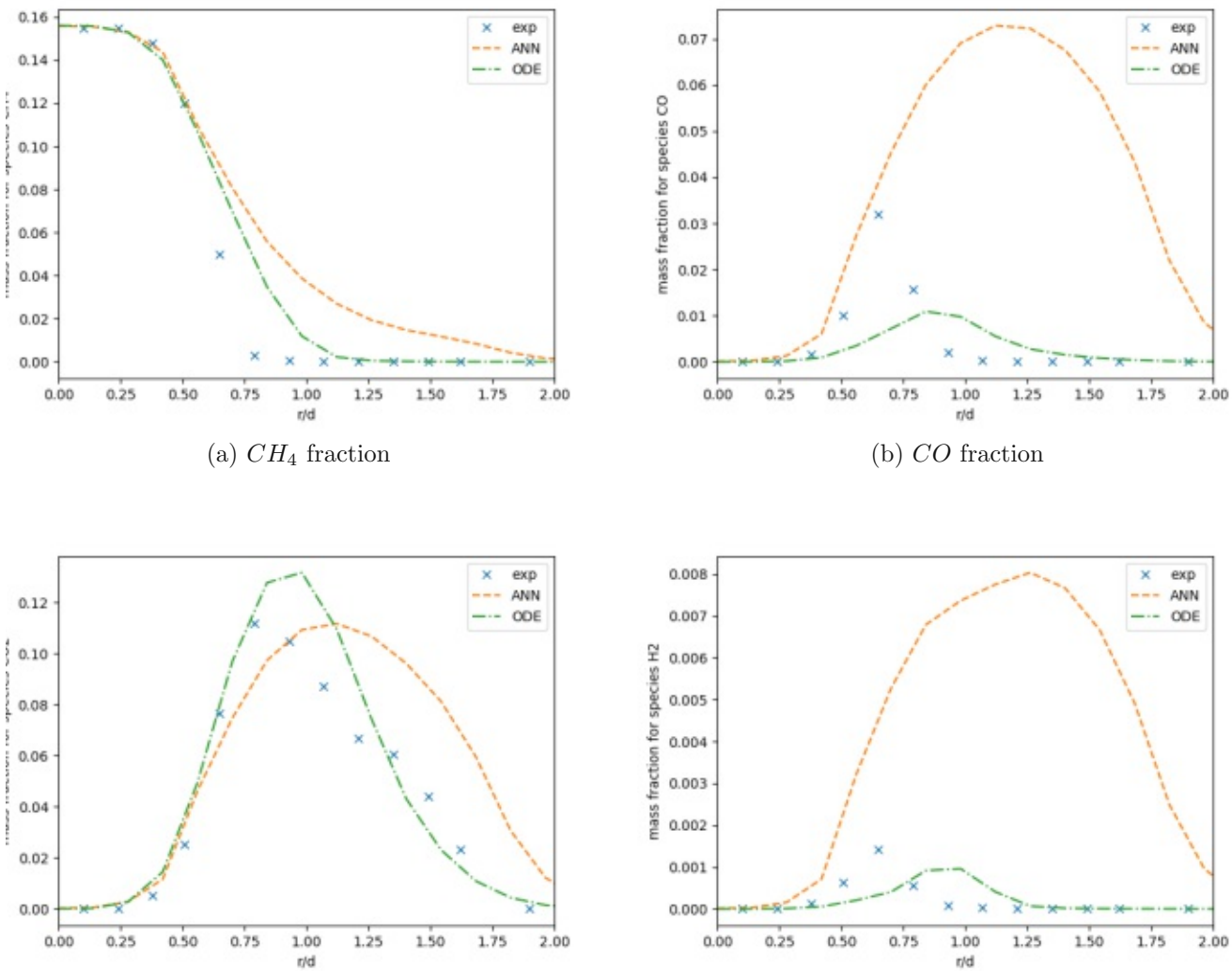
[126] C Goller and A Kuchler. "Learning task-dependent distributed representations by backpropagation through structure". In: *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 1. June 1996, 347–352 vol.1. DOI: 10.1109/ICNN.1996.548916.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

A Appendix

A.1 Additional Results

A.1.1 Radial profiles at x/d=3



(a) CH_4 fraction

(b) CO fraction

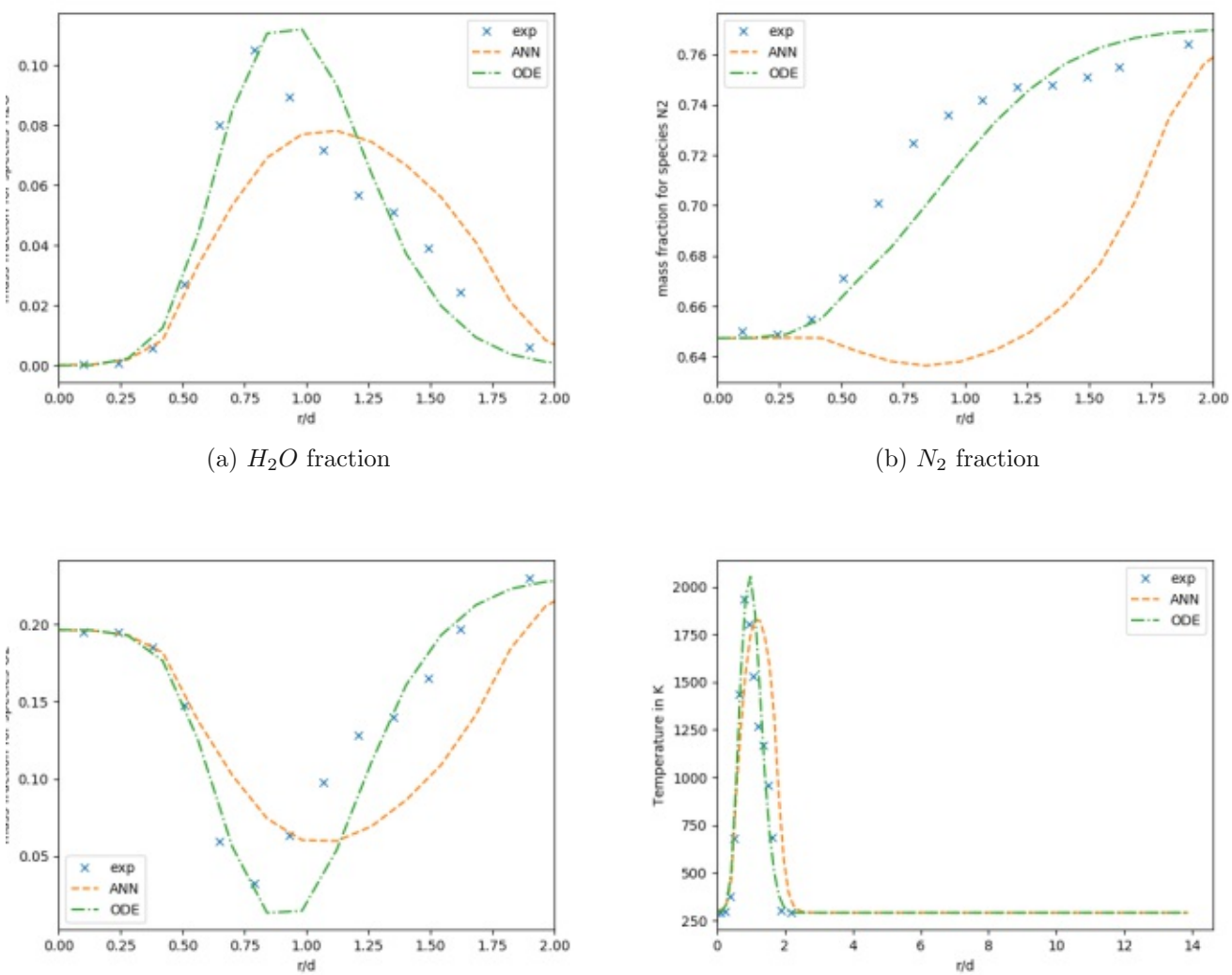
(c) CO_2 fraction

(d) H_2 fraction

Figure A.1: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 3$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Figure A.1 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 3$. The predictions for CH_4 and CO_2 are still good. CO and H_2 innacuracies after $r/d = 0.60$ have increased significantly compared to the cases with $x/d = 1$ and $x/d = 2$.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

(d) Temperature

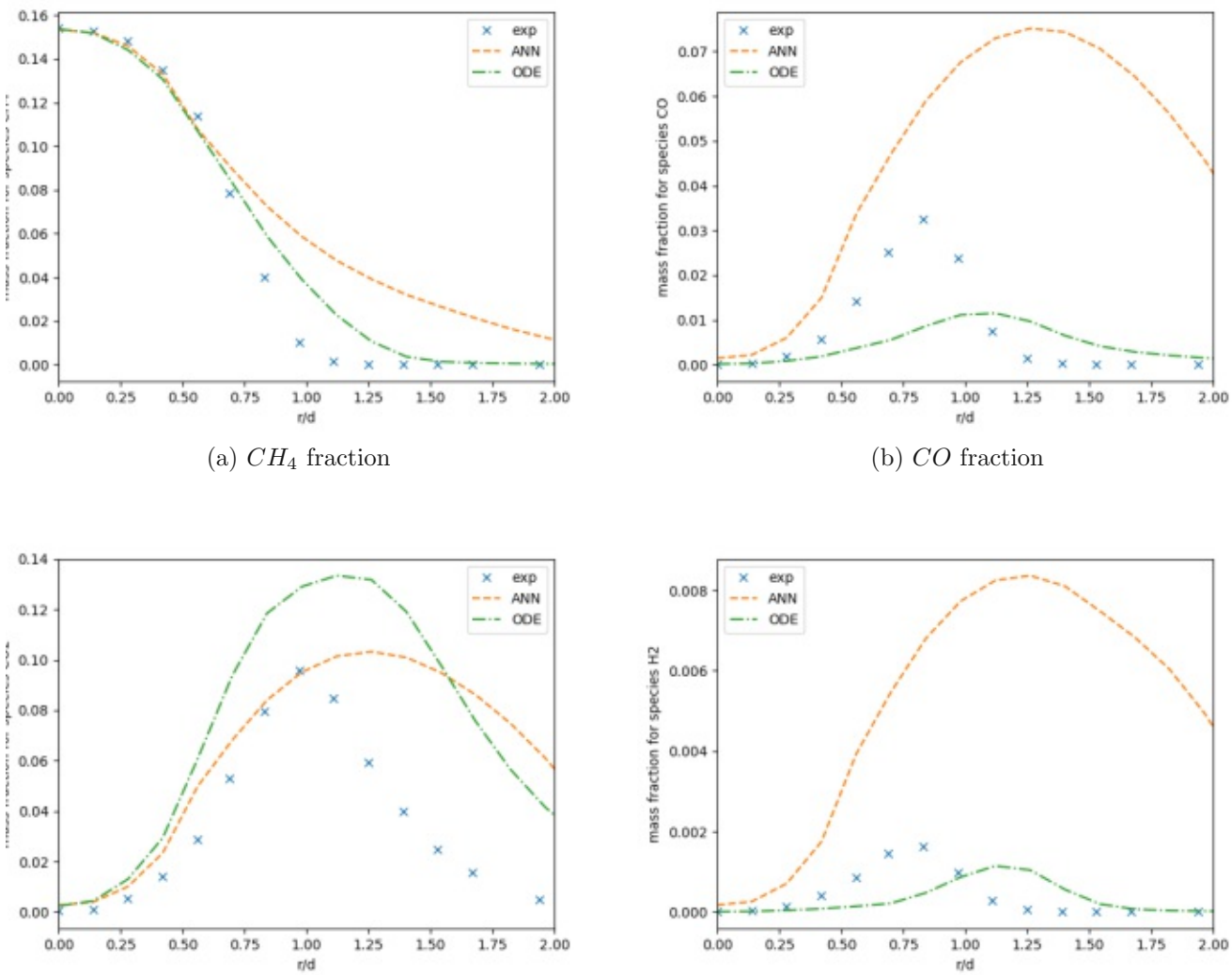
Figure A.2: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 3$

Figure A.2 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 3$. The predictions for H_2O, O_2 seem to be accurate. N_2 error

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

continues to increase exarcebating the problematic behavior (decrease in mass fraction).

A.1.2 Radial profiles at x/d=7.5



(a) CH_4 fraction

(b) CO fraction

(c) CO_2 fraction

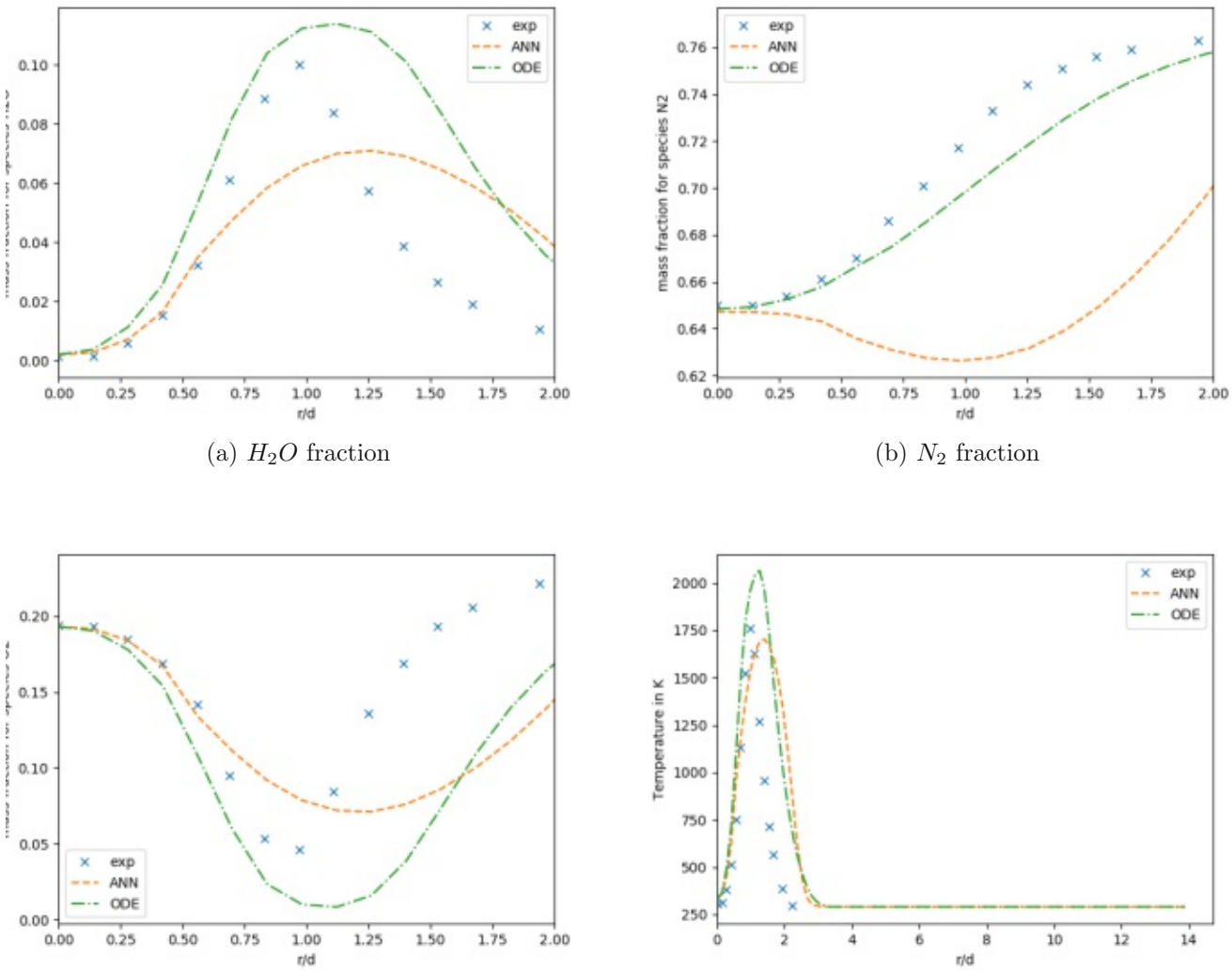
(d) H_2 fraction

Figure A.3: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 7.5$

Figure A.3 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 7.5$. Predictions are similar for the mass fractions presented in the radial profiles

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

seen in Figure A.1 with a larger deviation from the experimental data. The predictions for CH_4 loose precision for r/d higher than 0.75, while those for CO_2 do after $r/d = 1$. Mass fractions for CO and H_2 present a higher degree of innacuracy.



(a) H_2O fraction

(b) N_2 fraction

(c) O_2 fraction

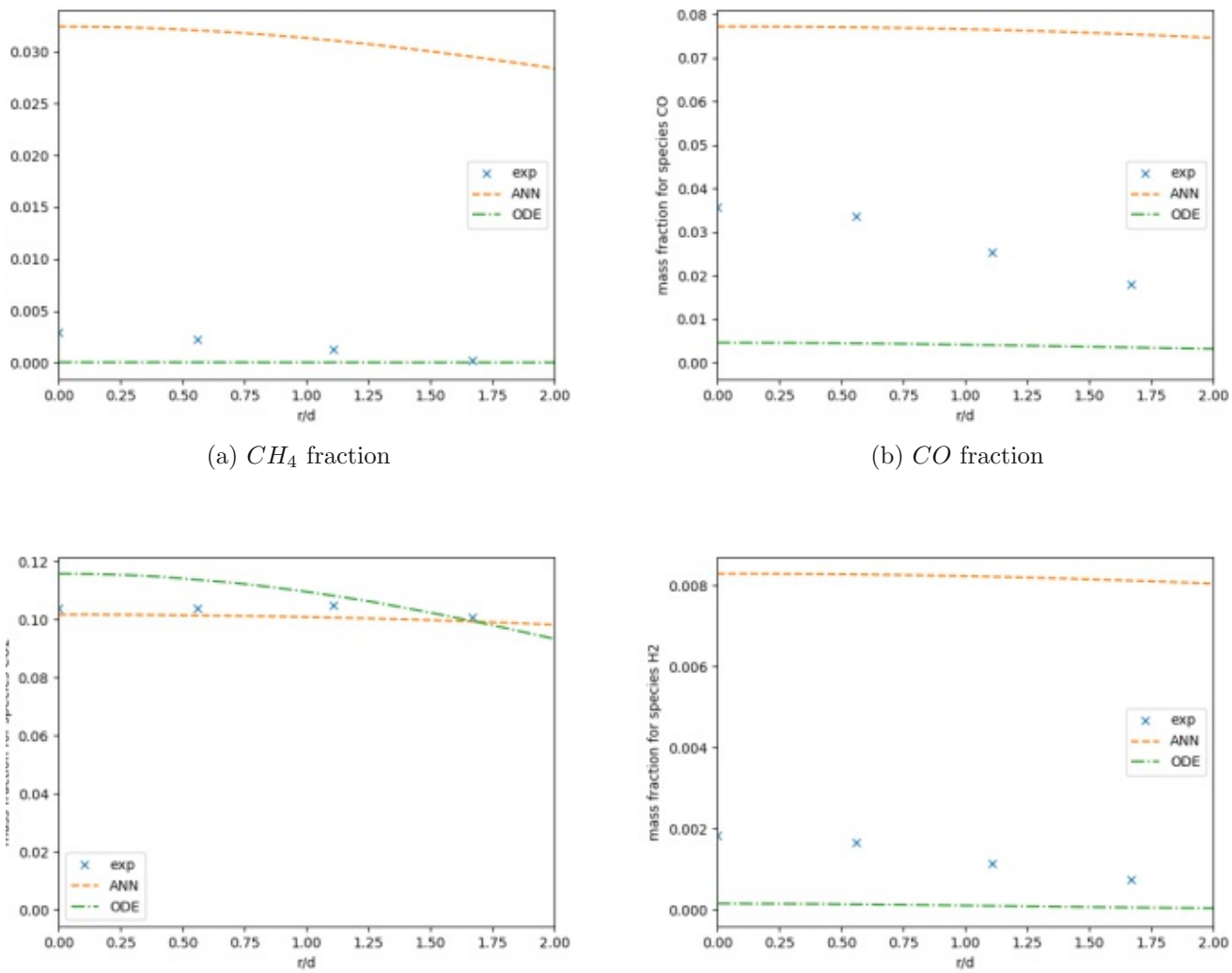
(d) Temperature

Figure A.4: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 7.5$

Figure A.4 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 7.5$. Water and oxygen predictions seem to be somewhat accurate, while nitrogen mass fractions seem to decrease initially, which is an undesired behavior.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

A.1.3 Radial profiles at x/d=45



(a) CH_4 fraction

(b) CO fraction

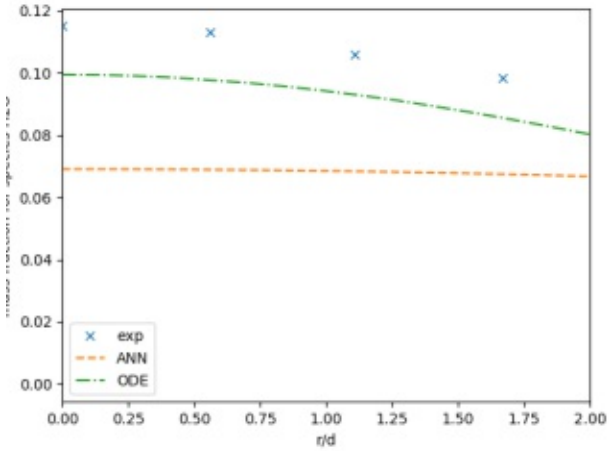
(c) CO_2 fraction

(d) H_2 fraction

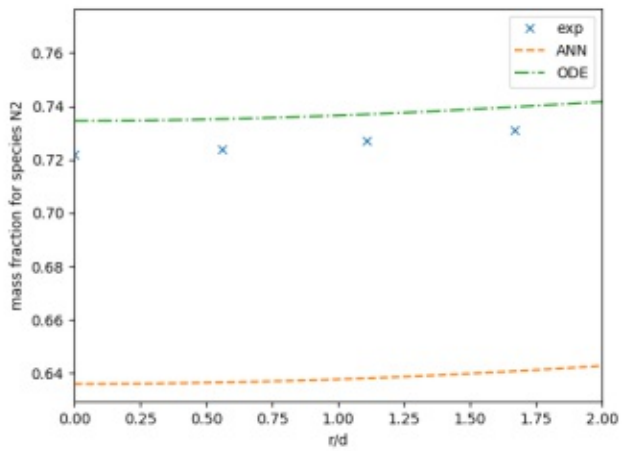
Figure A.5: Species mass fractions of CH_4, CO, CO_2 and H_2 for $x/d = 45$

Figure A.5 shows radial profiles for the mass fractions of CH_4, CO, CO_2 and H_2 with $x/d = 45$. Only the pedictions for CO_2 are still accurate. Somehow all curves seem to follow the shape of the integrator ones but offset.

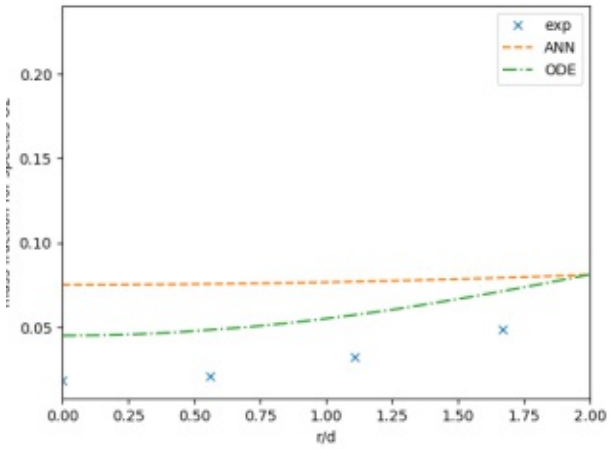
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN



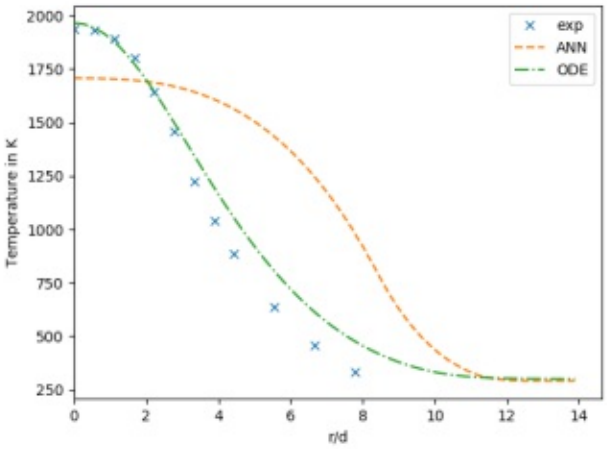
(a) H_2O fraction



(b) N_2 fraction



(c) O_2 fraction



(d) Temperature

Figure A.6: Species mass fractions of H_2O, N_2, O_2 and Temperature for $x/d = 45$

Figure A.6 shows radial profiles for the mass fractions of H_2O, N_2, O_2 and temperature with $x/d = 45$. All predictions seem to be no longer accurate and show almost no variation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.