

Artificial Neural Networks for Thermochemistry Incorporating Conservation Laws

Thomas Readshaw

Imperial College London

Department of Mechanical Engineering

*Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering of Imperial College
London and the Diploma of Imperial College*

December 2022

Declarations

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Originality

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others except where clearly specified in the text.

Thomas Readshaw

December 2022

Acknowledgements

I would like to thank my supervisor, Dr. Stelios Rigopoulos, for his excellent guidance and support of my work, as well as his willingness to allow me to explore ideas in my own way. I would also like to thank Prof. W.P. Jones for his helpful technical contributions and feedback. I would like to acknowledge the Engineering and Physical Sciences Research Council and Rolls-Royce plc for their financial support in the form of a Doctoral Training Partnership and Cooperative Award in Science and Technology.

I would also like to thank all of my friends from the offices of CAGB 674/675 and elsewhere: Malamas, Larry, Sun, Cary, Danny, Daniel, Stratos, Nav, Stephen, Sam and Michael for their welcome distractions and constant cheeriness. Particular thanks are also in order for Tianjie for his assistance and thoughts, and for moderating some of my wackier ideas with good humour and healthy realism. I am indebted to Dan (aka Jerry) for his companionship, our frequent lively discussions, our badminton sessions, and for always providing a pen for the crossword. I would also like to extend my thanks to all of the squash friends I have met over the years, but particularly Dave, Patrick, Pam, Franki, James, Steve and Seigo, without whom my Imperial experience would have been nowhere near as enjoyable.

I would like to express my gratitude to my Mum and Dad, Joe, Matthew, and Sherlock (but not Trevor), as well as the rest of my family, for their unwavering love and support. Finally, to Rachel, for her unconditional love, for always making me laugh and for her not infrequent motivation.

"With four parameters I can fit an elephant,
and with five I can make him wiggle his trunk."

John von Neumann

Related Work

T. Readshaw, S. Rigopoulos, and W.P. Jones. (In press). On the incorporation of conservation laws in machine learning tabulation of kinetics for reacting flow simulation. *Physics of Fluids*, 2023.

T. Readshaw, L.L.C. Franke, W.P. Jones and S. Rigopoulos. (Under review). Simulation of turbulent premixed flames with machine learning - tabulated thermochemistry.

T. Ding, T. Readshaw, S. Rigopoulos, and W. P. Jones. Machine learning tabulation of thermochemistry in turbulent combustion: An approach based on hybrid flamelet/random data and multiple multilayer perceptrons. *Combustion and Flame*, 231:111493, 2021.

T. Readshaw, T. Ding, S. Rigopoulos, and W.P. Jones. Modeling of turbulent flames with the large eddy simulation–probability density function (LES–PDF) approach, stochastic fields, and artificial neural networks. *Physics of Fluids*, 33:035154, 2021.

Abstract

The real time integration of the chemical kinetics in reacting flows consumes the majority of the computational time required when using many simulation techniques. Tabulation of the chemical kinetics using artificial neural networks (ANNs) has the potential to alleviate this computational bottleneck. ANN approaches should have high prediction accuracy but remain as general as possible in order to increase the range of applications and so the potential speed-up. Additionally, it is desirable for ANNs to be physically consistent with the chemical kinetics they replace.

In this thesis the generality of ANN approaches is advanced using a random data generation method, which takes data from canonical combustion problems and modifies it to expand the composition space covered. ANN accuracy is improved by using a quasi-second order training algorithm, an automatic approach to regularisation and by predicting the reaction behaviour of each species individually. The accuracy of the ANN tabulation method is demonstrated via its application to large eddy simulations using transported probability density function closure for the filtered reaction source terms, with numerical solution using stochastic fields.

Statistics obtained from two such simulations of a non-premixed CH_4 /air jet flame with a thirty one species mechanism using conventional real time integration and the ANN tabulation method respectively, show excellent agreement for major species and good agreement for minor species, while the time required for reaction source term calculations is reduced eighteen fold. The range of applications of ANN thermochemistry tabulation is then expanded by simulating turbulent swirl-stabilised bluff-body CH_4 /air premixed jet flames with the same thirty one species mechanism. Recirculation zones near the bluff-body in these flames require very high prediction accuracy to prevent the accumulation of prediction errors affecting downstream predictions, particularly for minor species. The required prediction accuracy is therefore obtained by automatically dividing the output ranges for each species by their magnitudes and using separate ANNs for each. Excellent agreement is observed between statistics collected from simulations using the ANN method and those obtained when using conventional integration, with a reaction source term calculation speed-up of fourteen times.

A method for ensuring physical consistency by enforcing the conservation laws when using ANN thermochemistry tabulation is then developed and applied to simulations of methane/air and propane/air flames. The results of these simulations show that enforcing the conservation laws positively impacts the accuracy of ANN predictions while incurring an insignificant extra computational cost. This conservation method can be readily appended to existing ANN architectures and also be applied to data generation and the prediction of reaction rates.

Contents

List of figures	16
List of tables	20
Nomenclature	22
1 Introduction	27
1.1 Background and motivation	27
1.2 Overview of thermochemistry tabulation with artificial neural networks	31
1.3 Contributions of the thesis	37
1.4 Thesis structure	39
2 Theoretical background	41
2.1 Combustion	41
2.1.1 Combustion chemistry and reaction mechanisms	41
2.1.1.1 Basics of the composition space	41
2.1.1.2 Modelling reaction mechanisms	44
2.1.2 Conservation equations	46
2.2 Large eddy simulations of combustion with the transported probability density function method and stochastic fields	49
2.2.1 Filtered equations of fluid motion	49

2.2.2	The transported probability density function method and solution with stochastic fields	52
2.2.3	Consistency of the stochastic fields	57
2.3	Machine learning	58
2.3.1	Multi-layer perceptrons	59
2.3.1.1	Principles of training	62
2.3.1.2	Training with the Levenberg-Marquardt algorithm	64
2.3.1.3	Preventing overfitting with Bayesian regularisation	66
2.3.2	The self-organising map	67
2.4	Thermochemistry tabulation with artificial neural networks	69
2.4.1	The SOM-MLP method	69
2.4.2	Random data generation	70
3	Development of a machine learning methodology for non-premixed flames	77
3.1	An artificial neural network architecture for improved generalisation	77
3.2	Application to the Sydney L flame	85
3.2.1	Experimental and simulation details	85
3.2.2	Sydney L flame simulation results	87
4	Development of a machine learning methodology for premixed flames	94
4.1	Generating training data for premixed flames	94
4.2	Artificial neural network training for premixed flames	96
4.3	Application to the Cambridge/Sandia swirl burner	99
4.3.1	Experimental and simulation details	99
4.3.2	Cambridge/Sandia swirl burner simulation results	102

5	A method for incorporating conservation laws in combustion thermochemistry	112
5.1	General principles of conservation method	113
5.1.1	Step 1: Generate initial predictions	118
5.1.2	Step 2: Select residual species	118
5.1.3	Step 3: Calculate mass fractions of residual species	120
5.1.4	Step 4: Rectify final residual species if needed	122
5.2	Application to reaction source term predictions	123
5.2.1	Results for CH ₄ /air flames with GRI 1.2	126
5.2.2	Results for C ₃ H ₈ /air flames with USC Mech II	130
5.3	Application to random data generation	134
5.4	Application to reaction rate predictions	138
6	Conclusions	139
6.1	Summary of thesis achievements	139
6.2	Suggestions for future work	142
A	Comparison of the moments arising from the modelled probability density function equation and the stochastic fields	144
A.1	The modelled probability density function equation	144
A.2	The stochastic fields	146
B	Example of satisfying conservation laws for a composition	151
	References	156

List of Figures

2.1	Diagram of an MLP with N_{in} inputs and N_{out} outputs showing neurons (circles) arranged in layers connected by weights (lines).	59
2.2	Modification to the Levenberg-Marquardt algorithm to allow the training of large networks with large data sets.	66
2.3	Composition space regions found in canonical combustion problem data using the SOM, denoted with different colours.	70
2.4	Indicative diagram showing relations between different composition spaces.	71
2.5	Indicative diagrams of composition spaces from 1D non-premixed laminar flames as well as the desired expanded composition spaces. .	73
2.6	Boundaries on randomly generated mass fraction with $b_3 = 4/3$	75
3.1	Comparison of composition space coverage of original 1D flamelet data and randomly generated data.	79
3.2	Performance in CO_2 and OH using architectures A-E.	82
3.3	Prediction error in individual data points as a function of the Euclidean distance between the respective scaled input and the closest SOM neuron.	83
3.4	Steady state flamelet profiles for temperature and CO mass fraction at strain rates 25, 50, 100, 300 and 530 /s using DI and the architecture E ANNs.	84
3.5	Steady state flamelet profiles for temperature and CO mass fraction at strain rate 1 /s using DI and the architecture E ANNs.	84
3.6	Radial and axial grid spacing at $x/D = 0$ and $r/D = 0$ respectively for the grid used in the Sydney L simulations.	87

3.7	Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI, as well as experimental values, for temperature, mixture fraction and N_2 and O_2 mass fraction.	89
3.8	Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI, as well as experimental values, for CO_2 , H_2O , CO and H_2 mass fraction.	90
3.9	Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for OH , H , O and H_2O_2 mass fraction.	91
3.10	Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for CH_3 , C_2H_2 , C_2H_4 and C_2H_6 mass fraction.	92
3.11	Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for HO_2 , CH_3OH , $HCCO$ and $HCCOH$ mass fraction.	93
4.1	Comparison of composition space coverage of original 1D premixed data and randomly generated data, showing vertical lines at constant enthalpy values in the premixed data.	97
4.2	Randomly generated data divided into three data sets, based on the boundaries in Table 4.4.	99
4.3	Radial and axial grid spacing at $x = 0$ and $r = 0$ respectively for the grid used in the SWB5 and SWB6 simulations.	102
4.4	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for temperature, equivalence ratio, CH_4 and O_2 mass fraction.	104
4.5	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for H_2O , CO_2 , CO and H_2 mass fraction.	105

4.6	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for H, O, OH and H_2O_2 mass fraction.	106
4.7	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for HO_2 , C_2H_2 , C_2H_4 and C_2H_6 mass fraction.	107
4.8	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for CH_2 , CH_3 , CH_2OH and CH_3O mass fraction.	108
4.9	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB5 simulations using ANNs and DI, as well as experimental values, for temperature, equivalence ratio, CH_4 and O_2 mass fraction.	109
4.10	Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB5 simulations using ANNs and DI, as well as experimental values, for H_2O , CO_2 , CO and H_2 mass fraction.	110
5.1	Illustration of a large overprediction of Y_2^* resulting in a mass fraction of Y_1^{con} which is calculated to be negative in order to satisfy the conservation law for the element under consideration.	115
5.2	Illustration of a large overprediction of Y_2^* which can be corrected in order to satisfy the conservation law for the element under consideration by calculating Y_2^{con}	116
5.3	CO mass fraction profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	127
5.4	H_2 mass fraction profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	127

5.5	Molar H/C ratio profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	128
5.6	Molar O/N ratio profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs when using different methods for reaction source term integration.	128
5.7	CO mass fraction profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	131
5.8	H ₂ mass fraction profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	131
5.9	Molar H/C ratio profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	132
5.10	Molar O/N ratio profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.	132
5.11	Composition space coverage of GRI 1.2 randomly generated data which shares the same element mass fractions as the 1D premixed flame EDs data.	137
5.12	Composition space coverage of GRI 3.0 randomly generated data which shares the same element mass fractions as the 1D laminar flamelet data.	137

List of Tables

3.1	Variables which can be altered in 1D flamelet simulations to change the sampled composition space.	78
3.2	Parameters used for random data generation on the 1D laminar flamelet data.	79
3.3	Summary of tested ANN architectures and their respective mean relative RMS error over all predicted species.	81
3.4	Average time taken for reaction source term computations using ANNs and DI, normalised to the time taken for reaction source term computations using DI.	88
4.1	Variables which can be altered in 1D premixed flame simulations to change the sampled composition space.	95
4.2	1D premixed flame simulation information.	96
4.3	Parameters used for random data generation on the 1D premixed flame data.	97
4.4	Maximum change in H_2 mass fraction over the randomly generated data, as well as the boundaries for division into data sets for MMLP training.	98
4.5	Operating conditions for SWB5 and SWB6.	100
4.6	Average time taken for reaction source term computations in SWB6 simulations using ANNs and DI, normalised to the time taken for reaction source term computations using DI.	111
4.7	Time taken for different parts of the simulations of SWB6 using VODE and the MMLPs.	111
5.1	Premixed flame simulation details.	124

5.2	GRI 1.2 training set information and ANN architecture.	125
5.3	USC Mech II training set information and ANN architecture.	125
5.4	Root mean square error in conserved quantities across the reaction step over all reacting points throughout simulations with non-equal diffusivities at $\phi = 1.0$ and GRI 1.2.	126
5.5	Percentage frequency of residual species permutations at different equivalence ratios for 1D premixed flames with NEDs for GRI 1.2. . .	129
5.6	Average time taken to perform one reaction step in GRI 1.2 1D pre- mixed flame simulations with NEDs at $\phi = 1.0$ with each integration method using a single Intel Xeon 6132 2.6 GHz CPU.	130
5.7	Root mean square error in conserved quantities across the reaction step over all reacting points throughout simulations with non-equal diffusivities at $\phi = 1.0$ for USC Mech II.	130
5.8	Percentage frequency of residual species permutations at different equivalence ratios for 1D premixed flames with NEDs for USC Mech II.	133
5.9	Average time taken to perform one reaction step in USC Mech II 1D premixed flame simulations with NEDs at $\phi = 1.0$ with each integration method using a single Intel Xeon 6132 2.6 GHz CPU. . .	134
5.10	Root mean square error between element mass fractions of randomly generated data and canonical combustion problem data for the GRI 1.2 mechanism.	136
5.11	Root mean square error between element mass fractions of randomly generated data and canonical combustion problem data for the GRI 3.0 mechanism.	136
B.1	Elements and species contained in the mechanism and their respective indices.	151
B.2	Mass fractions of species at various stages in Algorithm 1.	152
B.3	Target and predicted element mass fractions, calculated from species in Table B.2.	153
B.4	Initial and final compositions after using Algorithms 1-4.	156

Nomenclature

Roman letters

- A - proportion of the mass of a given species accounted for by a given element
- a - uniform random number
- $\mathbf{b}$ - bias vector
- b - factor determining size of random data changes
- C_d - sub-grid scale micro-mixing constant
- C_P - specific heat capacity at constant pressure
- C_s - Smagorinsky parameter
- $\mathcal{C}$ - index of candidate final residual species
- D - diameter
- $\mathcal{D}$ - diffusivity
- dW - increment of Wiener process
- E - reaction activation energy
- $\mathcal{E}$ - index of element in mechanism
- $\mathcal{F}$ - fine-grained probability density function
- H - enthalpy
- h - specific enthalpy
- $\mathbf{I}$ - identity matrix
- $\mathbf{J}$ - Jacobian matrix
- $\mathcal{J}$ - mass flux
- $\mathcal{J}^{en}$ - energy flux
- K - equilibrium constant
- k^f - elementary forward reaction rate
- k^r - elementary reverse reaction rate
- L - loss function

Le - Lewis number
 M - statistical moment
 m - amount of mass
 M^r - molecular weight
 N - amount of moles
 $\mathbf{n}$ - multi-layer perceptron hidden layer output
 N_C - number of candidate final residual species
 N_{data} - number of data
 N_e - number of elements to be conserved
 N_{el} - number of elements in mechanism
 N_{field} - number of stochastic fields
 N_{hidden} - number of hidden layers
 N_{in} - number of inputs
 N_{react} - number of reactions in mechanism
 $N_{neurons}$ - number of neurons
 N_{out} - number of outputs
 N_s - number of predicted species
 N_{scal} - number of thermochemical scalars
 N_{sp} - number of species in mechanism
 $N_{weights}$ - number of weights
 P - pressure
 $\mathcal{P}$ - probability density function
 Pr - Prandtl number
 $\dot{Q}$ - heat source term
 $\mathcal{Q}$ - indices of possible species from which a residual is selected
 R - universal gas constant
 $\mathbf{r}$ - multi-layer perceptron input
 r - radius
 $\mathcal{R}$ - index of rectifying species
 S - entropy
 $\mathbf{s}$ - multi-layer perceptron target
 $\mathbf{s}^*$ - multi-layer perceptron output
 Sc - Schmidt number
 $\mathcal{S}$ - index of residual species

T - temperature
 t - time
 $\mathcal{T}$ - index of most contributing species
 U - bulk velocity
 u - velocity
 u_c - correction velocity
 V - volume
 $\mathbf{v}^{som}$ - self-organising map neuron
 $\mathbf{W}$ - matrix of weights between multi-layer perceptron layers
 $\mathbf{w}$ - all weights of a multi-layer perceptron
 X - mole fraction
 x - spatial position
 Y - mass fraction
 Z - element mass fraction
 z - mixture fraction

Greek letters

α - index of thermochemical scalar
 β - index of thermochemical scalar
 γ - filter width
 ζ - learning rate
 η - dichotomic random variable
 θ - self-organising map neighbourhood function
 ϑ - reaction temperature exponent
 κ - thermal conductivity
 $\varkappa$ - rate of reaction progress variable
 λ - loss function weight
 μ - viscosity
 ν - stoichiometric coefficient
 ξ - stochastic field variable
 ϖ - strain rate
 ρ - density
 ϱ - reaction pre-exponential factor

σ - Prandtl or Schmidt number
 ς - self-organising map neighbourhood function width
 τ - viscous stress
 v - chemical formula
 Υ - micro-mixing time scale
 ϕ - equivalence ratio
 φ - thermochemical scalar
 χ - scalar dissipation rate
 ψ - thermochemical scalar phase space
 $\dot{\omega}$ - species reaction source term

Acronyms and abbreviations

1D - One dimensional
 act - activation function
 ANN - Artificial neural network
 BOFFIN - Boundary fitted flow integrator
 CCP - Canonical combustion problem
 CFD - Computational fluid dynamics
 CNN - Convolutional neural network
 CPU - Central processing unit
 CSP - Computational singular perturbation
 DCNN - Deconvolutional neural network
 DI - Direct integration
 DLR - German Aerospace Center (Deutsches Zentrum für Luft)
 DNS - Direct numerical simulation
 ED - Equal diffusivity
 FVI - Flame-vortex interactions
 ILDM - Intrinsic low-dimensional manifold
 ISAT - In-situ adaptive tabulation
 LES - Large eddy simulation
 LM - Levenberg-Marquardt
 MLP - Multi-layer perceptron
 MMLP - Multiple multi-layer perceptron

MPI - Message passing interface
MSE - Mean square error
NED - Non-equal diffusivity
ODE - Ordinary differential equation
PDF - Probability density function
RANDGEN - Random data generation method
RANS - Reynolds-averaged Navier-Stokes
RCCE - Rate-controlled constrained equilibrium
RMS - Root mean square
RMSE - Root mean square error
 sgs - sub-grid scale
SOM - Self-organising map
SWB - Cambridge/Sandia swirl burner
VODE - Variable-coefficient ordinary differential equation

Chapter 1

Introduction

1.1 Background and motivation

Anthropogenic emissions of greenhouse gases from the combustion of fossil fuels are contributing to global heating at an unprecedented rate. The 2015 Paris Agreement aims to limit the global temperature rise to 1.5 °C compared to pre-industrial levels in order to avoid the most damaging effects of global heating. The Paris Agreement requires its signatories to explore the development of low-carbon technologies to reduce greenhouse gas emissions across a wide range of economic sectors. The widespread penetration of renewable energy technologies makes the transition of the power generation and surface transport sectors to a low-carbon state technically and economically feasible in the coming decades. However the decarbonisation of the aviation sector will not be as straightforward.

While small scale trials of battery powered aircraft have been successful, the low energy density of current battery technologies compared to aviation fuels such as kerosene precludes their use in large commercial airliners. Using synthetic fuels, manufactured from carbon captured from the atmosphere, has the potential to greatly reduce the net carbon emission of commercial aviation. Currently however, the cost of manufacture of these synthetic fuels means they are not economically viable. The need is then clear for the continued development of combustion equipment for commercial airliners to increase fuel efficiency, for a reduction in carbon dioxide emissions if using conventional fuels or to reduce costs if using synthetic fuels. Such development will rely on computationally cheap and accurate computational fluid dynamics (CFD) simulations of combustion equipment.

There are two main sources of computational cost in combustion simulations. The first source is the range of turbulent scales which are simulated. Direct numer-

ical simulations (DNS) resolve the whole range of temporal and spatial turbulence scales present in a reacting flow, from the integral length scale all the way down to the Kolmogorov microscales. The full resolution of the turbulence means that DNS does not require a model for turbulence or a model for the interaction between turbulence and chemistry. However, in order to resolve all of the turbulent scales, small grid cell sizes are required and so large numbers of cells are used. As the smallest turbulent length scales decrease with increasing Reynolds number, the cells sizes must also decrease, which increases the computational expense of DNS. Typical Reynolds numbers encountered in industrial applications result in infeasible computational demands when using DNS, even when using state-of-the-art supercomputers.

Rather than resolving all of the turbulent scales, large eddy simulation (LES) resolves only the large, low-frequency scales present in the flow. This separation of scales is achieved using a low-pass filter. By not resolving the smaller, high-frequency scales the computational demands are lower, rendering simulations of industrial equipment with complex geometries and high Reynolds numbers feasible. Although the larger scales contain the bulk of the flow energy, the small scales do contribute significantly and so must be modelled. This presents a problem for LES of reacting flows. Important combustion phenomena occur at scales which are well below the resolved range of typical LES. Representing the interaction between the turbulence and the chemistry at the unresolved scales is then a key modelling choice.

One such modelling technique involves the solution of a transport equation for the one-point filtered joint probability density function (PDF) of the relevant scalars involved in the reacting system. Such an approach yields a closed form description of the otherwise unresolved reaction source terms, and indeed allows any otherwise unresolved one-point function to be computed. PDF methods are thus attractive from an LES perspective as they offer a potentially burning regime independent account of the interaction between the turbulence and chemistry at the unresolved scales. Terms requiring two-point information, such as convective fluxes in physical or composition space, remain unresolved and so must be modelled. However, it is supposed that the modelling of these terms is more straightforward than the modelling of the turbulence-chemistry interaction. A comprehensive treatment of PDF methods in reacting flows can be found in the reviews of O'Brien [1], Pope [2] and Dopazo [3]. One complication associated with PDF methods is that the dimensionality of the PDF scales with the number of reacting scalars, and so the PDF transport equation can contain a very large number of independent variables. Therefore numerical solution of the PDF transport equation is usually obtained using stochastic methods.

A commonly used stochastic solution method is based on Lagrangian stochastic particles [2], where a collection of particles are used to represent the scalar PDF,

coupled with an Eulerian approach for the velocity components. An alternative is the stochastic fields method, which is an Eulerian approach first formulated by Valiño [4] with an alternative derivation provided by Sabel'nikov and Soularud [5]. The method introduces an ensemble of stochastic fields which evolve according to stochastic partial differential equations and aim to describe an equivalent stochastic system to the joint PDF transport equation. The stochastic fields method is easily implementable into finite volume CFD codes, while it is easy to compute the required statistical moments. The stochastic fields method has been developed for use in LES by, for example, Jones and Navarro-Martinez [6] and has been applied successfully since to a variety of turbulent flames in different burning regimes [7, 8, 9, 10].

While the LES-PDF method may incur a significantly lower computational cost than DNS, the second source of significant computational cost in turbulent combustion simulations, and indeed the computational bottleneck in such simulations, is the real time computation of the chemical kinetics. The chemical kinetics of a reacting system are modelled using a system of ordinary differential equations (ODEs), where each ODE expresses the instantaneous net formation rate of one of the chemical species present in the considered reaction mechanism. Typical reaction mechanisms for combustion systems with simple fuels consist of tens of species and hundreds of individual reactions. For more complex reaction mechanisms such as those for long carbon chain fuels like kerosene or those for predicting pollutant formation, it is not uncommon to consider hundreds of species and thousands of reactions.

Many combustion modelling methods require this system of ODEs to be numerically integrated at every point in time and space throughout a simulation. Additionally, the wide range of chemical time scales present in such reacting systems often require stiff ODE solvers utilising small step sizes to be used, further complicating the solution procedure. The real time integration of the system of ODEs is required in DNS and simulations using the LES-PDF method. Further, if the numerical solution of the PDF equation is obtained using stochastic methods, the number of evaluations of the chemical kinetics is actually increased, as it must be evaluated for every stochastic particle or stochastic field employed. This computational bottleneck is also present in other modelling approaches including but not limited to: laminar flame computations, conditional moment closure [11, 12], multiple mapping conditioning [13], thickened flame methods [14], the linear eddy model [15] and partially stirred reactors. In simulations utilising such methods it is desirable to alleviate the computational bottleneck by reducing the time required for the integration of the system of ODEs describing the chemical kinetics.

One way of reducing the required time is to introduce a reduced kinetic model, which aims to simplify the chemical kinetics without sacrificing significant accuracy.

Mechanism reduction methods based on timescale analysis, such as the quasi steady state approximation [16], partial equilibrium [16], computational singular perturbation (CSP) [17, 18] and rate-controlled constrained equilibrium (RCCE) [19, 20] have previously been employed for reducing the size of the system of ODEs. However, the integration of the ODEs resulting from the reduced mechanisms (often together with additional algebraic equations) is often still prohibitively expensive for large-scale combustion computations.

Another way of reducing the required computational time utilises a class of methods based on the flamelet model [21, 22], which expresses the full composition in terms of a reduced set of parameters based around mixture fraction and/or progress variable. Such methods offer a very efficient reduction in the number of variables considered but are restricted to certain burning regimes where the assumptions inherent in the flamelet model hold. Other methods which condense the considered reacting system into a lower-dimensional space, such as the intrinsic low-dimensional manifold (ILDM) method [23, 24], flame prolongation of ILDM [25] and manifolds obtained through CSP [26] capture most of the important behaviour contained in a reaction mechanism with a reduced number of variables but the generality of such manifolds difficult to assess.

An alternative approach to reducing the computational time required for the solution of the chemical kinetics is through tabulation. Rather than performing the real time solution of the system of ODEs, a table containing the reaction behaviour of compositions which represent those to be encountered in the application can be assembled ahead of time. By using a cheap and efficient method to perform retrievals from the table during a simulation the computational bottleneck can be alleviated. Tabulation can also be used in conjunction with mechanism reduction and flamelet and manifold methods if desired, to further improve their computational performance.

There are several considerations to be taken into account when using tabulation. The first consideration is the amount of computational memory required to store the table. Each compute node in a distributed memory computer architecture, on which almost all large-scale combustion simulations are run, has its own dedicated memory which is not shared with any other compute nodes. This means that on such computer architectures the amount of memory available to each compute node is limited and tabulation methods should not be overly memory intensive. Another consideration for tabulation methods is the retrieval time, with the maximum computational speed-up obtained if the retrieval time is minimised. The time required for retrievals is linked to how efficiently the table is searched to find the best matching compositions to the one for which the retrieval is required and the interpolation method used. A further consideration is the accuracy of the chosen tabulation method in

the chosen application. The accuracy of the tabulation method is influenced by how well the compositions contained in the table suit the application, the table resolution and the interpolation method used. Additionally, it is desirable that the tabulation method used does not impose more restrictions on the flame regimes which can be simulated than any other stage in the modelling process.

The simplest approach to tabulation is to use a structured look-up table to record compositions and their respective reaction behaviour. It is necessary to store the compositions at a sufficient resolution to capture the highly non-linear behaviour of the chemical kinetics. However, as the table grows exponentially with dimension, storage of look-up tables for cases with more than a handful of variables quickly become infeasible. Limited memory savings can be achieved using local table refinement or a partitioning and then simplification of the composition space [27, 28], however the extension to reaction mechanisms with many species is still unachievable. Exploiting the self-similar behaviour of certain combustion situations [29] can reduce the memory requirements but tables generated in this way are necessarily limited in applicability.

An efficient tabulation method is that of in-situ adaptive tabulation (ISAT) [30]. ISAT reduces memory requirements by only tabulating the compositions encountered during a simulation. An efficient binary tree search method is used to perform fast retrievals from the table. The compositions from ISAT are sampled from the application and so are guaranteed to be well suited to it. However, ISAT introduces a speed penalty compared to pure ahead of time tabulation methods as the real time integration of the system of ODEs must be performed in order to create the table. Additionally, ISAT memory requirements will grow as the simulation progresses and new compositions are added to the table, particularly when accessing unfamiliar regions of the composition space.

An alternative approach to tabulation uses artificial neural networks (ANNs), which are a class of machine learning tools capable of a wide variety of tasks. The following section explains the theoretical foundations of ANNs and gives a brief review of ANN approaches to thermochemistry tabulation.

1.2 Overview of thermochemistry tabulation with artificial neural networks

Machine learning is a sub-field of artificial intelligence which aims to develop computational algorithms which are able to learn and adapt without explicit instruction by extracting patterns from data. The ability to utilise patterns in data rather than

formulating explicit algorithms has led to the proliferation of machine learning in a variety of otherwise difficult tasks such as pattern recognition, natural language processing, clustering, non-linear regression and dimensionality reduction. Machine learning has been applied to various tasks in a combustion modelling context, further details of which can be found in the comprehensive reviews of Ihme et al. [31] and Zhou et al. [32]. Artificial neural networks are a subset of machine learning which are loosely inspired by the biological neural networks that make up brains, and are made up of a collection of basis units called neurons. Any number of connections may exist between neurons. Signals can travel along connections between neurons if they are present. Each neuron typically has a process or function associated with it which collects and modifies incoming signals, although this is not a requirement. The limited number of restrictions on what constitutes an ANN has led to the development of countless different types, each designed to accomplish a given task.

There are two commonly used types of ANNs in combustion modelling. The first type are convolutional neural networks (CNNs), which are adept at extracting features from spatial data. This initially led to the widespread use of CNNs for image analysis tasks, such as image classification or image segmentation. The counterpart of a CNN, the deconvolutional neural network (DCNN) is able to produce spatial data from a set of features. Extracting features from spatial data or spatial data from features is useful in the context of combustion modelling, particularly when filtering operations are performed. Lapeyre et al. [33] trained a CNN-DCNN combination to estimate the unresolved flame surface density field from the filtered progress variable field. Nikolaou et al. [34] applied a DCNN to extract the unfiltered progress variable field from the filtered density and filtered density-progress variable product fields. The unfiltered progress variable field was then used to estimate the filtered progress variable field, allowing the closure of the unresolved scale progress variable variance and the filtered reaction rate. While CNNs have been used for combustion modelling, they are not well suited to thermochemistry tabulation as they operate most effectively when considering a spatial image, which is difficult to obtain when considering reaction source terms only.

The most commonly used type in the context of thermochemistry tabulation is the multi-layer perceptron (MLP). MLPs are a type of feedforward ANN, consisting of layers of neurons connected by weights, which are assigned numerical values. The neurons in an MLP may contain non-linear activation functions, while the weights assign relative importance to the signals between neurons. Inputs arrive at the first, or input, layer of the MLP, and pass through successive 'hidden' layers until reaching the final, or output, layer where the MLP outputs are produced. The values of the MLP weights are determined ahead of a simulation through a non-linear optimisation

process known as training, where representative inputs are presented to the MLP and the error between the target output and the MLP output are used to adjust the weights. MLPs containing non-linear activation functions and consisting of at least a single hidden layer are universal approximators capable of imitating any function. This allows MLPs to replace the real time integration of the system of ODEs by learning the underlying function describing the chemical kinetics. Then, once a composition is presented to the MLP, the rate of reaction or the evolution of the composition due to reaction over a given time step can be estimated.

The tabulation process using ANNs broadly consists of three main steps. The first step is to generate a table containing the representative composition space points for the desired application from which training will occur. The second step is to determine the ANN architecture, which may include some subdivision or simplification of the training set. The third step is to perform the training process, during which the ANN weights of the chosen architecture are adjusted to represent the reaction behaviour of the points contained in the training set. Once the weights are determined the ANNs can be used to replace the reaction mechanism integration in the application of choice.

Tabulation using ANNs has significant advantages over other tabulation methods. It is only the values of the weights, and very limited information about the structure of the ANNs, that need to be stored during a simulation, so memory requirements are both low and fixed. Additionally, retrievals are performed via a small number of matrix multiplications and calculations of the required non-linear activation functions. This is a considerably faster process than integrating the system of ODEs and is also not affected by issues relating to the stiffness of the ODE system.

However, the key issue with ANN tabulation of thermochemistry is accuracy. The overall accuracy of an ANN tabulation approach is determined by two main factors. The key factor relates to how similar the compositions contained in the training data set are to those encountered in the application. ANNs are able to interpolate within the composition space on which they are trained, but much like other tabulation methods, ANNs are not able to accurately extrapolate outside of the data they are trained on. This means that the training set must anticipate the composition space to be covered in the application. It is thus desirable to generate training data which is as general as possible, to increase the chances of the data being similar to the application. This has the added benefit of meaning ANNs trained on such data can be applied to families of similar problems, potentially increasing the speed-up obtained. Generating such data is a challenging issue, particularly when considering mechanisms with many species and so a high dimensional composition space. The second factor depends on how accurately the MLPs are able to imitate the function described by the system of ODEs over the data on which they are

trained. It is necessary to ensure that both factors are considered for a successful ANN tabulation. If the ANNs are well trained on data which is dissimilar to that encountered in the application, then their accuracy in the application will be poor. Similarly, if the ANNs are poorly trained on data which is similar to that encountered in the application then accuracy will also be poor.

Various approaches to generating training data have been used previously. Training data which is generated from the target application or from a simplification of it [35, 36] have good accuracy practically guaranteed, but lose a significant proportion of the potential speed-up due to having to simulate the application in the first place. Additionally, ANNs produced in this way lack generality and will exhibit poor performance in other applications unless they are very similar. For these reasons, the few works which generate training data directly from the application normally explore other aspects of the ANN method. For example, Blasco et al. [37] and Owoyele and Pal [38] assessed the accuracy of new ANN architectures, while Chen et al. [39] were concerned with reducing the memory requirements of ISAT using ANNs. Similarly, the focus of Große and Joos [40, 41] was on investigating whether chemical mechanism reduction could be performed by using ANNs to predict select species only. More general approaches to producing training data are desirable. The simplest and most general approaches sample data across the entire theoretically allowable composition space [42, 43, 44, 45, 46]. However, this method suffers from similar problems to look-up tables and is only feasible if the number of composition space variables is small. A reduction in memory requirements can be achieved using methods such as Latin Hypercube sampling [47], as performed by An et al. [48], however, some notion of the composition space to be covered is required and as the number of dimensions increases beyond tens of chemical species an overwhelming number of samples is needed to obtain adequate resolution.

A widely used approach utilises canonical combustion problems (CCPs) to generate training data. The use of CCPs allows data to be generated for systems with large numbers of reacting variables. Such data occupies the relevant region of composition space which is likely to be accessed in realistic combustion situations, but is sufficiently general that it is not specific to a given application, and ANNs trained on such data can be used for families of similar problems. Although the use of CCPs adds a small computational cost to the ANN training data generation, the CCP simulations are relatively inexpensive compared to large-scale turbulent reacting flow simulations. This remains true even as the dimensionality of the composition space to be sampled from increases.

There are a large number of CCPs which can and have been used to generate training data. For example, Sen and Menon [49] employed DNS of flame-vortex interactions (FVI) to generate ANN training data before applying the resulting MLPs

to premixed syngas/air flame simulations using a cubic domain containing a pair of counter rotating vortices superimposed on isotropic background turbulence. Sen and Menon [50] and Sen et al. [51] employed standalone linear eddy mixing simulations to generate realistic training data, before the application of the resulting MLPs to the LES of syngas/air flames. Chatzopoulos and Rigopoulos [52] proposed a method that sampled data from an ensemble of unsteady laminar flamelets, with an ignition process included to capture igniting states. Subsequently, the training data was partitioned with a type of ANN known as a self-organising map (SOM) [53], which is capable of finding similar regions in multi-dimensional data. After the composition space regions were found, a single MLP was trained to imitate the reaction behaviour of each region. This combined SOM-MLP approach was then applied to Reynolds-averaged Navier-Stokes (RANS) simulations with the transported PDF method for the non-premixed CH_4/H_2 German Aerospace Centre (DLR) flames A and B [54]. Franke et al. [55] extended the approach of Chatzopoulos and Rigopoulos [52] method to include extinguishing flamelets, before using the SOM-MLP method for LES of the non-premixed CH_4/air Sydney L flame [56], which features high levels of local extinction. The method of Franke et al. was also used to generate training data by Readshaw et al. [57], again for LES of the Sydney L flame, and Ding et al. [58] for LESs of the partially-premixed CH_4/air Sandia flames D-F [59]. Laubscher et al. [60] used the eddy dissipation model to generate training data for the LES of Sandia flame D, while a stochastic micromixing system was used by Wan et al. [61] and Nguyen et al. [62] to simulate turbulent syngas oxy-flames. Another CCP used to generate training data are perfectly stirred reactors, as used by Cerri et al. [63] and Haghshenas et al. [64].

A related class of works use MLPs in conjunction with the flamelet model, which as mentioned restricts the burning regimes which can be simulated but reduces the computational time required by tabulating all of the reaction variables using mixture fraction and/or progress variable. Therefore, the primary benefit of using MLPs in such cases is in the greatly reduced memory requirements. Kempf et al. [65] used a single MLP for each predicted flamelet variable to perform LES of a non-premixed piloted CH_4/air jet flame, while Ihme et al. [66, 67] used steady flamelet data to train MLPs before application to the LES of a non-premixed bluff-body swirl-stabilised CH_4/H_2 flame. Emami and Fard [68] used the flamelet model to generate training data for a RANS simulation of a non-premixed turbulent $\text{CH}_4/\text{H}_2/\text{N}_2$ flame, while Owoyele et al. [69] applied MLPs trained on unsteady flamelet model data to flames in internal combustion engines. Hansinger et al. [70] used an MLP with many hidden layers trained on flamelet/progress variable tables in an LES of the Sydney/Sandia piloted jet flame with inhomogeneous inlet [71].

As well as the utilisation of a variety of CCPs to generate training data, there

have also been numerous approaches to increasing ANN accuracy over the training data. In many works, an increase in accuracy has been targeted by splitting the ANN approach into two steps. The first step involves a simplification of the problem with the aim of making the second step less difficult, while the second step is the regression using the MLPs themselves. The simplification of the problem in the first step can be achieved in a variety of ways. Prior knowledge of combustion can be used to divide data based on some notion of reactivity, with Chen et al. [39] splitting data using mixture fraction and temperature windows. Wan et al. [61] split their training data into two parts depending on the mass fraction of oxygen. In both cases, individual MLPs were then trained to each data set. These MLPs are necessarily less general than MLPs trained on the overall data set, as they are specialised to certain parts of the composition space. However, if the regions of composition space used to train the individual MLPs contain compositions with similar reaction dynamics then it may be easier for the individual MLPs to learn these dynamics and so the overall accuracy may increase. The composition space can also be divided using data-driven approaches rather than explicitly using combustion knowledge. Numerous authors [37, 52, 55, 36, 72, 48] have used the SOM for composition space division, finding regions of similarity based purely on the Euclidean distance between points. Nguyen et al. [62] used the k-means algorithm, which is broadly similar to the SOM in that both algorithms use the Euclidean distance between sample points and vector nodes to represent similar regions of the considered multidimensional space. In fact, k-means can be considered as a special case of the SOM, in which all of the neurons are completely independent from each other. In both methods the number of clusters used is chosen prior to training, with separate MLPs trained to each resulting cluster. Using more clusters results in more specialised MLPs, which can result in easier training and therefore higher accuracy if the reaction dynamics within clusters are sufficiently similar.

Alternatively, the training data does not have to be divided and MLPs can be trained over it in its entirety. Using a single large MLP to make predictions on all the species over large regions or the entirety of the composition space is feasible using deep learning based approaches. Deep learning does not have a strict definition, but generally makes use of MLPs with at least four and up to several hundred hidden layers, each one containing a large number of neurons. For example, Wan et al. [61] and Nguyen et al. [62] used a four hidden layer MLP with neurons arranged 512-256-128-64 to predict reaction source terms for a reduced mechanism of eleven species. Another common method is to use a single MLP for each species [73, 65, 67, 69, 57], rather than a single MLP for all of the species. This is possible because although certain species outputs are well correlated across the reaction step [69], the outputs do not directly depend on each other, instead being functions of the input only. The resulting MLPs are then more general than MLPs trained to specific composition

space regions as they have been trained over large sections of the composition space and may be easier to train as it is simpler to focus on the reaction behaviour of a single species at a time.

1.3 Contributions of the thesis

While the aforementioned works have applied ANNs to thermochemistry tabulation, a number of research challenges remain. These challenges are described in this section, along with the contributions of the thesis which aim to tackle these challenges.

The first challenge relates to ensuring that the overall ANN approach remains general. There are two facets to this generality. The first is the generality of the training data. While many CCPs have been used in order to generate compositions representative of the application, little attention has been paid to how the ANN themselves alter the composition space during a simulation. As reacting fluid elements pass through the simulation domain, the ANNs repeatedly compute their evolution due to reaction. There is inevitably some error associated with these predictions. This error can accumulate as a simulation progresses, meaning that predictions are generally more accurate at the inlet to the domain and accuracy deteriorates moving downstream. The effect of this error is also to modify the encountered composition space. This can lead to composition space regions being accessed which were not in the training set and also lead to modifications of properties of the composition space. If the ANNs are not sufficiently robust to their own predictions errors then further errors can accumulate rapidly and lead to significant inaccuracies and in the extreme case instability of the simulation. Therefore, it is desirable to formulate training data generation methods which account for the ANN predictions errors and so increase ANN accuracy by increasing ANN robustness to their own prediction errors. The second part of the generality challenge is to ensure that any simplifications of the training set and the ANN training do not lead to detrimental specialisation. Approaches using the combined SOM-MLP architecture have shown that accuracy is increased on a reserved portion of the training set [37, 52, 55]. However, if the composition space encountered in the application does not share the same distribution as the training data, the specialisation inherent in the SOM-MLP method may be detrimental to the overall ANN accuracy.

The second challenge is the application of ANNs to thermochemistry tabulation for premixed flames. While there are many examples of ANN tabulation for non-premixed flames, there are relatively few examples for premixed flames [73, 49, 50, 51] and no examples of industrially relevant large-scale turbulent premixed flame

simulations. This suggests that choosing a CCP to target the composition space encountered in premixed flames is difficult. Indeed, Chi et al. [74] formulated an on-the-fly ANN training method, which operates in a broadly analogous manner to ISAT, to perform two 2-D turbulent DNS premixed flame simulations, as well as a turbulent 3-D DNS of premixed flame ignition and propagation from a spark. While on-the-fly training avoids the need to anticipate the premixed composition space, the method of Chi et al. requires repeated training at every few hundred iterations of a simulation, which reduces the speed-up benefit of ANN tabulation. It is therefore desirable to formulate a method capable of anticipating a premixed flame composition space for purely ahead of time ANN tabulation.

The third challenge for ANN tabulation is how to ensure physical consistency between the ANN predictions and theoretical conservation laws. ANNs making predictions on the composition change due to reaction or on reaction rates should satisfy the laws of conservation of the mass and the chemical elements, as well as the bounds on the composition space. This issue is important for ANN accuracy, as if the ANNs were able to obey mass and element conservation laws then properties of the composition space in the simulation would not be altered, and the robustness of ANNs to the effects of their own predictions errors would be less important.

The mass and element conservation laws manifest themselves as a series of linear constraints on the ANN outputs. In the context of atmospheric modelling, Beucler et al. [75] proposed a method for guaranteeing that ANN predictions satisfy certain linear constraints by predicting all but one variable and then appending a fixed 'constraint layer' to calculate the final variable such that an arbitrary linear constraint is satisfied. The method of Beucler et al. [75] allows the constraint to be applied during the ANN training, and so leads to both physical consistency and high accuracy. However, fixing the constraint during training also means that the variable which is calculated is also fixed. This is problematic for chemical kinetics as it may be desirable to calculate different species depending on the composition, in order to ensure that the composition remains bounded.

It is therefore more advantageous to construct a method which applies the constraints as a post-processing step, as performed by Bolton and Zanna [76], again in the context of atmospheric modelling. Applying the constraints as a post-processing step allows flexibility in which variables to calculate and so may allow unbounded compositions to be avoided, but also presents the problem of which variables to choose to calculate for each composition. A post-processing step has previously been used to apply constraints in the context of reaction mechanisms, with Wan et al. [61, 77] using an element conservation scheme to ensure that predictions of eleven species in a reduced mechanism satisfied the element conservation laws of the entire mechanism. While the method of Wan et al. [61, 77] was successful in satisfying

the conservation laws, the limited number of species meant that a manual selection and then recalculation of specific variables to satisfy each element was feasible. Formulation of a method which can automatically select which species to calculate to enforce the conservation laws is thus desirable.

Considering the challenges of ANN thermochemistry tabulation, the specific contributions of this thesis are:

- (i) The development of a method for producing training data and ANN architectures which are as general as possible, and are robust to the modifications of the composition space caused by ANN prediction errors themselves.
- (ii) The expansion of the range of applications of ANN thermochemistry tabulation by simulating a turbulent premixed jet flame.
- (iii) The formulation of a method which is capable of ensuring that ANN tabulation is physically consistent with conservation laws and composition space bounds.

The importance of these contributions is that they advance the applicability and accuracy of thermochemistry tabulation with ANNs. This means that ANN tabulated thermochemistry can be applied more reliably to alleviate the computational bottleneck present in simulation methods requiring the real time integration of the chemical kinetics including but not limited to: the LES-PDF method, DNS, laminar flame computations, the conditional moment closure method, the multiple mapping conditioning method, thickened flame methods, the linear eddy model and partially stirred reactors.

1.4 Thesis structure

In Chapter 2 of this thesis, an explanation of the required theoretical background necessary to perform simulations of turbulent flames with ANN tabulated thermochemistry is given. This includes the form composition space data takes and the modelling of reaction mechanisms, as well as the conservation equations necessary for simulating reacting flows. The LES-PDF method used for the turbulent flame simulations in this work and its numerical solution with stochastic fields are also described in Chapter 2, with the consistency of the moments arising from the stochastic fields with those arising from the modelled PDF equation examined. Following this, the individual principles of the SOM and MLPs are detailed, with considerations for their training and usage given. The combined SOM-MLP architecture is explained,

along with the formulation of a random data generation method capable of producing general ANN training data which can be used for turbulent flame simulations.

In Chapter 3 various ANN architectures are assessed in order to investigate the generality of the SOM-MLP combination, before the best performing architecture is applied to the LES-PDF simulation of a non-premixed CH_4 /air turbulent jet flame, Sydney flame L. Chapter 4 extends the methodology used in Chapter 3 to generate data for premixed flames as well as detailing a method which increases prediction accuracy by using multiple MLPs for each species. The updated methodology is then applied to LES-PDF simulations of premixed bluff-body swirl-stabilised CH_4 /air turbulent jet flames. In Chapter 5 a method is formulated which is capable of ensuring physical consistency when using ANNs in thermochemistry tabulation, which can be applied to both predictions and data generation. Finally, in Chapter 6 the main achievements of the thesis are summarised, with directions for future work provided which aim to expand the range of ANN applications further and tackle remaining issues.

Chapter 2

Theoretical background

In this chapter the theoretical background necessary to perform thermochemistry tabulation with artificial neural networks is explained. First, the basic form of composition space data and the foundations of chemical kinetics modelling are detailed. Conservation equations for reacting flows are described before the treatment of these conservation equations in large eddy simulation is explained. The transported probability density function method and the stochastic fields solution procedure which serves as the framework for the simulations of turbulent flames performed in this work are also detailed. The relevant machine learning tools used in these simulations: multi-layer perceptrons and the self-organising map, are outlined and considerations for their use given. Finally, a method of generating training data which is sufficiently general for the application to turbulent flames is described.

2.1 Combustion

2.1.1 Combustion chemistry and reaction mechanisms

2.1.1.1 Basics of the composition space

The composition of a chemical species is described by the amounts of the constituent chemical elements of the species. For example, the chemical composition of methane, CH_4 , is one atom of carbon, C, and four atoms of hydrogen, H. The molar mass of a chemical species is then defined as the sum of the molar masses of its constituent elements. Again using methane, the molecular mass, M^r , is:

$$M_{\text{CH}_4}^r = M_{\text{C}}^r + 4M_{\text{H}}^r. \quad (2.1)$$

2 Theoretical background

The chemical composition of a mixture of N_{sp} chemical species is described by the concentration of each species in the mixture. The concentration of each component can be described in different ways. For example, the mass fraction of a species i in a mixture is defined using:

$$Y_i = \frac{m_i}{m_{tot}}, \quad (2.2)$$

where m_i is the mass of the component and m_{tot} is the total mass of the mixture. Alternatively, the mole fraction of a component in a mixture is defined as:

$$X_i = \frac{N_i}{N_{tot}}, \quad (2.3)$$

where N_i is the number of moles of the component and N_{tot} is the total number of moles of the mixture. The mole and mass fractions are related through the molecular weights of the constituents and the average molecular weights of the mixture, M_{av}^r :

$$Y_i = X_i \frac{M_i^r}{M_{av}^r} = X_i \frac{M_i^r}{\sum_{j=1}^{N_{sp}} X_j M_j^r}. \quad (2.4)$$

By construction, the mass and mole fractions are bounded between zero and unity, while the sum of the mass and mole fractions is unity:

$$\sum_{i=1}^{N_{sp}} Y_i = 1 \quad \text{and} \quad 0 \leq Y_i \leq 1. \quad (2.5)$$

Additionally, as each species is made up of chemical elements the mass fractions of the species are related to the element mass fractions by:

$$\sum_{i=1}^{N_{sp}} A_{ij} Y_i = Z_j, \quad (2.6)$$

where A_{ij} is a matrix containing the proportion of the mass of species i accounted for by element j and Z_j is the mass fraction of element j . Further, the ratios between the element mass fractions may be fixed or have a defined range depending on the amounts of fuel and oxidiser. For example, for a domain consisting only of a hydrocarbon fuel inlet and an air inlet, where all species diffuse at the same rate and neglecting numerical errors, all compositions which contain an amount of C will also contain a corresponding amount of H according to the element ratio in

the fuel, even if the actual amount of C contained in the composition varies due to mixing. This remains true even for reacting compositions, as the elements are not created or destroyed by chemical reactions. If the fuel used in such a scenario is CH_4 , then the ratio of the H element mass fraction to the C element mass fraction in any composition is fixed at:

$$\frac{Z_{\text{H}}}{Z_{\text{C}}} = \frac{1}{3}. \quad (2.7)$$

The actual reaction behaviour of a chemical mixture depends on both its composition and its thermodynamic state. If the mixture is assumed to consist of ideal gases, the ideal gas law can be used:

$$PV = N_{\text{tot}}RT, \quad (2.8)$$

where P , V and T are the mixture pressure, volume and temperature respectively and R is the universal gas constant. A mixture of ideal gases can thus be fully described by its composition and two thermodynamic state variables. The energy contained in the mixture can be represented using the specific enthalpy rather than the temperature, with the specific enthalpy of a given species at a temperature T expressed as:

$$h_i = \int_{T_0}^T C_{P,i} dT + \Delta h_{f,i}^0, \quad (2.9)$$

where $C_{P,i}$ is the heat capacity of the species at constant pressure, and $\Delta h_{f,i}^0$ is the enthalpy of formation of the species at the reference temperature T^0 . The specific enthalpy of the mixture is defined as the sum of the species enthalpies weighted by their mass fractions:

$$h = \sum_{\alpha=1}^{N_{sp}} h_{\alpha} Y_{\alpha} = \int_{T^0}^T \left(\sum_{i=1}^{N_{sp}} C_{P,i} Y_i \right) dT + \sum_{i=1}^{N_{sp}} \Delta h_{f,i}^0 Y_i. \quad (2.10)$$

where the non-linear heat capacities can be estimated using empirical fits. If the pressure is fixed, then a composition is described by:

$$\varphi_{\alpha} = [h, Y_i], \quad (2.11)$$

where φ_{α} contains all of the thermochemical scalars needed to fully describe the

system. The dimensionality of the composition space is therefore strongly influenced by the number of species considered in the mechanism, which is commonly of the order of tens or hundreds.

2.1.1.2 Modelling reaction mechanisms

At constant pressure, the instantaneous net formation rates of the chemical species due to reaction, $\dot{\omega}_i$, are then some function of the current composition:

$$\dot{\omega}_i = \frac{dY_i}{dt} = F(h, Y_i) \quad (2.12)$$

where it should be noted that the specific enthalpy so defined does not change over the reaction step. The true species formation rates are unknown but can be modelled using a reaction mechanism. Reaction mechanisms focus on modelling the reaction behaviour of selected species and likely reaction pathways under a given range of operating conditions. Once such mechanisms are assembled, they are tuned with various numerical and experimental tests in order to fit the observed reaction behavior to a reasonable degree of accuracy. Although the reaction mechanisms themselves are models of the true chemical kinetics, for the purposes of this work the reaction mechanism is treated as the absolute truth. Such reaction mechanisms are made up of an ensemble of elementary reactions. The general form of an elementary reactions which may contain any of the N_{sp} species is given by:

$$\sum_{i=1}^{N_{sp}} \nu_{ij}^f v_i \leftrightarrow \sum_{i=1}^{N_{sp}} \nu_{ij}^r v_i, \quad (2.13)$$

where ν_{ij} are the molar stoichiometric coefficients of species i in reaction j , f and r denote the forward and reverse reactions respectively and v_i is the chemical formula of the species. The instantaneous formation rate of a species is given by the net contribution of all of the N_{react} elementary reactions:

$$\dot{\omega}_i = \sum_{j=1}^{N_{react}} (\nu_{ij}^r - \nu_{ij}^f) \varkappa_j, \quad (2.14)$$

where $\varkappa_j$ is the rate of progress variable for a given reaction, given by the difference between the forward and reverse reaction rates:

$$\kappa_j = k_j^f \prod_{i=1}^{N_{sp}} [X_i]^{\nu_{ij}^f} - \prod_{i=1}^{N_{sp}} k_j^r [X_i]^{\nu_{ij}^r}, \quad (2.15)$$

where k_j^f and k_j^r are respectively the forward and reverse reactions rates. The forward reaction rates are assumed to have an Arrhenius dependence on temperature:

$$k_j^f = \varrho_j T^{\vartheta_j} \exp\left(\frac{-E_j}{RT}\right), \quad (2.16)$$

where ϱ_j is the reaction pre-exponential factor, ϑ_j the temperature exponent of the reaction and E_j the activation energy for the considered reaction. The reverse reaction rates are related to the forward reaction rates through the equilibrium constant, K_j^C :

$$k_j^r = \frac{k_j^f}{K_j^C}, \quad (2.17)$$

with K_j^C given by:

$$K_j^C = \exp\left(\frac{\Delta S_j^0}{R} - \frac{\Delta H_j^0}{RT}\right) \left(\frac{P_{atm}}{RT}\right)^{\sum_{i=1}^{N_{sp}} (\nu_{ij}^r - \nu_{ij}^f)}, \quad (2.18)$$

where H^0 and S^0 are the enthalpy and entropy at standard state, P_{atm} is the atmospheric pressure and in this case Δ refers to the change when passing completely from reactants to products:

$$\frac{\Delta S_j^0}{R} = \sum_{i=1}^{N_{sp}} \left(\nu_{ij}^r - \nu_{ij}^f\right) \frac{S_i^0}{R} \quad (2.19)$$

$$\frac{\Delta H_j^0}{RT} = \sum_{i=1}^{N_{sp}} \left(\nu_{ij}^r - \nu_{ij}^f\right) \frac{H_i^0}{RT}. \quad (2.20)$$

Typical detailed reaction mechanisms for simple fuels such as CH_4 or H_2 consist of tens of species and hundreds of elementary reactions, while reaction mechanisms for more complex fuels such as bio-fuels or kerosene may contain hundreds of species and thousands of elementary reactions. As Equation 2.14 shows, the instantaneous formation rate of each species contains the contributions of all of the elementary

reactions the species participates in. The ordinary differential equation (ODE) governing the evolution of each species is thus complex, with the system of ODEs describing the behaviour of all of the species even more so. This system of ODEs along with an initial composition then defines a standard initial value problem. Finding the composition after a given time step Δt involves the numerical integration of the system of ODEs contained in the reaction mechanism:

$$Y_i(t + \Delta t) = Y_i(t) + \int_t^{t+\Delta t} \frac{dY_i}{d\tau} d\tau. \quad (2.21)$$

The typical range of chemical time scales contained in combustion systems means that stiff ODE solvers must be used, with the integration dominated by the smallest step sizes to ensure numerical stability. The numerical integration of the system of ODEs can be instead represented as a mapping:

$$[h(t), Y_i(t)] \mapsto [h(t), Y_i(t + \Delta t)], \quad (2.22)$$

where the mapping function is described by the reaction mechanism. Alternatively, this mapping can be represented using:

$$[h(t), Y_i(t)] \mapsto [h(t), \Delta Y_i], \quad (2.23)$$

where $\Delta Y_i = Y_i(t + \Delta t) - Y_i(t)$. It should be noted that the choice of integration method will also influence the mapping to a small degree and so for numerical consistency the ODE solver used to perform the integration of the training data should be the same as the one used in the CFD solver.

2.1.2 Conservation equations

The conservation laws for mass, momentum and energy can be expressed as a set of coupled partial differential equations for a given fluid. The continuity equation for mass is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \quad (2.24)$$

where ρ is the density, t is time and x_i and u_i are the spatial coordinate and velocity respectively in the i -direction. For a Newtonian fluid and neglecting body forces, the incompressible Navier-Stokes equations are:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2.25)$$

where P is the pressure and τ_{ij} is the viscous stress tensor. Assuming that the bulk viscosity is zero and decomposing the resulting stress tensor into its deviatoric and isotropic parts, τ_{ij} is given by:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) + \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, \quad (2.26)$$

where δ_{ij} is the Kronecker delta. The isotropic part of τ_{ij} is absorbed into the pressure, giving the conservation of momentum equations as:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \right]. \quad (2.27)$$

The conservation equations for the chemical species in a mixture of ideal gases are:

$$\frac{\partial \rho Y_\alpha}{\partial t} + \frac{\partial \rho u_i Y_\alpha}{\partial x_i} + \frac{\partial \mathcal{J}_{\alpha,i}}{\partial x_i} = \rho \dot{\omega}_\alpha, \quad (2.28)$$

where Y_α is the species mass fraction and $\alpha = 1, 2, \dots, N_{sp}$, $\dot{\omega}_\alpha$ is the reaction source term for the given species and $\mathcal{J}_{\alpha,i}$ is the diffusive flux for the species α in the i direction. Neglecting Soret effects and using Fick's Law the diffusive flux, $\mathcal{J}_{\alpha,i}$, is given by:

$$\mathcal{J}_{\alpha,i} = -\rho \mathcal{D}_\alpha \frac{\partial Y_\alpha}{\partial x_i}, \quad (2.29)$$

where $\mathcal{D}_\alpha$ is the equivalent binary diffusion coefficient of the species into the rest of the mixture. Assuming all species diffusivities are equal the species conservation equations become:

$$\frac{\partial \rho Y_\alpha}{\partial t} + \frac{\partial \rho u_i Y_\alpha}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\rho \mathcal{D} \frac{\partial Y_\alpha}{\partial x_i} \right) + \rho \dot{\omega}_\alpha, \quad (2.30)$$

where $\mathcal{D}$ is the mixture averaged diffusivity. Again neglecting body forces, the conservation equation for the enthalpy is:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{\partial P}{\partial t} + u_i \frac{\partial P}{\partial x_i} - \frac{\partial \mathcal{J}_i^{en}}{\partial x_i} + \tau_{ij} \frac{\partial u_j}{\partial x_i} + \dot{Q}, \quad (2.31)$$

2 Theoretical background

where $\mathcal{J}_i^{en}$ is the energy flux and $\dot{Q}$ the contribution of an external heat source, for example due to a spark or radiative flux. Assuming that Dufour effects are negligible, $\mathcal{J}_i^{en}$ can be expressed as:

$$\mathcal{J}_i^{en} = -\kappa \frac{\partial T}{\partial x_i} + \sum_{\alpha=1}^{N_{sp}} h_{\alpha} \mathcal{J}_{\alpha,i}, \quad (2.32)$$

where κ is the thermal conductivity. Assuming a low Mach number flow, the pressure terms and the viscous heating term are neglected. Additionally neglecting the heat source term, the enthalpy conservation equation is:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = -\frac{\partial \mathcal{J}_i^{en}}{\partial x_i}. \quad (2.33)$$

The enthalpy conservation equation can be simplified. The ratio of momentum diffusivity to thermal diffusivity can be expressed using the non-dimensional Prandtl number: $Pr = \mu C_P / \kappa$. Similarly, the ratio of momentum diffusivity to mass diffusivity is given by the non-dimensional Schmidt number: $Sc = \mu / \rho \mathcal{D}$. The Lewis number is then the ratio of mass diffusivity to thermal diffusivity and is given by: $Le = Pr / Sc$. Under the assumption of unity Lewis number, it can be shown [78] that $\mathcal{J}_i^{en}$ can be rearranged to:

$$\mathcal{J}_i^{en} = \frac{\kappa}{C_P} \frac{\partial h}{\partial x_i}, \quad (2.34)$$

and the conservation equation for the enthalpy becomes:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\kappa}{C_P} \frac{\partial h}{\partial x_i} \right). \quad (2.35)$$

It can be seen that under the aforementioned assumptions Equations 2.30 and 2.35 are analogous and so can be expressed in terms of a general reactive scalar φ_{α} , with $\alpha = 1, 2, \dots, N_{scal}$, from here on, where $N_{scal} = N_{sp} + 1$ which is the number of scalars required to describe the reacting system. The conservation equations for the reactive scalars are then:

$$\frac{\partial \rho \varphi_{\alpha}}{\partial t} + \frac{\partial \rho u_i \varphi_{\alpha}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu}{\sigma} \frac{\partial \varphi_{\alpha}}{\partial x_i} \right) + \rho \dot{\omega}_{\alpha}(\varphi_{\alpha}), \quad (2.36)$$

where σ is the Schmidt or Prandtl number, which are interchangeable. The source term for the enthalpy, $\dot{\omega}_{N_{scal}+1}$, is zero. The conservation equations for mass, momen-

tum and the reactive scalars serve as the basis for the LES-PDF method, described in the following section.

2.2 Large eddy simulations of combustion with the transported probability density function method and stochastic fields

To perform large scale turbulent combustion simulations with detailed chemistry at modest computational cost, large eddy simulation can be used. A separation of scales in LES is achieved through filtering using a low-pass filter. Only the large-scale, low frequency eddy motions are resolved, while the small-scale, high frequency eddy motions are unresolved and require modelling. For reacting flows, turbulent fluctuations at the unresolved small scales strongly influence the filtered chemical reaction source terms. The modelling approach for the turbulence-chemistry interaction is then a key step for producing accurate simulations. Transported probability density function methods allow for a potentially burning regime independent description of this interaction. It is too costly to calculate the transported PDF directly, so its numerical solution is obtained using stochastic methods. These stochastic methods introduce the requirement to perform more reaction source term computations, accentuating the need for reaction source term speed-up. The LES-PDF method so described is presented this chapter, along with the method of stochastic fields. Additionally, analysis is performed on the consistency of the moments arising from the stochastic system with those arising from the modelled PDF equation.

2.2.1 Filtered equations of fluid motion

The unfiltered conservation equations for the mass, momentum and reactive scalars from the previous section are summarised as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \quad (2.37)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2.38)$$

$$\frac{\partial \rho \varphi_\alpha}{\partial t} + \frac{\partial \rho u_i \varphi_\alpha}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\mu}{\sigma} \frac{\partial \varphi_\alpha}{\partial x_i} \right) + \rho \dot{\omega}_\alpha. \quad (2.39)$$

To perform large eddy simulation, these conservation equations are filtered. The

2 Theoretical background

spatially filtered version of a general function $f(\mathbf{x}, t)$ is defined as the convolution of the function with a filter function G over the domain Ω :

$$\bar{f}(\mathbf{x}, t) = \int_{\Omega} f(\mathbf{x}', t) G(\mathbf{x} - \mathbf{x}', \gamma(\mathbf{x})) d\mathbf{x}' \quad (2.40)$$

where $\bar{f}(\mathbf{x}, t)$ is the filtered function and γ is the characteristic width of the filter function. The filter function must be positive definite in order to ensure the boundedness of the filtered scalars and to maintain the correct sign of reaction source terms. The width of the filter may vary with spatial position. This means that in general the filter does not commute with the spatial derivative. Here however, the filter width is assumed to vary smoothly such that the filter function commutes with the spatial derivative with negligible round off error. The specific filter function used is a box filter, defined by:

$$G(\mathbf{x} - \mathbf{x}') = \begin{cases} \frac{1}{\gamma^3} & \text{if } |\mathbf{x} - \mathbf{x}'| < \frac{\gamma}{2} \\ 0 & \text{otherwise} \end{cases} \quad (2.41)$$

where γ is taken as the cube root of the local cell volume. The significant density variations found in reacting flows are accounted for through the use of Favre, or density weighted, filtering. The Favre filtered version of a general function is defined as:

$$\tilde{f}(\mathbf{x}, t) = \int_{\Omega} \rho f(\mathbf{x}', t) G(\mathbf{x} - \mathbf{x}', \gamma(\mathbf{x})) d\mathbf{x}', \quad (2.42)$$

where $\tilde{f}(\mathbf{x}, t)$ is the Favre filtered function. Favre filtered functions are related to spatially filtered ones by:

$$\tilde{f}(\mathbf{x}, t) = \frac{\overline{\rho f}}{\bar{\rho}}. \quad (2.43)$$

Application of Favre filtering to the mass and momentum conservation equations gives:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i}{\partial x_i} = 0 \quad (2.44)$$

and

$$\frac{\partial \bar{\rho} \tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{u}_j}{\partial x_j} = -\frac{\partial \bar{P}}{\partial x_i} + \frac{\partial \bar{\tau}_{ij}}{\partial x_j} \quad (2.45)$$

respectively. The filtered viscous stress tensor $\bar{\tau}_{ij}$ is approximated as:

$$\bar{\tau}_{ij} \approx \bar{\mu} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right), \quad (2.46)$$

while the sub-grid scale shear stress tensor τ_{ij}^{sgs} is introduced as:

$$\tau_{ij}^{sgs} = \bar{\rho} (\tilde{u}_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j), \quad (2.47)$$

and the filtered conservation of momentum equation is then:

$$\frac{\partial \bar{\rho} \tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{u}_j}{\partial x_j} = -\frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\bar{\mu} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right) \right] - \frac{\partial \tau_{ij}^{sgs}}{\partial x_j}, \quad (2.48)$$

where τ_{ij}^{sgs} requires modelling. Closure for τ_{ij}^{sgs} is obtained using a gradient approximation, with the deviatoric part of the sgs stress tensor τ_{ij}^{sgs} given by:

$$\tau_{ij}^{sgs} = \mu_{sgs} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right), \quad (2.49)$$

while the isotropic part is absorbed into the pressure term. The sgs viscosity, μ_{sgs} , is determined using the Smagorinsky model [79]:

$$\mu_{sgs} = \bar{\rho} (C_s \gamma)^2 \left\| \frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \right\|, \quad (2.50)$$

where C_s is the Smagorinsky parameter and $\|\cdot\|$ is the Euclidean norm. The Smagorinsky parameter depends on the local conditions of the flow and is obtained using the dynamic model of Piomelli and Liu [80]. The filtered conservation equations for the reactive scalars are:

$$\frac{\partial \bar{\rho} \tilde{\varphi}_\alpha}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{\varphi}_\alpha}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\bar{\mu}}{\sigma} \frac{\partial \tilde{\varphi}_\alpha}{\partial x_i} \right) + \bar{\rho} \tilde{\omega}_\alpha. \quad (2.51)$$

Introducing the sub-grid scale scalar fluxes $\mathcal{J}_{\alpha,i}^{sgs}$ as:

$$\mathcal{J}_{\alpha,i}^{sgs} = \bar{\rho}(\widetilde{u_i \varphi_\alpha} - \widetilde{u_i} \widetilde{\varphi_\alpha}), \quad (2.52)$$

the filtered conservation equations for the reactive scalars become:

$$\frac{\partial \bar{\rho} \widetilde{\varphi}_\alpha}{\partial t} + \frac{\partial \bar{\rho} \widetilde{u_i \varphi}_\alpha}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\bar{\mu}}{\sigma} \frac{\partial \widetilde{\varphi}_\alpha}{\partial x_i} \right) - \frac{\partial \mathcal{J}_{\alpha,i}^{sgs}}{\partial x_i} + \overline{\rho \dot{\omega}_\alpha} \quad (2.53)$$

where $\mathcal{J}_{\alpha,i}^{sgs}$ requires modelling. $\mathcal{J}_{\alpha,i}^{sgs}$ is closed using a gradient approximation and is given by:

$$\mathcal{J}_{i,\alpha}^{sgs} = -\frac{\mu_{sgs}}{\sigma_{sgs}} \frac{\partial \widetilde{\varphi}_\alpha}{\partial x_i}, \quad (2.54)$$

where σ_{sgs} is the sgs Schmidt or Prandtl number, assigned a constant value of 0.7 in simulations. The filtered species reaction rates, $\overline{\rho \dot{\omega}_\alpha}$, remain unclosed despite their significance.

2.2.2 The transported probability density function method and solution with stochastic fields

The difficulty of evaluating the filtered species formation rates, $\overline{\rho \dot{\omega}_\alpha}$, means that Equation 2.53 is not solved directly. Instead, a filtered joint one-point one-time sgs PDF for the reactive scalars, $\widetilde{\mathcal{P}}_{sgs}$, is introduced. The one-point one-time fine-grained PDF, $\mathcal{F}$, for one of the reactive scalars, φ , is given by:

$$\mathcal{F}(\psi; \mathbf{x}, t) = \delta(\psi - \varphi(\mathbf{x}, t)), \quad (2.55)$$

where δ is the Dirac delta and ψ is the phase space of the chosen scalar. The joint one-point one-time fine-grained PDF, $\mathcal{F}_{joint}$, of all of the reactive scalars is then obtained as the product of the individual fine-grained PDFs for each scalar:

$$\mathcal{F}_{joint}(\psi_\alpha; \mathbf{x}, t) = \prod_{\alpha=1}^{N_{scal}} \delta(\psi_\alpha - \varphi_\alpha(\mathbf{x}, t)), \quad (2.56)$$

where ψ_α is the phase space of the reactive scalars, φ_α , also referred to as the composition space. The filtered joint sgs PDF is then:

$$\tilde{\mathcal{P}}_{sgs} = \frac{1}{\bar{\rho}} \int_{\Omega} \rho \mathcal{F}_{joint} G(\mathbf{x} - \mathbf{x}', \gamma(\mathbf{x})) d\mathbf{x}'. \quad (2.57)$$

Physically, $\tilde{\mathcal{P}}_{sgs}$ describes the probability of finding a composition $\psi_{\alpha} \leq \varphi_{\alpha} \leq \psi_{\alpha} + d\psi_{\alpha}$ within the filter volume at a given point in space and time. A transport equation for $\tilde{\mathcal{P}}_{sgs}$ can be derived through standard methods, for example by following Gao and O'Brien [81], Colucci et al. [82] or Jaber et al. [83]:

$$\begin{aligned} \frac{\partial \rho \mathcal{F}_{joint}}{\partial t} + \frac{\partial \rho u_i \mathcal{F}_{joint}}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\frac{\mu}{\sigma} \frac{\partial \mathcal{F}_{joint}}{\partial x_i} \right) = \\ - \sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_{\alpha} \partial \psi_{\beta}} \left(\mathcal{F}_{joint} \frac{\mu}{\sigma} \frac{\partial \varphi_{\alpha}}{\partial x_i} \frac{\partial \varphi_{\beta}}{\partial x_i} \right) - \sum_{\alpha=1}^{N_{scal}} \frac{\partial \mathcal{F}_{joint}}{\partial \psi_{\alpha}} \rho \dot{\omega}_{\alpha}. \end{aligned} \quad (2.58)$$

After the application of the filtering, the transport equation for $\tilde{\mathcal{P}}_{sgs}$ is:

$$\begin{aligned} \underbrace{\frac{\partial \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial t}}_{\text{unsteady term}} + \underbrace{\frac{\partial \bar{\rho}(u_i | \widetilde{\boldsymbol{\varphi}} = \boldsymbol{\psi}) \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial x_i}}_{\text{sgs convection term}} - \underbrace{\frac{\partial}{\partial x_i} \left(\frac{\mu}{\sigma} \frac{\partial \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial x_i} \right)}_{\text{diffusion term}} = \\ - \underbrace{\sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_{\alpha} \partial \psi_{\beta}} \left(\tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) \frac{\mu}{\sigma} \left(\frac{\partial \varphi_{\alpha}}{\partial x_i} \frac{\partial \varphi_{\beta}}{\partial x_i} | \widetilde{\boldsymbol{\varphi}} = \boldsymbol{\psi} \right) \right)}_{\text{micro-mixing term}} \\ - \underbrace{\sum_{\alpha=1}^{N_{scal}} \frac{\partial}{\partial \psi_{\alpha}} \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) \dot{\omega}_{\alpha}(\boldsymbol{\psi})}_{\text{reaction source term}}, \end{aligned} \quad (2.59)$$

where it has been assumed that $\bar{\rho} = \rho(\boldsymbol{\psi})$. The sgs convection term is split into a resolved and unresolved part:

$$\bar{\rho}(u_i | \widetilde{\boldsymbol{\varphi}} = \boldsymbol{\psi}) \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) = \bar{\rho} \tilde{u}_i \tilde{\mathcal{P}}_{sgs} + \bar{\rho} \left((u_i | \widetilde{\boldsymbol{\varphi}} = \boldsymbol{\psi}) - \tilde{u}_i \right) \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}), \quad (2.60)$$

with the unresolved part modelled using a gradient approximation:

2 Theoretical background

$$\bar{\rho} \left((u_i | \widetilde{\varphi} = \psi) - \tilde{u}_i \right) \tilde{\mathcal{P}}_{sgs}(\psi) = - \frac{\mu_{sgs}}{\sigma_{sgs}} \frac{\partial \tilde{\mathcal{P}}_{sgs}(\psi)}{\partial x_i}. \quad (2.61)$$

To ensure that $\tilde{\mathcal{P}}_{sgs}$ behaves correctly in the laminar limit, following Brauner et al. [84], the micro-mixing term is split into resolved and unresolved parts in a similar manner to the sgs convection term:

$$\begin{aligned} \sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \left(\tilde{\mathcal{P}}_{sgs}(\psi) \frac{\mu}{\sigma} \left(\frac{\partial \varphi_\alpha}{\partial x_i} \frac{\partial \varphi_\beta}{\partial x_i} | \widetilde{\varphi} = \psi \right) \right) = \\ \sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \tilde{\mathcal{P}}_{sgs}(\psi) \frac{\mu}{\sigma} \left(\frac{\partial \tilde{\varphi}_\alpha}{\partial x_i} \frac{\partial \tilde{\varphi}_\beta}{\partial x_i} \right) + M_{sgs}, \end{aligned} \quad (2.62)$$

where M_{sgs} is given by:

$$\begin{aligned} M_{sgs} = \\ \sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \left(\tilde{\mathcal{P}}_{sgs}(\psi) \frac{\mu}{\sigma} \left[\left(\frac{\partial \varphi_\alpha}{\partial x_i} \frac{\partial \varphi_\beta}{\partial x_i} | \widetilde{\varphi} = \psi \right) - \left(\frac{\partial \tilde{\varphi}_\alpha}{\partial x_i} \frac{\partial \tilde{\varphi}_\beta}{\partial x_i} \right) \right] \right). \end{aligned} \quad (2.63)$$

M_{sgs} is modelled using the Linear Mean Square Estimation [85] or Inter Exchange with the Mean model [86]:

$$M_{sgs} = \frac{C_d}{\Upsilon_{sgs}} \sum_{\alpha=1}^{N_{scal}} \frac{\partial}{\partial \psi_\alpha} \left[(\psi_\alpha - \tilde{\varphi}_\alpha) \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\psi) \right], \quad (2.64)$$

where C_d is the sgs mixing constant, assigned the constant value of 2.0, and Υ_{sgs} is the micro-mixing timescale, which also needs to behave correctly in the laminar limit. Υ_{sgs} is therefore given by:

$$\Upsilon_{sgs} = \frac{\bar{\rho} \gamma^2}{\mu_{sgs}}. \quad (2.65)$$

Collecting all of the modelled terms, the final transport equation for $\tilde{\mathcal{P}}_{sgs}$ is:

$$\begin{aligned}
& \frac{\partial \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial t} + \frac{\partial \bar{\rho} \tilde{u}_i \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{sgs}}{\sigma_{sgs}} \right) \frac{\partial \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi})}{\partial x_i} \right) = \\
& - \sum_{\alpha=1}^{N_{scal}} \sum_{\beta=1}^{N_{scal}} \frac{\partial^2}{\partial \psi_\alpha \partial \psi_\beta} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) \frac{\mu}{\sigma} \left(\frac{\partial \tilde{\varphi}_\alpha}{\partial x_i} \frac{\partial \tilde{\varphi}_\beta}{\partial x_i} \right) \\
& - \frac{C_d}{\Upsilon_{sgs}} \sum_{\alpha=1}^{N_{scal}} \frac{\partial}{\partial \psi_\alpha} \left[(\psi_\alpha - \tilde{\varphi}_\alpha) \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) \right] \\
& - \sum_{\alpha=1}^{N_{scal}} \frac{\partial}{\partial \psi_\alpha} \bar{\rho} \tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}) \dot{\omega}_\alpha(\boldsymbol{\psi}).
\end{aligned} \tag{2.66}$$

where it can be seen that the reaction source terms appear in closed form and do not require modelling. The high dimensionality of the phase space of $\tilde{\mathcal{P}}_{sgs}$ precludes the direct numerical solution of Equation 2.66. Instead, a stochastic system is introduced which aims to share the same statistical moments as the $\tilde{\mathcal{P}}_{sgs}$ transport equation. To perform this, $\tilde{\mathcal{P}}_{sgs}$ is represented using $n = 1, 2, \dots, N_{field}$ stochastic fields for each of the $\alpha = 1, 2, \dots, N_{scal}$ reactive scalars:

$$\tilde{\mathcal{P}}_{sgs}(\boldsymbol{\psi}; \mathbf{x}, t) = \frac{1}{N_{field}} \sum_{n=1}^{N_{field}} \frac{1}{\bar{\rho}} \int_{\Omega} \rho G(\mathbf{x} - \mathbf{x}', \gamma(\mathbf{x})) \prod_{\alpha=1}^{N_{scal}} \delta(\psi_\alpha - \xi_\alpha^n(\mathbf{x}, t)) d\mathbf{x}'. \tag{2.67}$$

Using standard methods, for example as in Gardiner [87], and here following Valiño [4] or Sabel'nikov and Soulard [5] the evolution of each stochastic field can be described using the following stochastic partial differential equation:

$$\begin{aligned}
& \bar{\rho} d\xi_\alpha^n + \bar{\rho} \tilde{u}_i \frac{\partial \xi_\alpha^n}{\partial x_i} dt - \frac{\partial}{\partial x_i} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{sgs}}{\sigma_{sgs}} \right) \frac{\partial \xi_\alpha^n}{\partial x_i} \right] dt = \\
& \sqrt{2\bar{\rho} \frac{\mu_{sgs}}{\sigma_{sgs}}} \frac{\partial \xi_\alpha^n}{\partial x_i} dW_i^n - \frac{\bar{\rho} C_d}{2\Upsilon_{sgs}} (\xi_\alpha^n - \tilde{\varphi}_\alpha) dt + \bar{\rho} \dot{\omega}_\alpha^n(\xi_\alpha^n) dt,
\end{aligned} \tag{2.68}$$

where the Itô interpretation of the stochastic integral has been used. The increments of the Wiener process, dW_i^n , are different for each field but independent of the spatial location $\mathbf{x}$ and are approximated by $\eta_i^n \sqrt{dt}$ where η_i^n is a $\{-1, 1\}$ dichotomic random vector. The solutions of Equation 2.68 are continuous and differentiable in space, and are smooth on the scale of the filter width. The fields are continuous but not

differentiable in time, hence the expression of Equation 2.68 in terms of the time increment. The filtered reactive scalars are easily obtained by averaging over the stochastic fields:

$$\tilde{\varphi}_\alpha = \frac{1}{N_{field}} \sum_{n=1}^{N_{field}} \xi_\alpha^n \quad (2.69)$$

The filtered equations for continuity, momentum and the stochastic fields are solved numerically using an in-house, block-structured, boundary conforming coordinate LES code known as BOFFIN-LES [88]. BOFFIN-LES utilises a second-order-accurate finite volume method, based on an implicit low-Mach-number pressure based formulation with co-located variable storage and Rhie and Chow [89] pressure smoothing. Second order central differencing schemes are used for all spatial derivatives, with the exceptions of the convective term in the momentum equation where an energy conserving discretisation scheme is used, and the convective term in the stochastic fields equation where a total variation diminishing scheme is used to avoid unphysical overshoots and undershoots. The method of fractional steps is employed for the discretisation of the stochastic fields, where the first step contains the convection-diffusion terms, the second the micro-mixing and finally the reaction source terms. Implicit Crank-Nicholson schemes are used for the convection-diffusion and micromixing terms, while a Euler-Maruyama scheme is used for the stochastic term. The code is fully parallelised using domain decomposition and message passage interface (MPI) routines for the exchange of information across sub-domain boundaries. Further information on the spatial and temporal discretisation schemes used in BOFFIN-LES can be found in Prasad [90] and Jones et al. [91], while a general review of CFD numerical techniques and discretisation schemes can be found in Ferziger and Peric [92].

The number of stochastic fields used is chosen ahead of time. While theoretically an infinite number of fields is required to obtain moments of any order, in practice reasonable accuracy for lower order moments in turbulent combustion simulations can be achieved with a modest number of fields [93]. For example, it is common to employ eight stochastic fields [7, 6, 8, 9, 10] to obtain first and second-order moments. Whatever the number of fields, it is clear from Equation 2.68 that each of the stochastic fields has its own reaction source term, and thus the number of chemical reaction solver calls increases linearly with the number of fields. The solution of the reaction source terms in each stochastic field typically occupies the majority of the computational time required for an LES-PDF simulation using the stochastic fields method. This amplifies the need for speed-up methods for reaction source term computations when using stochastic methods. Additionally it is desirable that the

speed-up method used is capable of being burning regime independent, or at least can be tailored for the specific burning regime encountered in a given test case, as imposing restrictions on the burning regime via the reaction source term speed-up method would not allow the full potential of stochastic methods to be utilised.

Tabulating thermochemistry with ANNs is therefore ideally suited for stochastic methods, and can be easily incorporated into fractional step simulations of Equation 2.68 by replacing calls to the conventional reaction solver with calls to the ANNs, making sure that any temperature or flammability bounds imposed on the ANN training data set are also imposed in the application. The same ANNs can be used for the reaction source term computations for an arbitrary number of fields, so tabulation memory requirements do not increase with the number of fields.

2.2.3 Consistency of the stochastic fields

The work in this section and in Appendix A has been published in Readshaw et al. [57].

The solutions to Equation 2.68 do not have a physical meaning, but instead are supposed to form an equivalent stochastic system to the one found by the solution to 2.66. However, Wang et al. [94] suggested that this is not the case. The equivalence of the moments arising from Equation 2.66 to Equation 2.68 can be examined by comparing the conservation equations describing the one-point moments arising from the two approaches. For simplicity, the PDF of a single scalar variable, $\varphi(\mathbf{x}, t)$ will be considered with the density assumed constant. The equation for the moments from the stochastic fields can be obtained using standard methods, with a full derivation given in Appendix A. The equation for the n^{th} moments from the stochastic fields are:

$$\begin{aligned} \rho \frac{\partial M_n}{\partial t} + \rho \bar{u}_i \frac{\partial M_n}{\partial x_i} + n(n-1) \frac{\mu}{\sigma} \overline{\varphi^{n-2} \left(\frac{\partial \varphi}{\partial x_i} \right)^2} = \\ \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{sgs}}{\sigma_{sgs}} \right) \frac{\partial M_n}{\partial x_j} \right] - n \rho \frac{C_d}{\Upsilon_{sgs}} (M_n - M_{n-1} \bar{\varphi}) + n \rho \overline{\dot{\omega} \varphi^{n-1}}. \end{aligned} \quad (2.70)$$

The equation for the moments arising from the modelled PDF equation can be readily calculated, with the full derivation also given in Appendix A. The following equation for the n^{th} moment from the modelled PDF equation results:

$$\begin{aligned} \rho \frac{\partial M_n}{\partial t} + \rho \bar{u}_i \frac{\partial M_n}{\partial x_i} + n(n-1)M_{n-2} \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x_i} \right)^2 = \\ \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{sgs}}{\sigma_{sgs}} \right) \frac{\partial M_n}{\partial x_j} \right] - n\rho \frac{C_d}{\Upsilon_{sgs}} (M_n - M_{n-1} \bar{\varphi}) + n\rho \overline{\dot{\omega} \psi^{n-1}}. \end{aligned} \quad (2.71)$$

The equations for the moments calculated in Equations 2.70 and 2.71 differ by the term:

$$n(n-1) \frac{\mu}{\sigma} \left(\overline{\varphi^{n-2} \left(\frac{\partial \varphi}{\partial x_i} \right)^2} - \overline{\varphi^{n-2}} \left(\frac{\partial \bar{\varphi}}{\partial x_i} \right)^2 \right). \quad (2.72)$$

The zeroth and first moments are therefore identical. For higher order moments, this term is negligible if $\mu_{sgs} \gg \mu$ and is zero if the flow is laminar and there are no sgs variations. In the majority of LES computations the former situation will prevail over the majority of the solution domain but the flow is quite likely to be laminar or quasi-laminar in small and sometimes important regions. In these latter circumstances the direct viscous terms become dominant and so the moments arising from the stochastic fields and the modelled PDF equation can be considered to be practically equal. The discrepancy highlighted here differs from the analysis of Gallot-Lavallée [95], where an averaging was erroneously applied across a product.

2.3 Machine learning

The tabulation of reaction mechanisms can be thought of as a non-linear regression problem, where the function to be approximated is the one described by the chemical mechanism. This non-linear regression can be performed effectively using multi-layer perceptrons. There are a number of parameters which must be carefully chosen to maximise MLP accuracy, which include the training algorithm and the value of any associated parameters, the MLP architecture, which determines the overall number of weights, and the types of activation function used. Considerations for the selection of these parameters are given in Section 2.3.1. Approaches to chemical mechanism tabulation also commonly use clustering of the composition space, where the aim is to divide the composition space into regions with similar behaviour thus simplifying the regression step. The self-organising map (SOM) has been previously used for performing the composition space division, and is described in Section 2.3.2.

2.3.1 Multi-layer perceptrons

Multi-layer perceptrons are a class of feedforward artificial neural networks. Feedforward in this context means that information moves only in one direction from the start to the end of the network without any cycles or loops. Figure 2.1 shows a general MLP architecture, consisting of neurons (circles) arranged in layers which are joined together using weights (lines). The first (leftmost) layer is the input layer, which takes the network inputs, while the last (rightmost) layer is the output layer and gives the network outputs. The layers between the input and output layers are known as hidden layers. Each weight has a corresponding numerical value, while each neuron, apart from the neurons on the input layer, has an associated activation function. The neurons marked b are the bias neurons, which do not take an input but instead are given a constant input value of $+1$.

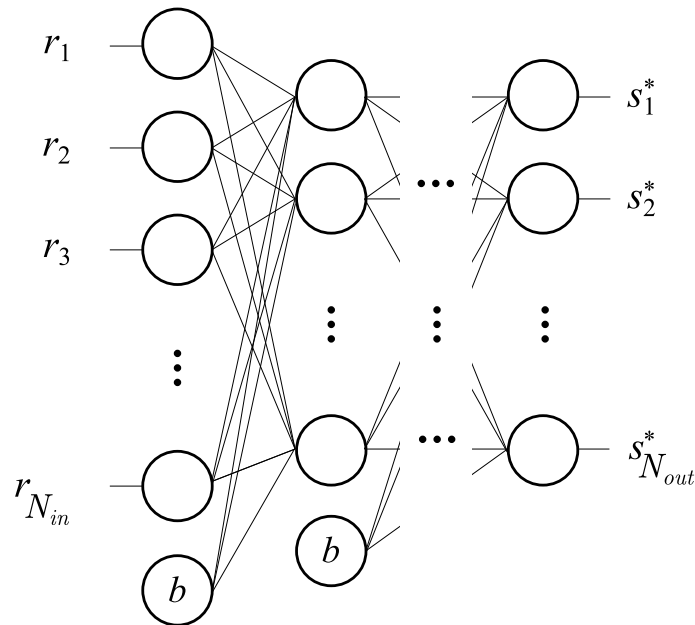


Figure 2.1: Diagram of an MLP with N_{in} inputs and N_{out} outputs showing neurons (circles) arranged in layers connected by weights (lines).

It is the numerical values of the weights and the types of activation functions which are key to the performance of multi-layer perceptrons. If at least one layer of the multi-layer perceptron has non-linear activation functions, then the MLP is a universal approximator and can be applied to non-linear regression problems. The specific numerical values of the weights determine which problem the MLP is approximating. The process of determining the weights is itself a non-linear optimisation process, known as training. For MLPs, training is a supervised learning process, where the MLP makes predictions for a given input and the error between the network output and the target output is used to adjust the weights to minimise a given loss function.

2 Theoretical background

The network output, $\mathbf{s}^*$, for the corresponding input, $\mathbf{r}$, is obtained by feeding the input forward through the network layers. For a network with N_{hidden} hidden layers, the output from the neurons in the first hidden layer, $\mathbf{n}^1$, is calculated using:

$$\mathbf{n}^1 = \text{act}^1 (\mathbf{W}^1 \mathbf{r} + \mathbf{b}^1), \quad (2.73)$$

where act^1 is the activation function for the first hidden layer, $\mathbf{W}^1$ the matrix of weights connecting the first hidden layer neurons to the neurons in the previous layer (in this case, the input layer) and $\mathbf{b}^1$ the vector of weights connecting the first hidden layer neurons to the bias neuron in the previous layer. This process is repeated through the remaining hidden layers using:

$$\mathbf{n}^l = \text{act}^l (\mathbf{W}^l \mathbf{n}^{l-1} + \mathbf{b}^l) \quad \text{with } l = 2, 3, \dots, N_{hidden}, \quad (2.74)$$

before the network output, $\mathbf{s}^*$, is retrieved using

$$\mathbf{s}^* = \text{act}^{N_{hidden}+1} (\mathbf{W}^{N_{hidden}+1} \mathbf{n}^{N_{hidden}} + \mathbf{b}^{N_{hidden}+1}). \quad (2.75)$$

The number of inputs and the number of outputs of the multi-layer perceptron are generally determined by the problem at hand. The number of hidden layers, the number of neurons in each hidden layer and the types of activation functions used on each layer are all tunable parameters which must be determined by the user. Along with the number of network inputs and outputs, the number of hidden layers and number of neurons in each hidden layer determine the number of weights the network has according to:

$$N_{weights} = (N_{in} + 1) \times N_{neurons}^1 + \sum_{l=2}^{N_{hidden}} (N_{neurons}^{l-1} + 1) \times N_{neurons}^l + (N_{neurons}^{N_{hidden}} + 1) \times N_{out}, \quad (2.76)$$

where $N_{weights}$ is the number of weights in the network, N_{in} is the number of inputs, N_{out} is the number of outputs and $N_{neurons}^l$ is the number of neurons in hidden layer l . Presuming that the standard of training is roughly the same regardless of network size, as the number of weights increases so does the predictive power of the network, meaning that a network with more weights is able to better reproduce the target outputs for a given problem than a network with fewer weights. However larger

networks also have greater inference time, so to gain the maximum speed-up when using ANN tabulation the network size should be kept as small as possible.

With this in mind, there are no strict rules for determining the MLP architecture, although a number of heuristics can be used. The first is that although single hidden layer MLPs are universal approximators, MLPs with two hidden layers have significantly more predictive power than those with one hidden layer. Addition of further hidden layers generally provides more accuracy, although the accuracy increases in progressively smaller increments with each additional hidden layer added. It is important to avoid significant changes in the number of neurons between hidden layers, as significant narrowing or expansion between layers may lead to more difficult training. An additional consideration is the type of activation function used. Any function can be used, as long as at least one hidden layer contains non-linear activation functions. Commonly used activation functions include hyperbolic tangent, sigmoids, rectified linear units, leaky rectified linear units and linear functions. The time required to compute the activation function is also a factor in the ANN tabulation time, with hyperbolic tangent and sigmoid functions taking the longest and linear functions being very cheap to compute. However, training with rectified linear units and their variations is generally more difficult due to the discontinuity present in such functions. Finally, MLP training is more successful when the numerical range of the inputs and outputs is consistent. This means that variables are almost always scaled to a common range, which can be performed using any normalisation or standardisation approach.

Once an MLP architecture is determined, training of the MLP can take place. This training is performed over a data set consisting of N_{data} pairs of inputs and their corresponding outputs. After the network output is calculated it can be used along with the target output, $\mathbf{s}$, to adjust the MLP weights and train the MLP. The measure of difference between the target output and the network prediction over the training data set is expressed in terms of a loss function, with the mean square error (MSE) given by:

$$L(\mathbf{w}) = \frac{1}{N_{data}} \sum_{p=1}^{N_{data}} [(\mathbf{s}_p - \mathbf{s}_p^*)^T (\mathbf{s}_p - \mathbf{s}_p^*)], \quad (2.77)$$

where L is the loss function and $\mathbf{w}$ is the vectorised version of the weight and bias matrices. The aim of training is to find the optimal weights which minimise the given loss function using the error between predictions and targets.

2.3.1.1 Principles of training

Training begins by setting the network weights to some initial values $\mathbf{w}_0$. An iterative procedure then occurs where the weights at iteration k are used to determine the new weights at iteration $k + 1$. The new weights can be obtained by considering a Taylor expansion of $L(\mathbf{w})$ around the old weights:

$$\begin{aligned} L(\mathbf{w}_{k+1}) &= L(\mathbf{w}_k + \Delta\mathbf{w}_k) = \\ &L(\mathbf{w}_k) + \Delta\mathbf{w}_k^T \nabla L(\mathbf{w}_k) + \frac{1}{2} \Delta\mathbf{w}_k^T \nabla^2 L(\mathbf{w}_k) \Delta\mathbf{w}_k + \dots, \end{aligned} \quad (2.78)$$

where a first order Taylor expansion gives:

$$L(\mathbf{w}_{k+1}) = L(\mathbf{w}_k + \Delta\mathbf{w}_k) = L(\mathbf{w}_k) + \Delta\mathbf{w}_k^T \nabla L(\mathbf{w}_k). \quad (2.79)$$

The value of the loss function will decrease the fastest when $\Delta\mathbf{w}_k$ is aligned along $-\nabla L(\mathbf{w}_k)$, so the new weights that move towards the minimisation of the loss function are given by:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \zeta \nabla L(\mathbf{w}_k) = \mathbf{w}_k - \zeta \left. \frac{\partial L}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_k} \quad (2.80)$$

where the learning rate, ζ , has been introduced to reduce the size of the weight updates as the first order approximation is only suitable for small $\Delta\mathbf{w}_k$. The initial value of the learning rate (and if/how it updates during the training process) is determined by the user. A large learning rate will lead to large weight updates and so potentially faster convergence, although oscillations and stability issues may arise if the rate is too large. A small learning rate is generally more stable but has slower convergence. Therefore typically learning rates are large at the start of training and decay as training progresses to fine tune the weights. In Equation 2.80, the weight update occurs after all of the training data is presented to the MLP. Faster convergence to the minimum of an approximation of the loss function can be found by performing a weight update after a small number of inputs are presented. If each data point is randomly chosen from the training set and presented to the MLP in turn, the MSE can be approximated using the squared error for a given data point p :

$$\hat{L}(\mathbf{w}_p) = (\mathbf{s}_p - \mathbf{s}_p^*)^T (\mathbf{s}_p - \mathbf{s}_p^*). \quad (2.81)$$

While the minimisation of the MSE will occur where the gradient of the MSE with respect to the weights is zero, the approximate gradient, or stochastic gradient, of the MSE with respect to the weights is:

$$\nabla \hat{L} = \nabla((\mathbf{s}_p - \mathbf{s}_p^*)^T(\mathbf{s}_p - \mathbf{s}_p^*)), \quad (2.82)$$

while the weight updates for the stochastic gradient descent algorithm are:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \zeta \nabla \hat{L}(\mathbf{w}_k) = \mathbf{w}_k - \zeta \left. \frac{\partial \hat{L}}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_k}, \quad (2.83)$$

with the iteration and the data point now interchangeable. Splitting the vectorised weights and biases back into their respective matrices and vectors for each layer, the weight and bias updates for the stochastic gradient descent algorithm are:

$$\Delta w_{ij}^l = -\zeta \frac{\partial \hat{L}}{\partial w_{ij}^l}, \quad (2.84)$$

and

$$\Delta b_i^l = -\zeta \frac{\partial \hat{L}}{\partial b_i^l}. \quad (2.85)$$

The derivative of the approximate MSE on the output layer is easy to calculate, and so weight updates for the output layer are straightforward. As the error at the output layer is not an explicit function of the weights in the hidden layers, the chain rule must be used to evaluate the gradient of the loss function with respect to the hidden layer weights and biases. This uses a method known as backpropagation, further details of which can be found in Hagan et al. [96].

The basic principles behind stochastic gradient descent are used in a variety of different gradient based algorithms for MLP training such as AdaGrad, RMSProp and Adam which are all commonly used to train MLPs. Further details on these algorithms and others can be found in the review of Ruder [97]. Gradient based algorithms are popular due to the advancement of deep learning, where MLPs containing millions or even billions of weights are used. Although computing the gradient of the loss function with respect to the weights is costly for a large number of weights, it remains feasible. However, for large networks it is currently not feasible to include quasi-second order or second-order information, which approximates or contains second derivatives of the loss function with respect to the weights. How-

ever, second-order or quasi-second-order algorithms can have faster convergence and be less prone to local minima, meaning that higher accuracy can be achieved.

2.3.1.2 Training with the Levenberg-Marquardt algorithm

One such quasi-second-order algorithm is the Levenberg-Marquardt algorithm [98], which has fast convergence without overwhelming computational cost. The Levenberg-Marquardt algorithm considers the sum of squared errors (over the entire training set, which is directly proportional to the MSE) as the loss function, and can be derived by including the second order term in the Taylor expansion:

$$\begin{aligned} L(\mathbf{w}_{k+1}) &= L(\mathbf{w}_k + \Delta \mathbf{w}_k) = \\ &L(\mathbf{w}_k) + \Delta \mathbf{w}_k^T \nabla L(\mathbf{w}_k) + \frac{1}{2} \Delta \mathbf{w}_k^T \nabla^2 L(\mathbf{w}_k) \Delta \mathbf{w}_k. \end{aligned} \quad (2.86)$$

By taking the gradient with respect to the weight updates and setting equal to zero:

$$\nabla L(\mathbf{w}_k) + \nabla^2 L(\mathbf{w}_k) \Delta \mathbf{w}_k = 0, \quad (2.87)$$

the weight update is then found by rearrangement:

$$\Delta \mathbf{w}_k = -(\nabla^2 L(\mathbf{w}_k))^{-1} \nabla L(\mathbf{w}_k). \quad (2.88)$$

As $\nabla^2 L(\mathbf{w}_k)$ may not be invertible a second term is added:

$$\Delta \mathbf{w}_k = -(\nabla^2 L(\mathbf{w}_k) + \zeta \mathbf{I})^{-1} \nabla L(\mathbf{w}_k), \quad (2.89)$$

where ζ can be increased to ensure $\nabla^2 L(\mathbf{w}_k) + \zeta \mathbf{I}$ is invertible. ζ can be thought of as a learning rate, although as ζ becomes large weight updates decrease and the method approaches that of steepest descent, which is reliable but slow converging. Similarly, as ζ becomes small the method approaches the Gauss-Newton method which is fast converging but can be unstable. The Hessian, $\nabla^2 L(\mathbf{w}_k)$, is not computed in its entirety (hence the method is quasi-second-order) but is approximated as:

$$\nabla^2 L(\mathbf{w}_k) \approx 2\mathbf{J}^T \mathbf{J}, \quad (2.90)$$

where $\mathbf{J}$ is the Jacobian matrix containing terms of the form:

$$\mathbf{J}(\mathbf{w}) = \left. \frac{\partial \mathbf{e}}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_k}, \quad (2.91)$$

with the error vector $\mathbf{e}$ containing the error for each individual network output for each of the samples in the training set and so being $(N_{out} \times N_{data}) \times 1$ in size. The vector containing the gradient of the loss function with respect to the weights can also be expressed in terms of the Jacobian and the error vector, giving the weight updates in the Levenberg-Marquardt algorithm as:

$$\Delta \mathbf{w}_k = -(\mathbf{J}^T \mathbf{J} + \zeta \mathbf{I})^{-1} \mathbf{J}^T \mathbf{e}. \quad (2.92)$$

The terms in the Jacobian are analogous to those calculated in the stochastic gradient descent algorithm, although their computation is slightly more involved as there is now a term for each input point presented and for each MLP output. Further details on how to calculate these terms can be found in Hagan et al. [96]. While the LM algorithm is fast converging and accurate, a key issue is the size of the Jacobian, which is $(N_{out} \times N_{data}) \times N_{weights}$ and so depends on the number of weights and number of data points in the training set. The number of MLP outputs is less important as that is normally relatively small and is fixed by the problem at hand. However, the number of weights and the number of data can easily reach tens or hundreds of thousands each. The Jacobian matrix may then have millions of entries and so this precludes the use of the LM algorithm for deep learning architectures or applications with large numbers of data where it is unfeasible to even store the Jacobian matrix.

The LM algorithm can be rendered feasible for chemical mechanism tabulation with large data sets by recognising that the Jacobian matrix is never required in its entirety, but instead is used as an intermediate to acquire the approximate Hessian (of size $N_{weights} \times N_{weights}$) and the gradient (of size $N_{weights} \times 1$). Through the modification of Wilamowski and Yu [99], the size of the Jacobian can be decoupled from the size of the data set. Figure 2.2 shows the modification. Rather than performing row-column multiplication of the Jacobian across all of the training set samples to generate individual scalar values in the Hessian, column-row multiplication can be performed. This generates a sub-Hessian of size $N_{weights} \times N_{weights}$. By repeating this process across all the input samples and iteratively summing the sub-Hessian, the final Hessian can be created. This procedure can also be performed with the gradient. This method can also be readily performed batch-wise, to reduce the number of summations over the sub-Hessian/sub-gradient. The repeated calculation and summation of the sub-matrices does incur a slight computation cost compared to

$$\begin{array}{c}
 \begin{array}{c} N_{out} \times N_{data} \\ \boxed{} \\ J^T \end{array} \times \begin{array}{c} \boxed{} \\ N_{out} \times N_{data} \\ J \end{array} = \begin{array}{c} N_{weights} \\ \boxed{} \\ \text{Hessian} \end{array} N_{weights} \\
 \\
 \begin{array}{c} \boxed{} \\ N_{weights} J^T \end{array} \times \begin{array}{c} \boxed{} \\ N_{weights} \\ J \end{array} = \begin{array}{c} N_{weights} \\ \boxed{\text{sub - Hessian}} \end{array} N_{weights}
 \end{array}$$

Figure 2.2: Modification to the Levenberg-Marquardt algorithm to allow the training of large networks with large data sets.

using the Jacobian in its entirety, however the fact that this method allows the LM algorithm to be used for realistic MLP sizes with large data sets is a key advantage.

2.3.1.3 Preventing overfitting with Bayesian regularisation

The modified LM algorithm is a very capable MLP training method. However, the ultimate aim is to maximise performance of the MLP in the application. Assuming that the training set and application data share a similar distribution, generally as the accuracy of the MLP in the training set increases, the accuracy in the testing set will increase too. However, there may come a point during the training process where the MLP is learning exactly the patterns in the training set and not the general underlying function, which is known as overfitting. Large networks with more predictive power are more susceptible to overfitting. To prevent problems with overfitting, it is common to choose the size of MLPs such that they are capable of performing the task at hand to an acceptable degree of accuracy, and then employ a technique known as regularisation to prevent overfitting and improve the generality of the MLPs.

Common regularisation methods include early stopping, L1 and L2 regularisation

and dropout. Each of these methods suffers from having further tunable parameters which should be optimised to ensure the best performance possible is achieved. An alternative approach is to use Bayesian regularisation [100], which has the key advantage that the regularisation parameters are adjusted automatically and do not need to be tuned. Bayesian regularisation involves the modification of the sum of squared errors loss function to include a term containing the sum of squares of the weights themselves:

$$L = \lambda_1 \sum_{p=1}^{N_{data}} (\mathbf{s}_p - \mathbf{s}_p^*)^\top (\mathbf{s}_p - \mathbf{s}_p^*) + \lambda_2 \mathbf{w}^\top \mathbf{w}, \quad (2.93)$$

where λ_1 and λ_2 are parameters that are adjusted automatically during the training process. The Bayesian regularisation method aims to maximise the posterior density $\mathcal{P}(\mathbf{w}|\mathbf{r}_p, \lambda_1, \lambda_2)$, with $\mathbf{r}_p$ referring to all of the training data. If a Gaussian prior distribution is assumed for the network weights, maximising $\mathcal{P}(\mathbf{w}|\mathbf{r}_p, \lambda_1, \lambda_2)$ can be shown to be equivalent to minimising the regularised loss function given by Equation 2.93. Further details of the implementation of Bayesian regularisation and how λ_1 and λ_2 are modified during training can be found, for example, in Dan Foresee and Hagan [101] or Hagan et al. [96].

From a practical perspective, a weight with a value of zero will not feed any input it receives forward, so keeping the magnitudes of the weights low can be thought of as decreasing the effective size of the MLP. Bayesian regularisation also allows an estimation of the effective number of MLP weights used. If the effective number of MLP weights used is close to the maximum number available, then this suggests that increasing the size of the MLP and then retraining will give a significant increase in accuracy. After the MLP training is complete, the MLPs can be used to replace the chemical mechanism in the application of choice.

2.3.2 The self-organising map

The self-organising map [53] is a type of ANN capable of finding regions of similarity in multidimensional data. The SOM consists of a user-defined number of neurons, each one represented by a vector in the multidimensional space considered. There are no connections between the SOM neurons.

There are two stages to using the SOM. The first is the training phase, where training data is used to move the SOM neurons in the multidimensional space to attempt to represent any regions of similarity in the training data. Training the self-organising map is an unsupervised process, using a competitive learning approach.

2 Theoretical background

After training, the SOM neuron vectors are fixed and the second stage occurs when the SOM is used, with new data presented and assigned to a given SOM neuron based on a metric of closeness between the presented point and the vector associated with each SOM neuron.

At the start of the training phase, the SOM neuron vectors are initialised to random values. Then, a randomly chosen data point, $\mathbf{r}$ from the training set is presented to the SOM. The Euclidean distances between all of the SOM neurons and the presented training point are computed. The closest SOM neuron to the presented training point in Euclidean distance is the winning neuron, $\mathbf{v}_i^{som}$. Then, all of the SOM neuron vectors, j , are adjusted using:

$$\mathbf{v}_j^{som}(k+1) = \mathbf{v}_j^{som}(k) + \zeta(k) \theta(i, j, k) \|\mathbf{r}(k) - \mathbf{v}_j^{som}(k)\|, \quad (2.94)$$

where k is the current iteration, ζ is a learning rate introduced to moderate changes, $\|\cdot\|$ is the Euclidean norm and θ is a Gaussian neighbourhood function, given by:

$$\theta(i, j, k) = \exp \left(-\frac{\|\mathbf{v}_j^{som}(k) - \mathbf{v}_i^{som}(k)\|}{2\zeta^2(k)} \right), \quad (2.95)$$

where ζ^2 defines the width of the Gaussian kernel. The purpose of the neighbourhood function is to move SOM neurons which are close to the winning neuron towards the presented data point, while leaving SOM neurons which are distant from the winning neuron unchanged. Training continues for a predetermined number of iterations or until the changes in the SOM neurons over the whole data set are smaller than a predetermined threshold, during which time the learning rate and the neighbourhood width decrease monotonically to fine tune the SOM neuron vectors.

At the end of the training phase, all of the training data can be assigned to SOM regions by computing the Euclidean distance between a given point from the training data and all of the SOM neuron vectors. This divided training data can then be used for MLP training. As similar regions of the composition space will likely have similar reaction behaviour, the motivation for using the SOM is that the task of imitating the chemical mechanism over each SOM region is made easier for the MLPs that follow.

To use the SOM in an application, only the SOM neuron vectors need to be stored. Points encountered in an application are assigned to a given SOM region in the same way that the training data is divided once training has finished, with points assigned to the SOM neuron which is closest in Euclidean distance. Although assigning new points to SOM regions requires the computation of the Euclidean

distance between every presented point and all of the SOM neurons, the number of SOM neurons is generally of the order of tens or hundreds and so the process is relatively quick.

2.4 Thermochemistry tabulation with artificial neural networks

In this section, the SOM-MLP architecture is outlined. Following this, a method of generating combustion realistic ANN training data which is not specific to a given test case and is capable of producing ANNs which can successfully replace the integration of the reaction source terms in simulations of turbulent flames [57, 58] is explained.

2.4.1 The SOM-MLP method

A commonly used approach (for example, [37, 52, 55, 36, 48]) for tabulation of combustion thermochemistry involves the combination of the SOM with MLPs. Once suitable training data is collected and scaled such that each variable shares a common range, the SOM can be trained as described in Section 2.3.2 to find regions of similarity in composition space. Figure 2.3 indicates the resulting composition space regions found by applying the SOM to canonical combustion problem data.

Once the training data is divided into these regions, they can be effectively treated as individual data sets. For each of the coloured data sets in Figure 2.3, separate MLPs can be trained. These MLPs specialise to the reaction dynamics contained in their respective composition space regions and so are potentially able to achieve higher accuracy than MLPs trained over larger regions of the composition space where the reaction dynamics may be more varied. The number of SOM neurons chosen influences the size of the resulting composition space regions and so the degree to which individual MLPs specialise to their respective composition space regions. Typically, a few hundred SOM neurons are used, which produces accurate MLPs without encountering significant problems with overfitting due to a small number of data in each composition space region. Using the SOM-MLP method with a single MLP for each resulting composition space region has been shown to be more accurate than using a single MLP over the entire composition space [44].

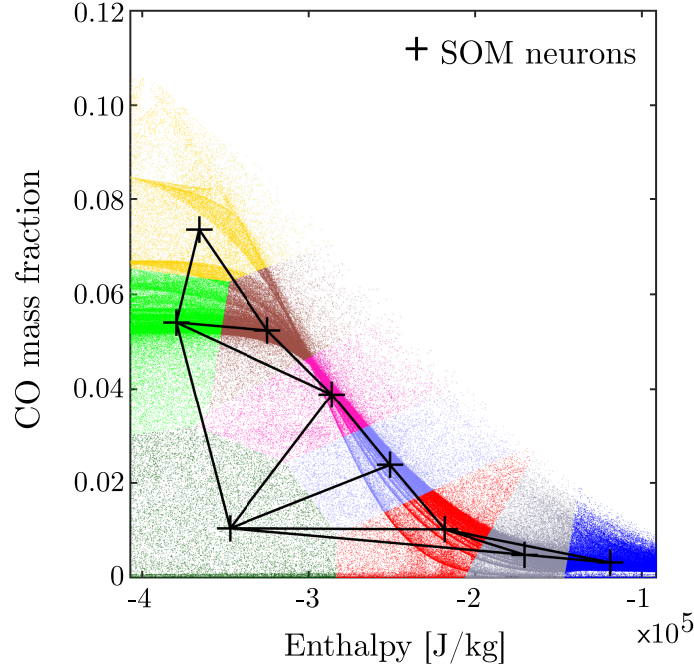


Figure 2.3: Composition space regions found in canonical combustion problem data using the SOM, denoted with different colours.

2.4.2 Random data generation

In Section 2.1.1 the bounds of the composition space resulting from the mathematical definition of the composition space variables were given. These bounds define the largest size the composition space can possibly be, and is referred to here as the theoretical composition space. The composition space can be of a very high dimensionality depending on the number of species contained in the mechanism. It is possible to sample from the theoretical composition space directly (in mathematics this is known as sampling from the unit simplex). However, for a number of reasons, the realistic composition space that is accessed in reacting flow simulations is often much smaller than the theoretical composition space. Firstly, although the species mass fractions are bounded between zero and unity, depending on the reaction mechanism chosen the actual maximum value reached for many species mass fractions is often orders of magnitude less than unity. Secondly, the thermodynamic conditions of the given problem influence the composition space. For example, if the pressure is atmospheric and constant the composition space becomes significantly smaller than if pressure is variable. Thirdly, the chosen fuel and oxidiser and their respective quantities further restricts the size of the composition space. Therefore, it is both unnecessary and undesirable to sample from across the theoretical composition space. Figure 2.4 illustrates theoretical and realistic composition spaces, as well as indicative smaller composition spaces when considering different flame regimes.

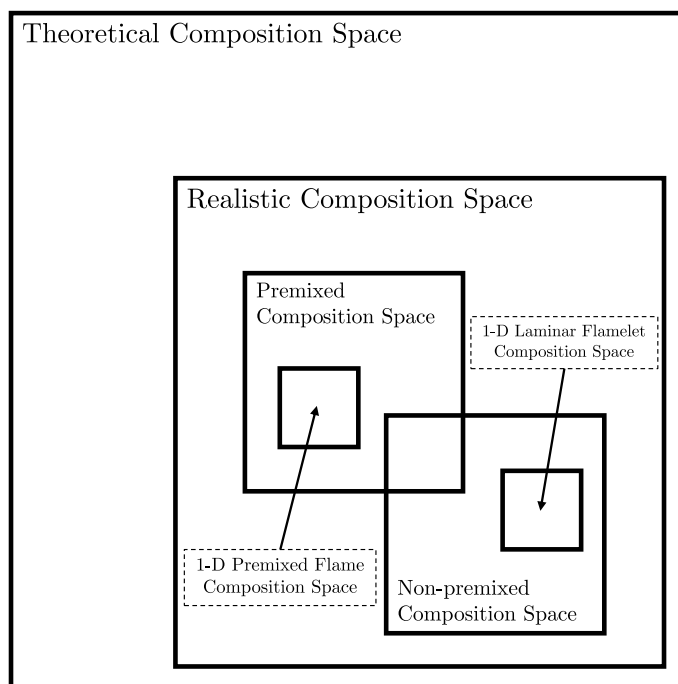


Figure 2.4: Indicative diagram showing relations between different composition spaces.

Rather than sampling from the theoretical composition space, realistic composition space data can be obtained through the use of canonical combustion problems. Data collected from these CCPs is necessarily less general than data collected from the theoretical composition space and so CCPs which are well suited to the desired application should be chosen. In this work two different CCPs are used depending on the flame regime. For non-premixed flames, 1D laminar flamelets are used, while for premixed flames, 1D premixed flames are used. Indicative composition spaces covered by these CCPs are also given in Figure 2.4. Despite targeting a realistic area of the composition space, the use of 1D flames, or indeed any other CCP does have limitations. The first is that all of the data collected from the CCP conforms to the equations used to simulate the CCP. In a general turbulent flame case it is very unlikely that the data will conform exactly to CCP equations. This means that the use of CCP on its own can not be expected to cover a general turbulent flame composition space. In Figure 2.4, this is illustrated by the 1D flame data occupying only a portion of the composition space of its respective flame regime.

The second limitation is a technical issue caused by the ANNs themselves. In applications reacting points will pass through the ANNs many times as the simulations advances in time. Because of the inevitable error associated with ANN predictions, this means that as time progresses the ANNs themselves modify the composition space they encounter. This modification happens in four main ways. The first is that the maximum species mass fractions of some species may become higher than previously encountered. As long as the maximums do not greatly exceed

those found in the training data set this is not a significant issue. The second way is that the ANNs may predict unphysical negative mass fractions. These can be set to zero if encountered. In the absence of enforced conservation, the third way is that the ANN predictions will not satisfy mass conservation. Even without enforced conservation, mass conservation errors are easy to rectify by normalising the sum of the species mass fractions to unity. The final way that ANNs without enforced conservation modify the composition space is by altering the element mass fractions and the ratios between them. The alteration of the element mass fractions is not a big issue as the CCP data generally contains varying mixtures of fuel/oxidiser and so a wide range of element mass fractions. However, the ratios between element mass fractions are problematic. Considering a 1D CH_4 /air flame with no differential diffusion, any composition in the accessed composition space will have the same ratio between certain elements (neglecting numerical errors). For example, the ratio of the H/C element mole fraction will be 4 for any composition containing C or H. Likewise, the ratio of the O/N element mole fraction will be the same as that in air, approximately 0.264. Without conserving the element mass fractions, it is not easy to rectify these variable ratios. Therefore, in simulations without enforced conservation it is desirable to incorporate data with variable element ratios into the ANN training data set.

To overcome the aforementioned limitations in the CCP data while still targeting a relevant area of the composition space, a random data generation method is used. This method takes the CCP data as a basis, before modifying it to expand its generality by increasing the mass fraction bounds and if desired includes data with variable element ratios. The aim of random data generation is to expand the generality of the data sampled from the CCPs and introduce variable element ratios to the data, while remaining in the relevant region of composition space. This means that while the bounds of the composition space variables should stay roughly as they are, intermediate values away from the bounds can have significant changes to increase coverage within the composition space bounds. As explained, a composition is described by:

$$\varphi_\alpha = [h, Y_i]. \quad (2.96)$$

The treatment of the enthalpy is straightforward as its bounds are quite well defined by the CCP data. Additionally, modifying intermediate values of enthalpy too heavily will lead to compositions which are very unlikely to be accessed and do not occupy the relevant area of the composition space. Therefore, the enthalpy should only undergo minor modifications. As pressure is assumed to be constant in the application, then it can be ignored and not feature in the tabulation process at

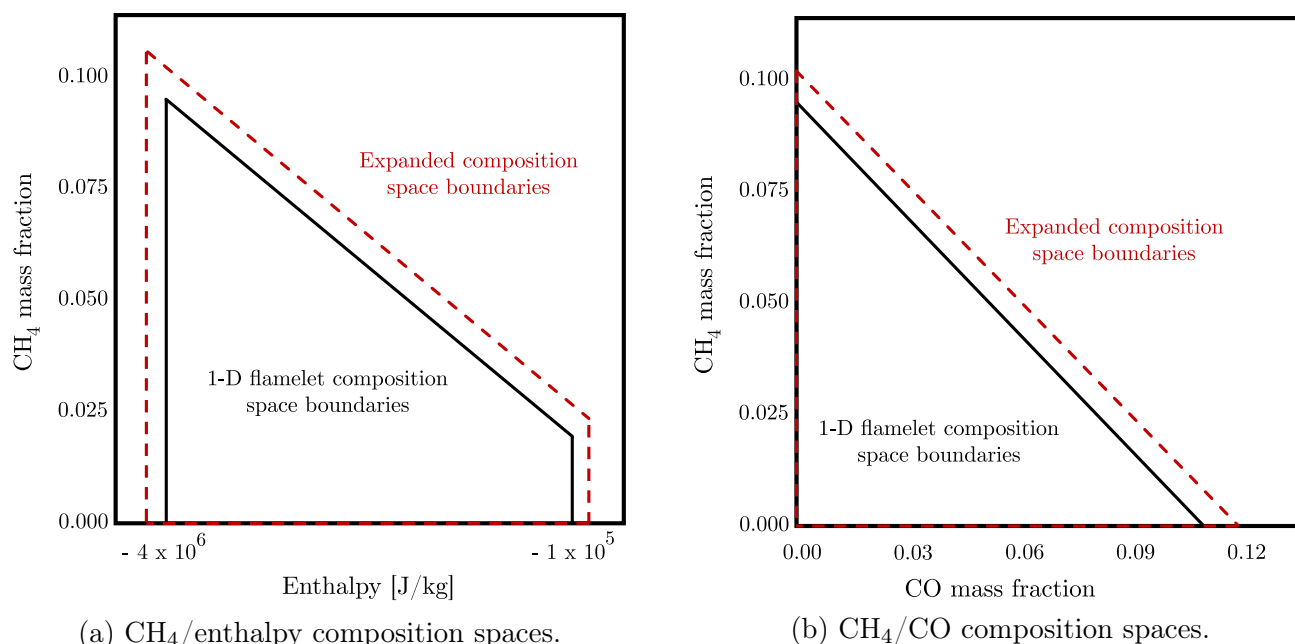


Figure 2.5: Indicative diagrams of composition spaces from 1D non-premixed laminar flames as well as the desired expanded composition spaces.

all.

The treatment of the species is more involved. Like the enthalpy (and pressure if included), major species bounds are quite well defined by the amounts of fuel and oxidiser and the ratios between elements, so altering values near the boundaries too much will lead to unphysical values. Figure 2.5a shows an indication of the CH_4 /enthalpy composition space when sampling from 1D non-premixed CH_4 /air laminar flamelets within the flammability limits. The maximum value of CH_4 is linearly related to the enthalpy, while its boundaries are given by the imposed flammability limits. Although the linear relation between CH_4 and enthalpy no longer needs to be satisfied in the random data, the expanded composition space boundaries for these variables should be reasonably close to the original composition space.

Furthermore, as the major species contribute to the vast majority of the elements mass fractions, keeping the composition space boundaries roughly as they are will also prevent lots of data with unrealistic element mass fractions and ratios being generated. Figure 2.5b shows an indication of the CH_4 /CO composition space. The diagonal line on Figure 2.5b arises because of the element constraints, meaning that a large amount of CH_4 cannot exist at the same time as a large amount of CO as there is a limited amount of the C element. Again, this means that the composition space boundaries dictated by the element constraints should not change too much.

Compositions containing lesser amounts of CH_4 , CO and the other major species can be altered significantly without generating unlikely data from an element per-

spective. This is also true for the minor species which do not contribute much to the element mass fractions because they occur in small amounts. This also means that the minor species boundaries are not well known ahead of time and depend much more on the specifics of the mechanism and application. Finally, no negative species mass fractions should be generated as these will be automatically unphysical, indicated by the lines on the axes in Figures 2.5a and 2.5b.

This leads to requirements for a random data generation method which a) introduces small amounts of random noise to some composition space variables (enthalpy, pressure if considered, high magnitude major species) and b) introduces significant amounts of random noise to other composition space variables (low magnitude major species, minor species), without causing negative species mass fractions. To achieve this, the enthalpy (and pressure) can be modified using:

$$h^r = h + a_1 \frac{\max(h) - \min(h)}{b_1}, \quad (2.97)$$

where h^r is the randomly generated value based on the original value of h , a_1 is a uniform random number between minus one and one and b_1 is a factor to determine how large the changes in h are allowed to be. For hydrocarbon fuel/air flames the N_2 mass fraction is generally both large and does not change much due to reaction, even with nitrogen chemistry included. Therefore in such flames it can be altered in an analogous way to the enthalpy:

$$Y_{N_2}^r = Y_{N_2} + a_2 \frac{\max(Y_{N_2}) - \min(Y_{N_2})}{b_2}, \quad (2.98)$$

where a_2 is a different uniform random number between one and minus one and b_2 is a factor to determine the size of the allowed changes in Y_{N_2} . The remaining species mass fractions are then modified individually using:

$$Y_i^r = 10^{\left(c + \frac{a_{3,i}c}{b_3}\right)} \quad \text{where} \quad c = \log_{10}(Y_i), \quad (2.99)$$

where Y_i^r is the randomly generated species mass fraction for the chosen species and Y_i the respective mass fraction of the chosen species in the original data set. Again, $a_{3,i}$ is a vector of different uniform random numbers between minus one and one, while b_3 is another factor used to vary the size of changes in the data. Figure 2.6 shows the bounds (solid lines) of random mass fractions generated using Equation 2.99 when $b_3 = 4/3$ as well as the original mass fraction (dashed line). The use of Equation 2.99 ensures that no negative species mass fractions or mass fractions exceeding one are generated. Additionally, Figure 2.6 shows that the bounds are the

wider when the mass fraction is smaller, before becoming narrower as the mass fraction increases, which is in keeping with the desired behaviour. Once the randomly generated species mass fractions are determined, they are normalised to satisfy the definition of mass fractions. For a fuel/air flame where N_2 is modified linearly:

$$Y_i^r = (1 - Y_{N_2}^r) \frac{Y_i^r}{\sum_{i=1}^{N_{sp}-1} Y_i^r}, \quad (2.100)$$

while for a general reacting system where N_2 is not treated separately:

$$Y_i^r = \frac{Y_i^r}{\sum_{i=1}^{N_{sp}} Y_i^r}. \quad (2.101)$$

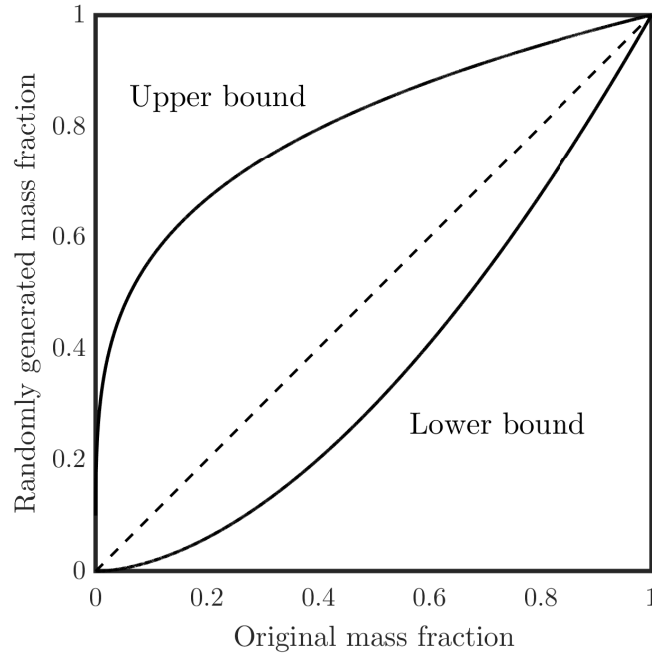


Figure 2.6: Boundaries on randomly generated mass fraction with $b_3 = 4/3$.

A question arises of how much randomness to introduce to the variables using b_1 , b_2 and b_3 . This depends on how much the composition space expansion is desired, but through experience it has been found that using $b_1 = b_2 = b_3 \approx 10$ leads to data which is significantly different from the CCP data but still remains in a realistic area of the composition space. After the random data is generated, the element mass fractions and ratios are calculated. Each of the random data compositions passes through an acceptance-rejection step to ensure that the element mass fraction ratios are within predefined limits. If the newly generated point does not lie within the element ratio window then it is discarded and new a_1, a_2 and $a_{3,i}$ generated for that point until the element ratios lie within the window. As with the ranges used to control the degree of randomness in the data, these element ratio limits are selected

2 Theoretical background

based on experience of the likely element ratio ranges experienced in simulations without enforced conservation using ANNs. It has been found that using windows of $\approx \pm 5\%$ around the exact element ratio (in conjunction with $b_1 = b_2 = b_3 \approx 10$) leads to realistic data without a significant number of failures.

Chapter 3

Development of a machine learning methodology for non-premixed flames

Chapter 2 explained the theoretical background necessary for thermochemistry tabulation with ANNs. However, the specifics of the ANN architecture and the method used to generate the training data can greatly influence overall simulation accuracy. The overarching aim is to produce ANNs which are as general as possible, and are therefore not specialised to a single flame and can be applied to a variety of flame cases with high prediction accuracy. To produce general ANN training data, the random data generation method of Ding et al. [58] is used. Data collected from LES-PDF simulations of a non-premixed turbulent jet flame with the stochastic fields method are used to assess the generality of different ANN architectures. The most general ANN architecture is then used to perform reaction source term computations in the LES-PDF simulations of the non-premixed flame with a comparison made with simulations performed with conventional reaction source term integration.

The work in this chapter has been published in Readshaw et al. [57].

3.1 An artificial neural network architecture for improved generalisation

The canonical problem selected to generate data for non-premixed flame applications is that of 1D laminar flamelets. Such flamelets are 1D in mixture fraction, z , and are governed by Equation 3.1:

$$\rho \frac{\partial Y_i}{\partial t} = \rho \frac{\chi(z)}{2} \frac{\partial^2 Y_i}{\partial z^2} + \dot{\omega}_i, \quad (3.1)$$

with χ the scalar dissipation rate, given by:

$$\chi(z) = \frac{\varpi}{\pi} \exp \left[-2(\operatorname{erfc}^{-1}(2z))^2 \right], \quad (3.2)$$

where ϖ is the strain rate and erfc^{-1} the inverse of the complementary error function. Varying the strain rate varies the level of diffusion in the flamelets, allowing a range of conditions between equilibrium and extinction to be simulated.

To collect training data, Equation 3.1 was discretised and then simulated with a fractional step method. Data was sampled at randomly selected mixture fraction values within the CH_4/air mixture fraction flammability limits, interpolating if necessary, as the simulation progressed in time. Once the flamelet reached steady state or extinguished the simulation was stopped to avoid collecting large amounts of similar data. Table 3.1 gives a list of variables which can be altered in 1D flamelet simulations in order to change the sampled composition space data. Where applicable, these should be matched to the values anticipated to occur in the application.

Table 3.1: Variables which can be altered in 1D flamelet simulations to change the sampled composition space.

1D Flamelet Variable
Unburnt temperature
Pressure
Strain rate
Pilot location
Time step
Samples per time step
Simulation length
Mixture fraction sampling limits

To generate training data for the application of ANNs to non-premixed flames one hundred such flamelets were simulated using a fixed time step of 1×10^{-6} s, with strain rates sampled randomly between 1 s and 1100 s. Half of the flamelets were initialised in a similar fashion to Chatzopoulos and Rigopoulos [52] and Franke et al. [55], where an artificial pilot was used to ignite the flamelets, with the pilot held for 1000 time steps before being removed to allow the flamelet profile to develop over the whole domain. The other half of the flamelets were initialised by setting the whole domain to equilibrium, in order to collect more composition states close

Table 3.2: Parameters used for random data generation on the 1D laminar flamelet data.

Parameter	Value
b_1	8
b_2	8
b_3	10
Molar H/C window	3.8 - 4.2
Molar O/N window	0.254 - 0.274
z window	0.02 - 0.10

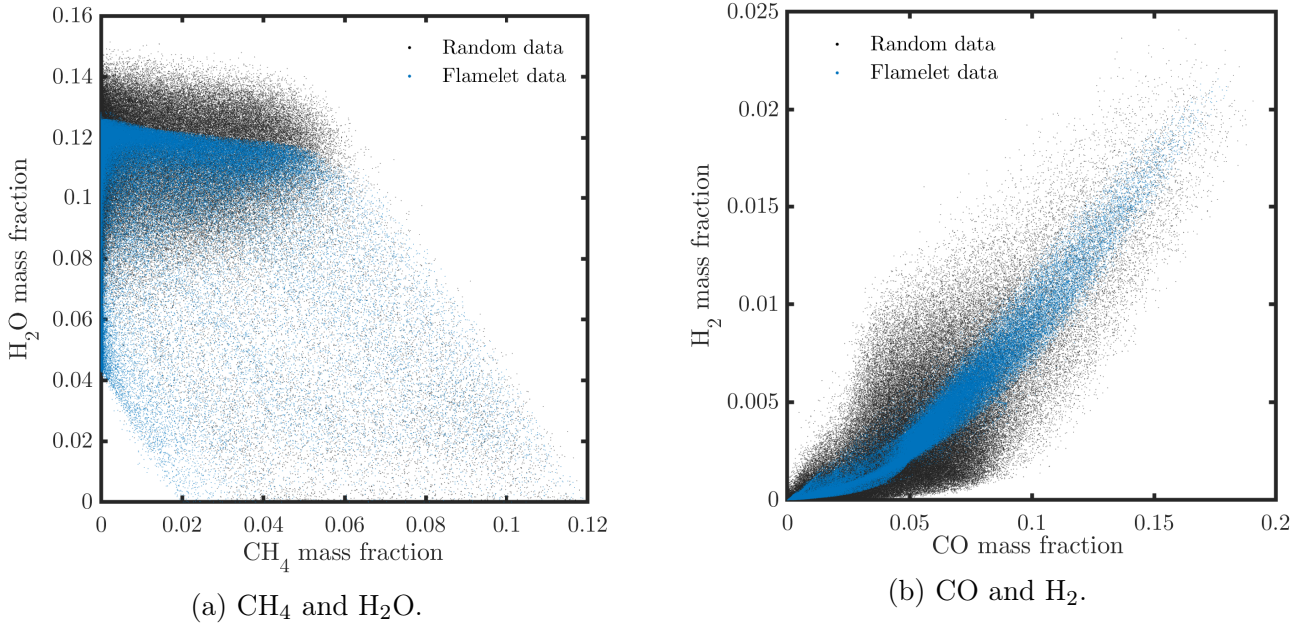


Figure 3.1: Comparison of composition space coverage of original 1D flamelet data and randomly generated data.

to equilibrium. The initial unburnt temperature of the flamelets was also varied between 200 K and 500 K, to generate composition states with an expanded enthalpy range. Data with a temperature below 500 K and a mixture fraction outside of the mixture fraction range $0.02 \leq z \leq 0.10$ were not collected due to low levels of reaction. Overall, approximately 250 000 data points were collected from the 1D laminar flamelets.

This data was then used as a basis for random data generation. The parameters used for the random data generation are given in Table 3.2. Figure 3.1 shows a comparison of the flamelet and random data for CH_4 vs H_2O mass fraction and CO vs H_2 mass fraction, showing that the random data occupies a slightly expanded area of the composition space while still remaining within the relevant bounds.

The randomly generated data was combined with the original flamelet data be-

fore being shuffled, to give an overall data set of approximately 500 000 composition points. These points were then integrated over a time step of 1×10^{-6} s to create the necessary input-change pairs for ANN training.

The next step in the ANN process is to select a suitable architecture to maximise accuracy in the application. The SOM-MLP method has been shown to give higher accuracy in the reserved part of the training set compared to using a single MLP for all of the species across the entire training set. However, the performance in the reserved portion of the training set is much less important than the performance in the application. Therefore, to get a true indication of performance, ANN architectures should be assessed on their accuracy over data representative of the application. It would be possible to achieve high accuracy in the reserved portion of the training set, but achieve low accuracy in the application by specialising to the training set and not having good general performance. To test the performance of several ANN architectures, approximately 500 000 data points were sampled from an LES-PDF simulation of Sydney flame L with stochastic fields, using DI of the reaction source term. Full details of the simulation from which data was taken are given in the following section. The hybrid random/flamelet data set was used for ANN training, with the points collected from the Sydney L simulation not included in the training process and only used to assess the performance of the trained ANN architectures.

Table 3.3 gives a summary of all of the tested ANN architectures. Architecture A was the best performing architecture from Franke et al. [55], consisting of a SOM of 100 neurons and an MLP predicting all of the species for each region. Architecture B was the same as architecture A, but used a single MLP to predict each species in each region. For the 30 species for which predictions are required in the GRI 1.2 mechanism, this means there were 3000 MLPs in total in architecture B. Architecture C applied a SOM of 4 neurons to the major species: CH_4 , O_2 , H_2O , CO , CO_2 and N_2 only, with a single MLP for each species trained on the resulting regions. Architecture D used a SOM of 4 neurons applied to all of the species, again training a single MLP for each species to each region. Finally, architecture E did not use the SOM at all, training a single MLP for each species across the whole composition space.

For all of the architectures, each of the input variables in the overall training set was scaled linearly between -1 and 1. For architecture E, the corresponding output variables were also scaled linearly between -1 and 1. For architectures A-D which used the SOM, after SOM training was performed on the scaled inputs, the input/change pairs in the resulting clusters were unscaled and then re-scaled to the maximum and minimum of each region so that the scaled variables for each region were between -1 and 1. In all of the tested architectures, the MLPs used took as

Table 3.3: Summary of tested ANN architectures and their respective mean relative RMS error over all predicted species.

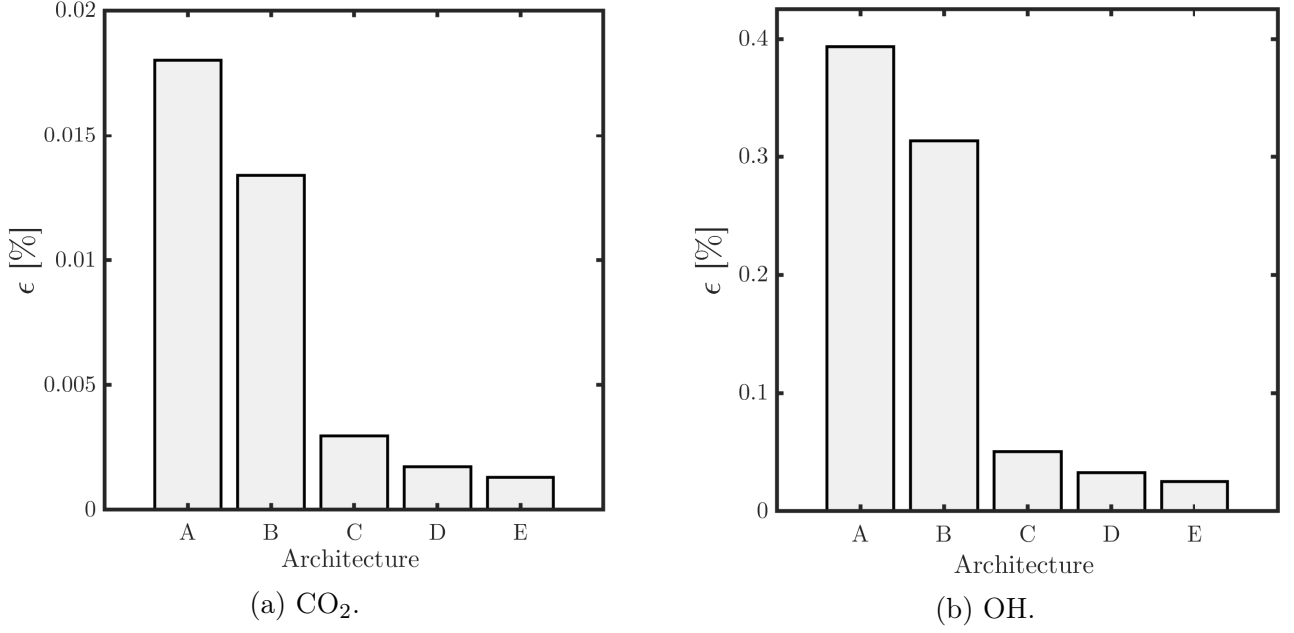
Architecture	SOM neurons	Number of MLPs	Other features	Mean ϵ [%]
A	100	100	-	0.202
B	100	3000	-	0.152
C	4	120	Major species only	0.063
D	4	120	-	0.061
E	-	30	-	0.059

input all thirty one species and the enthalpy, and contained a single hidden layer of thirty neurons. The number of outputs for each MLP depends on whether the MLP is predicting the thirty reactive species or a single species. Hyperbolic tangent activation functions were used on the hidden layers, with linear activation functions used on the output layers. Training was performed using the Levenberg-Marquardt algorithm, with the overall training set split 75%/25% into training/validation sets and six steps without improvement in the validation set used to determine the end of training for each network. After training, the Sydney L testing data was used, with all of the predictions then unscaled. The RMS error between the MLP predictions and the Sydney L targets for each of the i predicted species, ϵ_i , relative to the maximum value in the training data, was calculated using:

$$\epsilon_i = 100\% \times \frac{1}{\max(Y_i^{tr})} \times \sqrt{\frac{1}{N_{data}} \sum_{p=1}^{N_{data}} (s_i^p - s_i^{*,p})^2}, \quad (3.3)$$

where N_{data} is the number of data in the Sydney L testing set, $\max(Y_i^{train})$, the maximum of value of the species in the training data set, s_i^p the network target for the p th input point and $s_i^{*,p}$ the corresponding MLP prediction. Figure 3.2 show the errors for each architecture for CO₂ and OH, while Table 3.3 also shows the mean relative RMS error averaged over all of the species. It can be seen that architecture A has the worst accuracy in the Sydney L test set. Architecture B is significantly more accurate than architecture A, showing that separating the species and using a single MLP for each one leads to more effective training.

A significant increase in accuracy compared to architecture B is seen by using architectures C and D, which is a consequence of reducing the number of SOM neurons from 100 to 4. This is because using a large number of SOM neurons leads to a specialisation to the training data which does not translate to high accuracy


 Figure 3.2: Performance in CO₂ and OH using architectures A-E.

in the application. The relative absolute error across all of the species for a given point, ε^p , can be calculated using:

$$\varepsilon^p = \frac{1}{N_{sp} - 1} \sum_{i=1}^{N_{sp}-1} \frac{|s_i^p - s_i^{*,p}|}{\max(Y_i^{train})}, \quad (3.4)$$

where $|\cdot|$ is the absolute value. Figure 3.3 shows this error for each output prediction plotted against the Euclidean distance between the respective input point and the closest SOM neuron, with the training data shown in Figure 3.3a and the Sydney L testing data shown in Figure 3.3b. Figure 3.3a shows that for points close to the centre of their assigned cluster, the predictions are accurate, while the accuracy generally decreases slightly as the Euclidean distance from the centre increases. On the other hand, Figure 3.3b shows that even at small Euclidean distances the accuracy is lower with many points having a very high error. Additionally, the decrease in accuracy with increasing Euclidean distance is more obvious for the testing data.

Architectures C and D themselves showed very similar performance, meaning that the most meaningful regions of composition space are found in terms of the major species and regions of similarity in the minor species are not as relevant. The best performing architecture is E, which does not use the SOM at all. The key limitation of the SOM is that it identifies regions of similarity in the training data and trains ANNs to specialise to them. If lots of the data encountered in the application lies in other areas of the composition space around SOM neurons then,

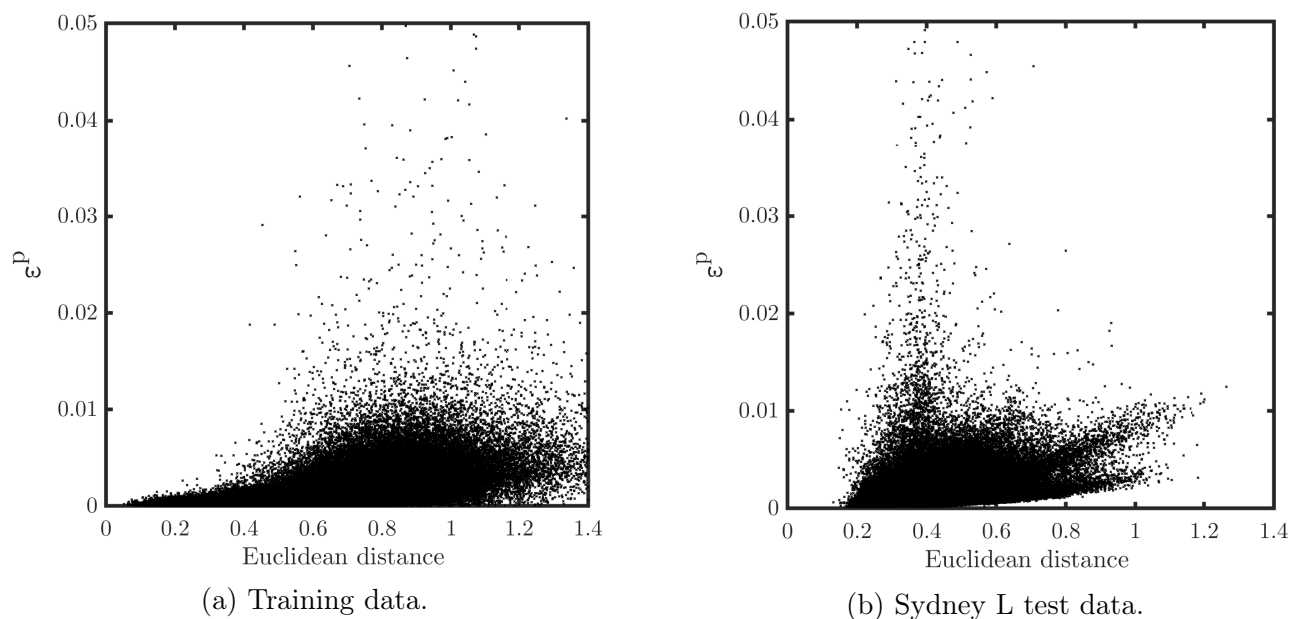


Figure 3.3: Prediction error in individual data points as a function of the Euclidean distance between the respective scaled input and the closest SOM neuron.

as Figure 3.3 shows, accuracy will suffer. By using a single MLP for each species training can be performed more effectively over the whole composition space and the resulting ANNs are more general, having not been forced to specialise to the SOM regions.

The aim of the random data generation is to add generality to the training data. To ensure that this added generality does not impede the performance of the ANNs in the original data, architecture E was validated by replacing reaction source term integrations in 1D CH_4 /air flamelet simulations with strain rates between 1 /s and 530 /s (just below the extinction strain rate). The simulations used a time step of 1×10^{-6} s, with the artificial pilot used to ignite the flames held for 1000 steps before the flames were simulated until steady state. Figure 3.4 shows the steady state 1D flamelet profiles from the simulations using ANNs, as well as comparative profiles from simulations using DI, for temperature and CO mass fraction.

Figure 3.4 shows that the accuracy of the ANNs in the 1D flamelet simulations is good, with slight differences in the profiles appearing as the strain rate decreases to 25 /s. These differences appear because at lower strain rates the effective rate of diffusion in the flamelets is lower. This means that the inevitable errors in the ANNs predictions are able to accumulate, and feed back into the inputs to the ANNs at following time steps. If the errors in predictions are too large relative to the diffusion, this accumulation of errors can cause simulations to become unstable. Figure 3.5 shows the steady state profiles for temperature and CO mass fraction at a strain rate of 1 /s. At this extreme strain rate, the rate of diffusion is very low,

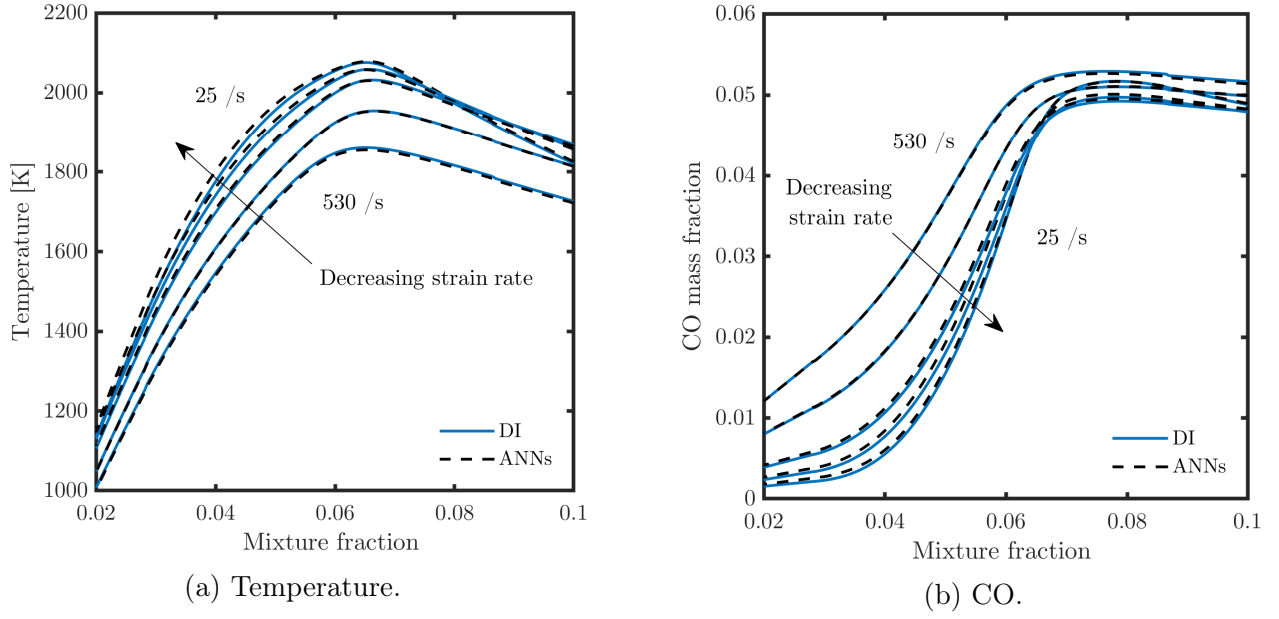


Figure 3.4: Steady state flamelet profiles for temperature and CO mass fraction at strain rates 25, 50, 100, 300 and 530 /s using DI and the architecture E ANNs.

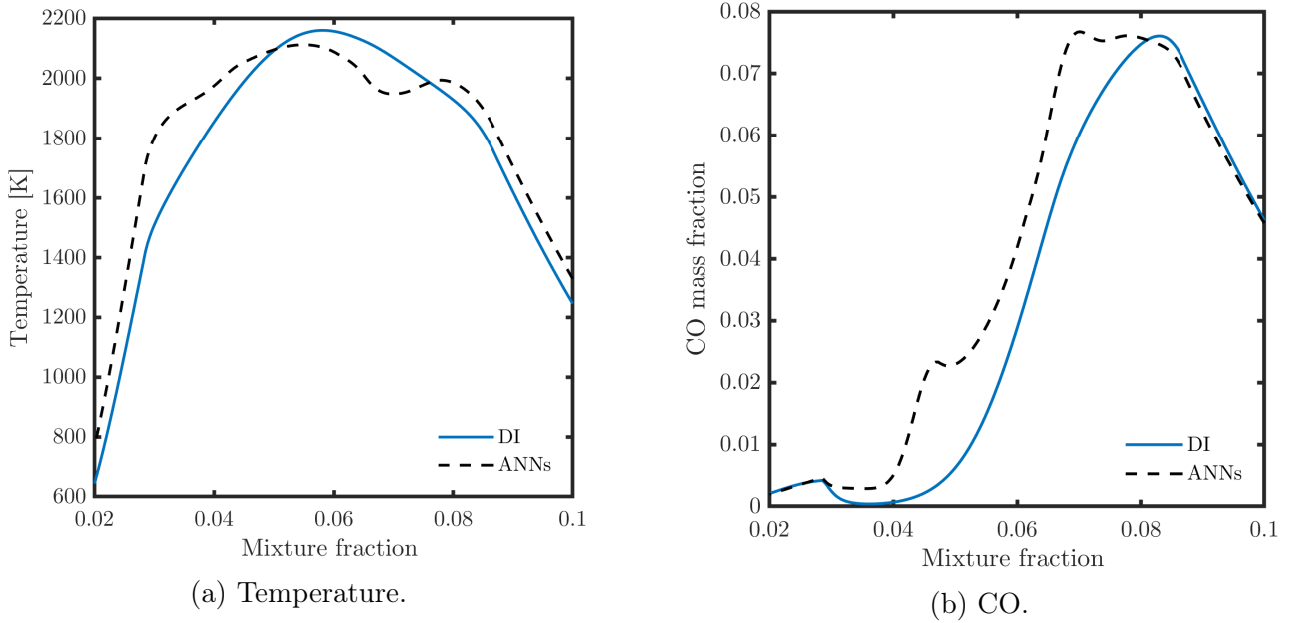


Figure 3.5: Steady state flamelet profiles for temperature and CO mass fraction at strain rate 1 /s using DI and the architecture E ANNs.

and the equilibrium pilot is practically conserved. Inaccuracies in the prediction of equilibrium states thus lead to a rapid accumulation of errors, and an obvious deviation of the ANN profile from the DI one. However, this flamelet simulation was run for 0.1 s, meaning 100 000 steps at 1×10^{-6} s. This means that although the errors in predicting chemical equilibrium have accumulated, errors at individual time steps are not so large as to make the simulation totally unstable. Such an extreme accumulation of errors does not occur in flame simulations, where fluid elements do not remain in the reaction zone indefinitely, although special attention should be given to the prediction of reaction behaviour around chemical equilibrium to prevent simulation instabilities. Following this analysis, the MLPs used in architecture E were applied to LES-PDF simulations of the Sydney L flame.

3.2 Application to the Sydney L flame

3.2.1 Experimental and simulation details

The non-premixed flame used as the test case is the Sydney L flame. The Sydney L flame is a piloted, non-premixed turbulent CH_4 /air jet flame, which was studied experimentally by Masri et al. [56]. The geometry of the burner exit is identical to the one used in the well-known Sandia flame series, with a fuel jet of diameter $D = 7.2$ mm surrounded by an annular pilot of diameter $D_P = 18.2$ mm. The pilot is itself surrounded by an air co-flow. The fuel jet is exclusively CH_4 , while the pilot used in the experiment contains a fully burnt mixture of $\text{C}_2\text{H}_2/\text{H}_2$ /air, chosen to give the same element mass fractions as a stoichiometric mixture of CH_4 /air. The bulk velocities of the fuel jet and air co-flow were 41 m/s and 15 m/s respectively, with both having an inlet temperature of 293 K. The bulk velocity of the pilot was not measured, but was assumed in the experiment to be 24 m/s based on the known unburnt pilot bulk velocity and a fully burnt pilot mixture temperature of 2600 K. Means and fluctuations of species mass fractions were calculated from Raman-Rayleigh measurements of number densities across the radius at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$.

Previous LES studies of Sydney flame L have been performed. Prasad et al. [9] employed the LES-PDF method with 8 stochastic fields in a simulation using a very fine body fitted grid of approximately 7.2 million cells. A reduced version of the GRI 3.0 mechanism [102] consisting of 19 species was used. Good agreement between the simulation and experiment was seen between mean and RMS radial profiles at all axial locations for temperature, mixture fraction, and CO_2 and H_2 mass fraction. Wang et al. [103] also employed the LES-PDF method, but used a solution scheme

based on Lagrangian stochastic particles, using approximately 22 million such particles. Wang et al. [103] used the GRI 1.2 mechanism consisting of 31 species, with ISAT used to accelerate reaction source term computations. Reasonable agreement was seen between the mean profiles from the simulation and the experimental values, although the flame spreading rate was over predicted at $x/D = 20$ and $x/D = 30$, leading to significant differences in the profiles of temperature and O_2 and CO_2 mass fraction. Additionally, the peak values of CO were significantly under predicted. Franke et al. [55] also performed LES-PDF simulations using the stochastic field method with 8 stochastic fields and an RCCE reduced version of the GRI 1.2 mechanism consisting of 16 species. A grid of approximately 2.75 million cells was used, with good agreement observed at $x/D = 10$ between the mean and RMS radial profiles from the simulation and the experimental data. At higher axial locations the flame spread was over predicted while the amount of local extinction was under predicted, leading to significant over predictions of temperature, CO_2 and H_2O at $x/D = 20$ and $x/D = 30$.

To perform simulations of the Sydney L flame in this work, a rectangular physical domain of 250 mm(axial) $\times$ 100 mm $\times$ 100 mm was represented using a Cartesian finite volume grid of approximately 3.2×10^6 cells arranged $224 \times 120 \times 120$. Cell widths of around 0.4 mm were used near the burner exit, with a smooth radial expansion. Figure 3.6 shows the radial and axial grid spacing at $x/D = 0$ and $r/D = 0$ respectively. The overall domain was split into 288 sub-domains, with a CPU assigned to perform the computations for each one. Unsteady velocity profiles for the jet, pilot and air co-flow inlets were generated based on the experimental measurements using the digital turbulence method of Di Mare et al. [104] and applied at the inlet plane. The GRI 1.2 mechanism for CH_4 was used, consisting of 31 species and 175 reactions. The temperatures and compositions of the jet and air co-flow were set to their respective experimental values. Using a pilot temperature of 2600 K was found to lead to very large over predictions of the temperature compared to the experimental values at $x/D = 10$, so the temperature of the pilot was set to 2200 K, although the bulk velocity of 24 m/s was still used. The pilot composition was set to that obtained from an equilibrium calculation of a stoichiometric CH_4 /air mixture.

Two simulations of the Sydney L flame were performed: one using ANNs and one using direct integration (DI) with the VODE solver [105]. The flames were simulated using the LES-PDF method with stochastic fields as described in Chapter 2.2, with eight stochastic fields used. A constant time step of 1×10^{-6} s was used throughout both simulations. The simulations were performed for three flow through times, based on the jet bulk velocity, before collecting statistics over eight flow through times. In contrast to Franke et al. [55], the simulation with ANNs used ANNs exclusively, rather than using DI to reach statistical steady state before taking

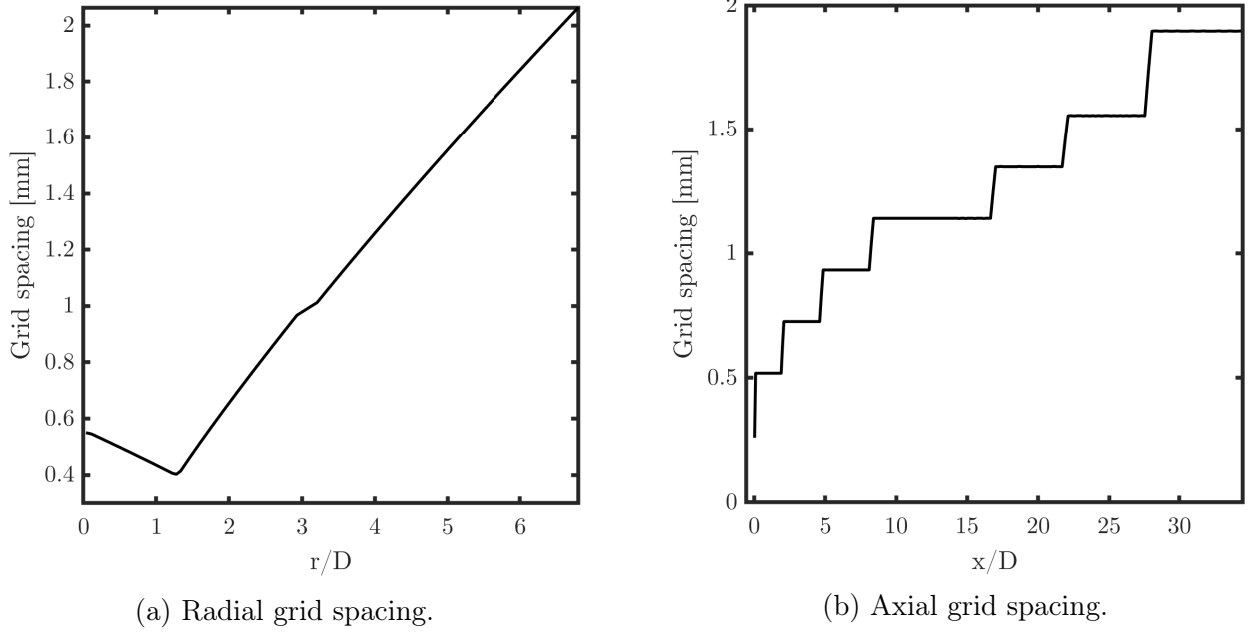


Figure 3.6: Radial and axial grid spacing at $x/D = 0$ and $r/D = 0$ respectively for the grid used in the Sydney L simulations.

statistics with ANNs.

3.2.2 Sydney L flame simulation results

Figures 3.7 and 3.8 show mean and rms radial profiles at all three axial locations for temperature, mixture fraction and the major species for which experimental data is available, using both reaction source term integration methods. It can be seen in Figures 3.7 and 3.8 that the DI and ANN profiles are practically coincident, with very minor discrepancies at the higher axial positions for CO and H₂. Although exactly replicating the experimental results is not the purpose of these simulations, reasonable agreement is seen between the experiment and the simulations at $x/D = 10$. As in several previous studies, extinction at $x/D = 20$ is not well captured in the simulations, leading to a significant over prediction of combustion compared to the experiment at $x/D = 30$.

Figures 3.9 - 3.10 show mean and rms radial profiles at all three axial locations for OH, H, O, H₂O₂, CH₃, C₂H₂, C₂H₄ and C₂H₆. With the exception of H₂O₂ and C₂H₆, Figures 3.9 - 3.10 show that agreement between the ANN and DI simulations is good, with the shape and peak of the radial profiles at all three axial locations successfully reproduced for the considered species. In Figures 3.9 - 3.10 it can also be seen that agreement tends to slightly deteriorate at higher axial positions, as the error in ANN predictions accumulate as fluid particles move downstream. Figure 3.11 shows the radial profiles for HO₂, CH₃OH, HCCO and HCCOH. Here, as

Table 3.4: Average time taken for reaction source term computations using ANNs and DI, normalised to the time taken for reaction source term computations using DI.

Method	Reaction time	Total time	Reaction/Total [%]
DI	1.000	1.411	70.9
ANN	0.056	0.467	12.0

with H_2O_2 and C_2H_6 , there are more obvious differences between the ANN and DI simulations, although the agreement is still reasonable. This shows that even though an approach focused on the general predictive capability of the ANNs is much more accurate than one focused on specialisation through clustering, there is still scope for methods which increase ANN accuracy and allow for more accurate predictions of minor species. Although all of the ANNs used in the ANN simulations shared the same architecture, because a single MLP is used for each species it is possible that different sized ANNs could be used for species which are harder to predict to increase accuracy.

Figures 3.7 - 3.11 show that using separate ANNs for each species has allowed the weights to be used very effectively, with a small number of weights producing high accuracy. The total number of weights contained in the ANNs used was 30 630 for a mechanism of 31 species. In contrast, Franke et al. [55] used approximately 750 000 weights for a 16 species mechanism, An et al. [36] used approximately 4 500 000 weights for a 41 species mechanism and Wan et al. [61] used approximately 180 000 weights for a mechanism of 11 species. Using a smaller number of weights also generally means a reduction in the number of floating point operations which must be performed for ANN retrievals, and so the potential speed benefit when using ANNs is larger. Table 3.4 shows the time taken to complete reaction source term computations as well as the total time for one step of the Sydney L simulations, averaged over the entire simulation and normalised to the time taken to integrate the reaction source terms using DI.

Table 3.4 shows that when using DI the majority of the total simulation time is taken by reaction source term computations. In contrast, when using the ANNs the time taken to perform reaction source term computations reduces approximately 18-fold. This means that the chemical reaction step is no longer the bottleneck in the simulations, and only takes up 12% of the required CPU time.

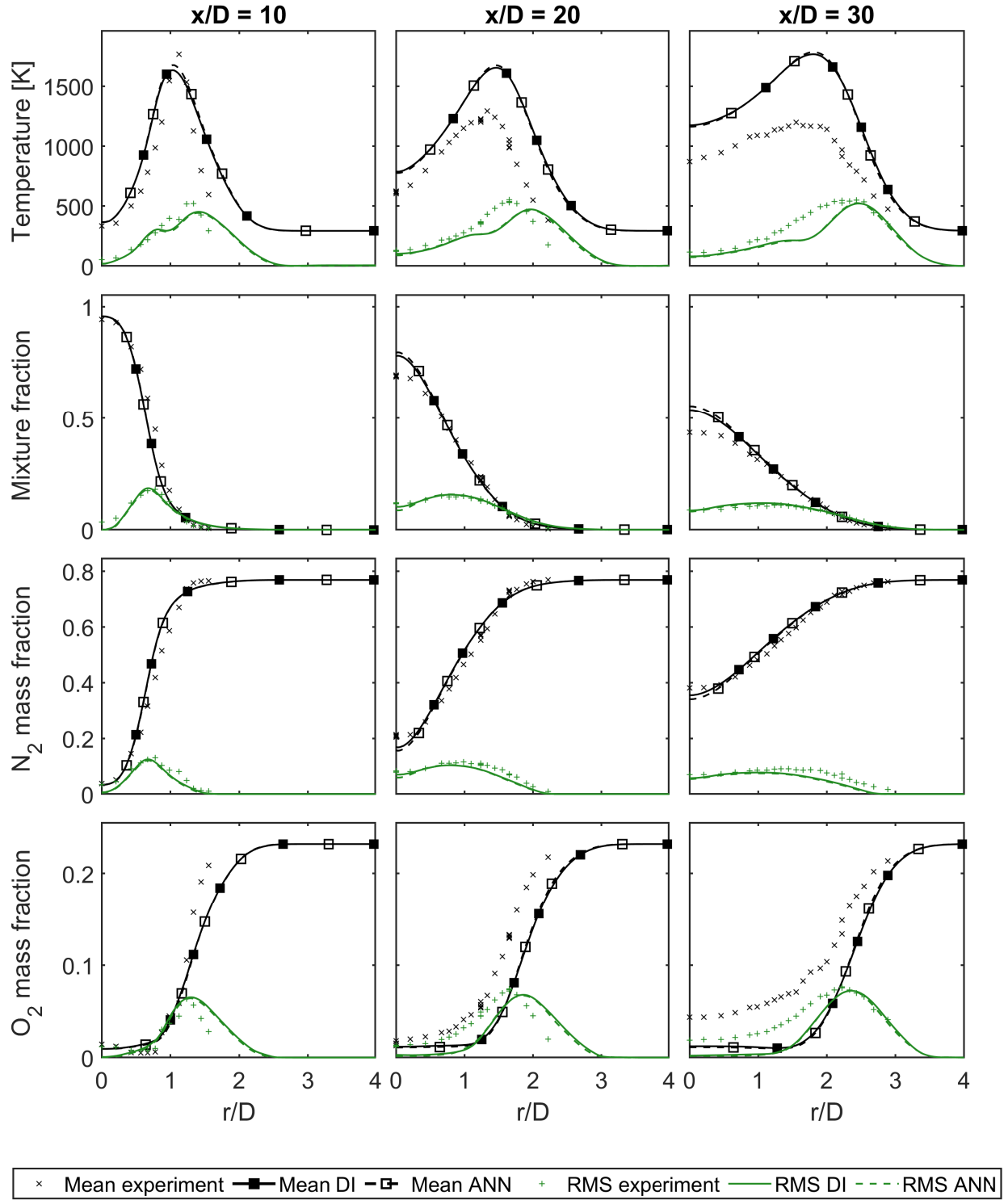


Figure 3.7: Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI, as well as experimental values, for temperature, mixture fraction and N_2 and O_2 mass fraction.

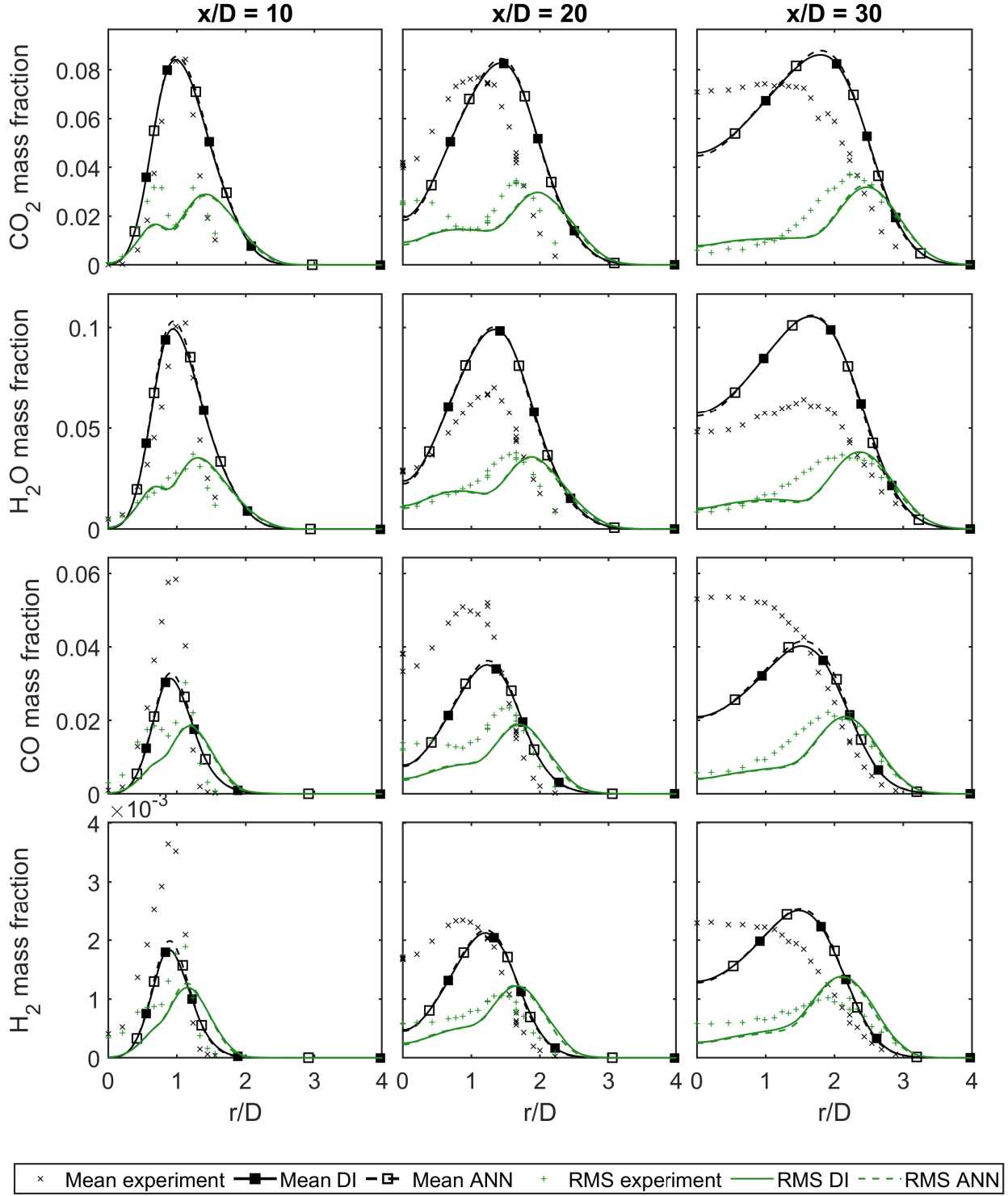


Figure 3.8: Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI, as well as experimental values, for CO₂, H₂O, CO and H₂ mass fraction.

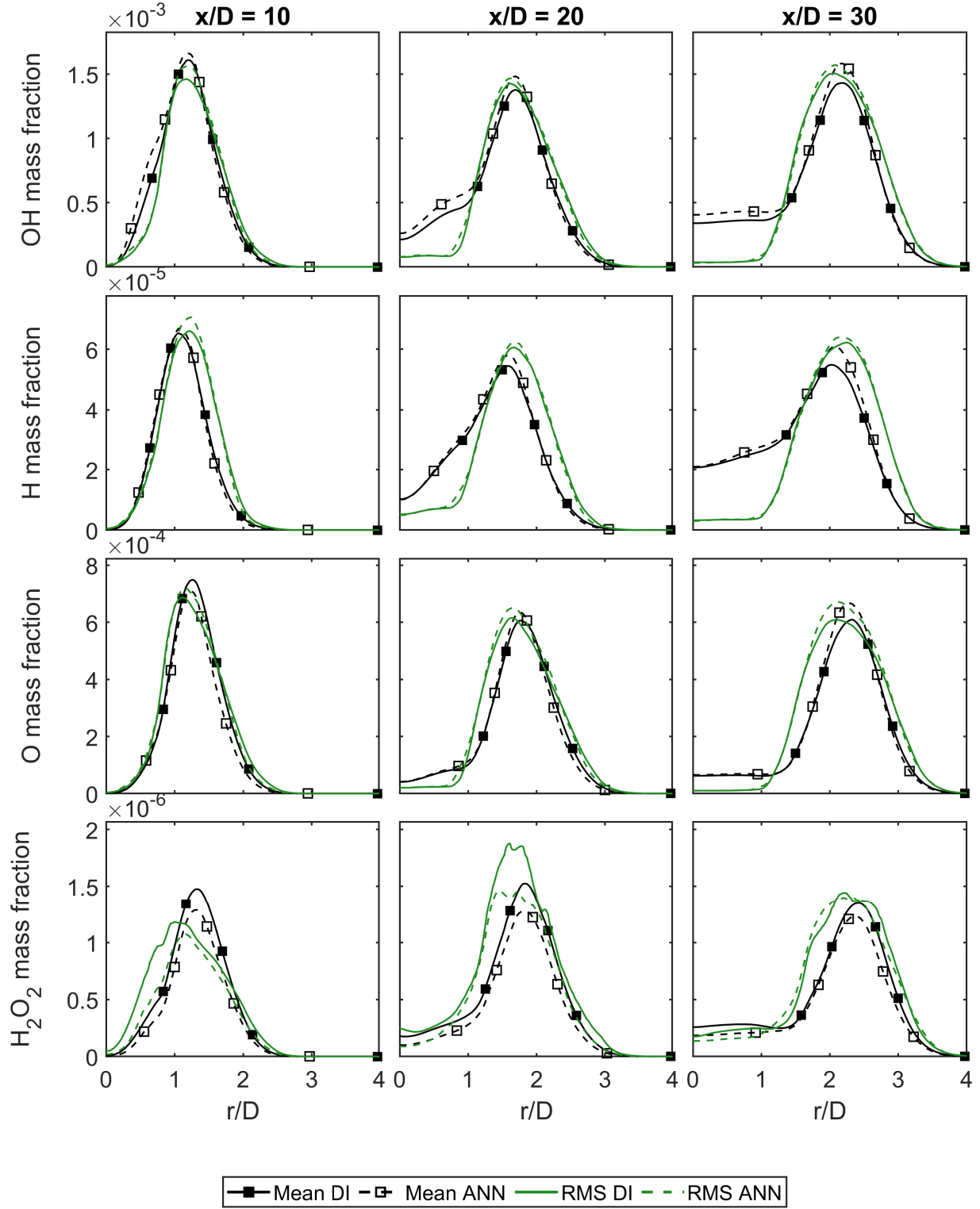


Figure 3.9: Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for OH, H, O and H₂O₂ mass fraction.

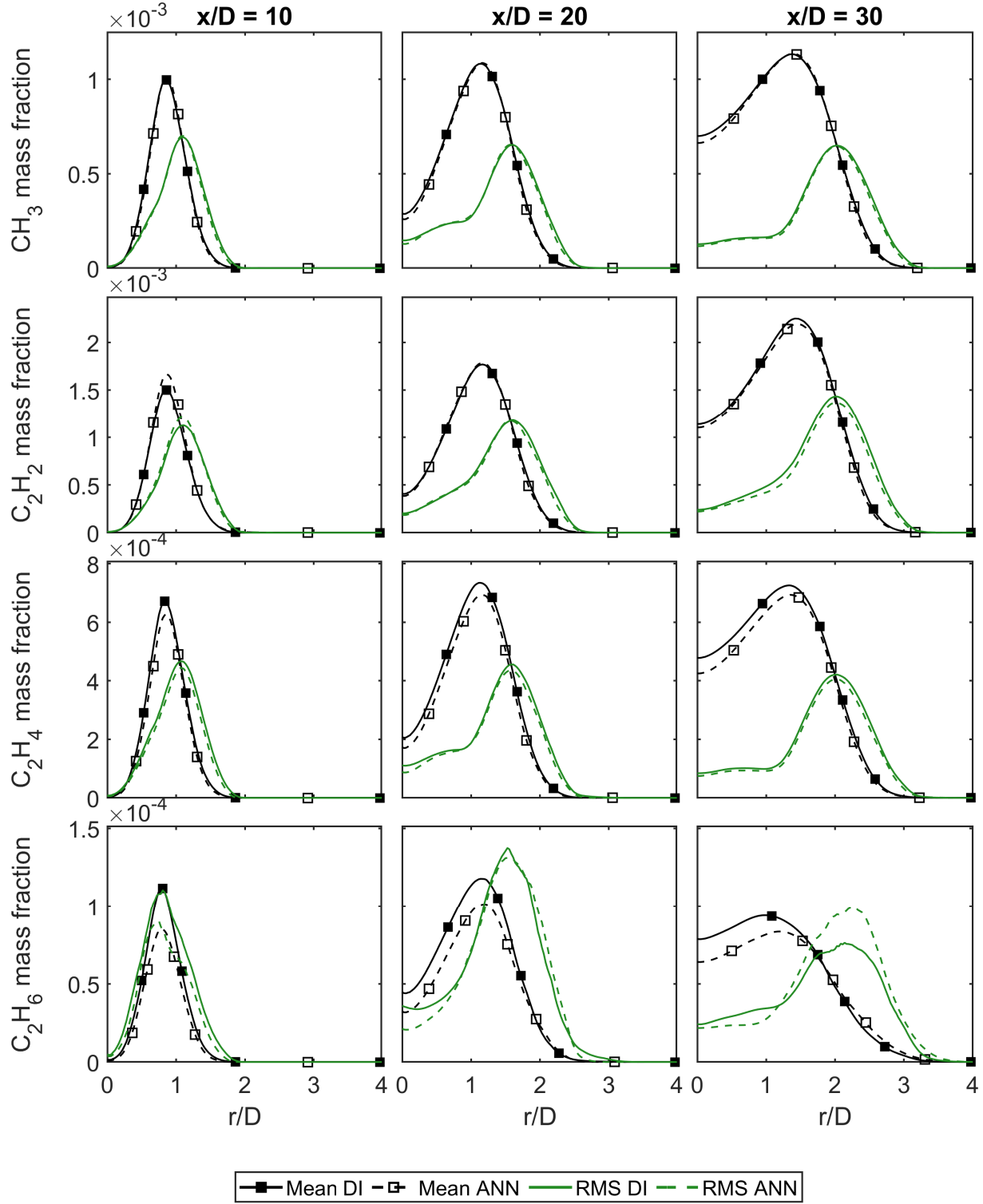


Figure 3.10: Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for CH₃, C₂H₂, C₂H₄ and C₂H₆ mass fraction.

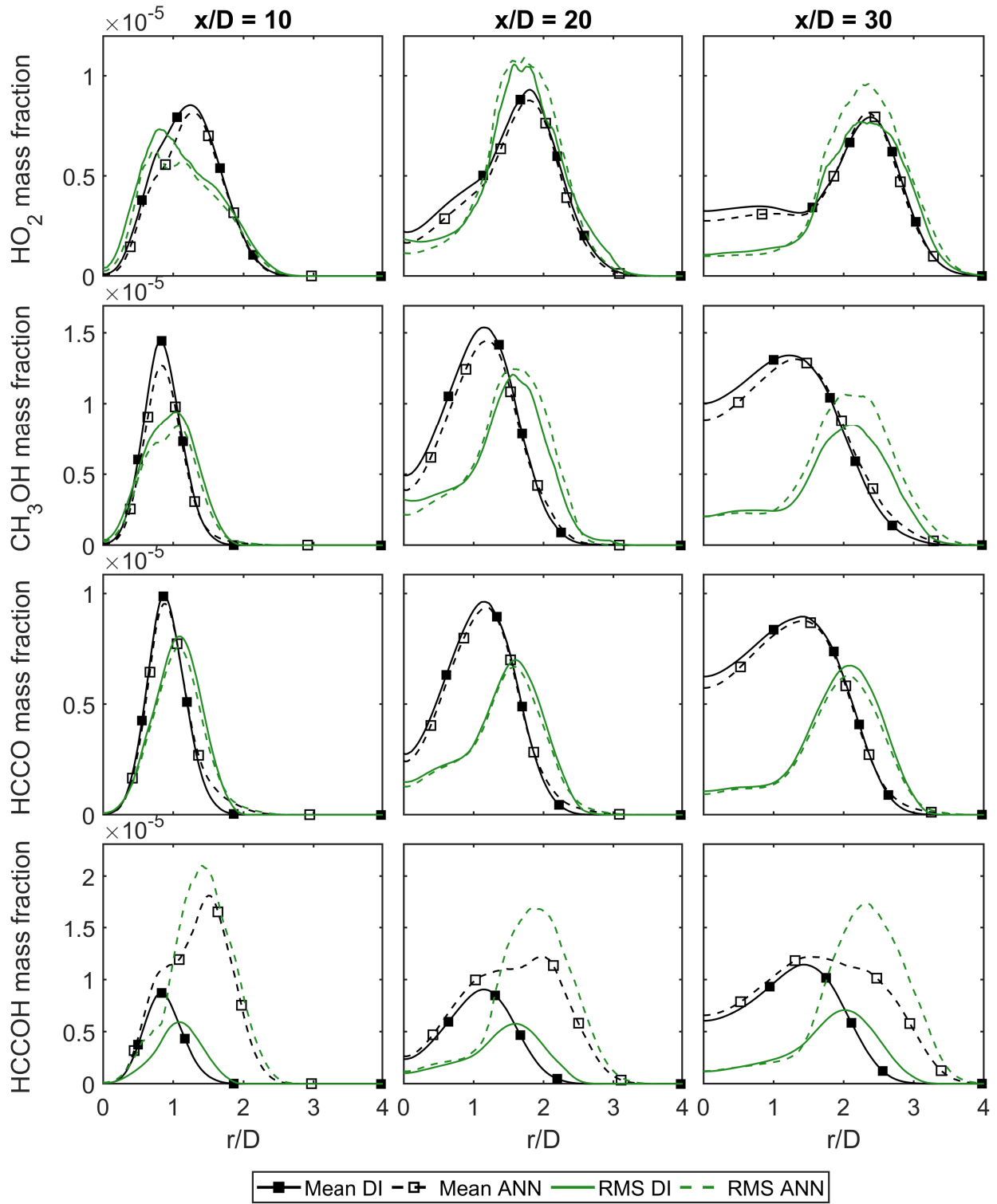


Figure 3.11: Radial mean and rms profiles at axial locations $x/D = 10$, $x/D = 20$ and $x/D = 30$ for simulations using ANNs and DI for HO_2 , CH_3OH , HCCO and HCCOH mass fraction.

Chapter 4

Development of a machine learning methodology for premixed flames

Chapter 3 showed that ANNs trained on randomly generated data and focused on generalisation rather than specialisation are capable of simulating turbulent non-premixed flames. For non-premixed flames, the choice of canonical combustion problem is quite straightforward, with the unsteady flamelet model well used. This is also true of partially premixed flames which behave mostly as non-premixed flames, such as the Sandia flames D-F. However, for other combustion regimes, the choice of canonical combustion problem becomes less obvious. This may explain the limited application of ANNs to premixed turbulent flames. In this chapter, the thermochemistry tabulation method is extended to premixed flames, with the Cambridge/Sandia swirl burner used as a test case.

The work in this chapter has been submitted for publication in Readshaw et al. [106].

4.1 Generating training data for premixed flames

To generate training data for premixed flame applications, the canonical problem of 1D premixed flames is used. Rather than being 1D in mixture fraction space, such flames are 1D in physical space, with a constant equivalence ratio/mixture fraction across the domain and are governed by Equation 4.1:

$$\rho \frac{\partial Y_i}{\partial t} + \frac{\partial \rho(u + u_c)Y_i}{\partial x} = \frac{\partial}{\partial x} \left(\rho \mathcal{D}_i \frac{M_i^r}{M_{av}^r} \frac{\partial X_i}{\partial x} \right) + \dot{\omega}_i, \quad (4.1)$$

where u is the velocity, x the spatial position, ρ the density, $\mathcal{D}_i$ the diffusivity and

M_{av}^r the mean molecular weight. u_c is a correction velocity used to ensure mass conservation [107] and is given by:

$$u_c = \sum_{i=1}^{N_{sp}} \mathcal{D}_i \frac{M_i^r}{M_{av}^r} \frac{\partial X_i}{\partial x}. \quad (4.2)$$

The 1D premixed flames are initialised by setting half of the domain to equilibrium conditions and the other half to unburnt conditions. By sampling many equivalence ratios within the given fuel and oxidiser flammability limits, a data set consisting of a wide range of premixed conditions can be collected. Training data collection occurs in an analogous manner to the 1D flamelets, with Equation 4.1 discretised and then simulated with a fractional step method. Data is collected at randomly selected points in physical space across the domain, interpolating between the two nearest known points if necessary. Again, a minimum sampling temperature can be set to prevent large amounts of sampling from the unburnt section of the domain. As with the laminar flamelets, once the flame reaches steady state the simulation is stopped. The variables which can be altered to change the sampled composition space from 1D premixed flames are given in Table 4.1. Again, the 1D premixed flame variables should be matched to the application as closely as possible. Further, if non-equal diffusivities are used in the application they should be used in the 1D premixed flames. Likewise if equal diffusivities are to be used in the application then it does not make sense to include differential diffusion in the CCPs.

Table 4.1: Variables which can be altered in 1D premixed flame simulations to change the sampled composition space.

1D Premixed Flame Variable
Unburnt temperature
Pressure
Equivalence ratio
Time step
Samples per time step
Simulation length
Minimum sampling temperature

To generate data to simulate premixed turbulent flames, 1D premixed flame simulations were used in conjunction with the random data generation method. In an analogous way to the 1D flamelets used in the previous chapter, data was sampled from an ensemble of 1D laminar premixed flames. However, the use of 1D premixed flames presents an issue. The 1D laminar flamelets used for non-premixed flames are continuous in mixture fraction. With equal species diffusivities, the mixture fraction

and enthalpy are linearly related, so the 1D flamelets are also continuous in enthalpy. This is in contrast to 1D premixed flames with equal species diffusivities, where a single value of equivalence ratio corresponds to a single value of mixture fraction and enthalpy. This means that short of simulating several thousand premixed flames, it is difficult to get smooth coverage in enthalpy by using the 1D premixed flame data alone. This is problematic for ANN training, as the resulting ANNs will be naturally biased to the sample equivalence ratios.

This problem was overcome using the random data generation method. One hundred 1D premixed flames were simulated using a fixed time step of 1×10^{-6} s, with the equivalence ratio varied linearly between $0.45 \leq \phi \leq 1.70$. Table 4.2 gives the premixed flame simulation details. By sampling randomly from across the physical domain, 500 000 data points were collected. Figure 4.1a shows these points, with obvious vertical lines at the sampled equivalence ratios. This data was used as a basis for random data generation, with Table 4.3 giving the parameters used for the random data generation. Figure 4.1b shows the resultant random data, where the sampling deficiency is no longer evident and it is clear that the coverage in enthalpy is much smoother. The original premixed flame data was then discarded, with the random data integrated using the VODE solver over a time step of 1×10^{-6} s to form the input/change pairs.

Table 4.2: 1D premixed flame simulation information.

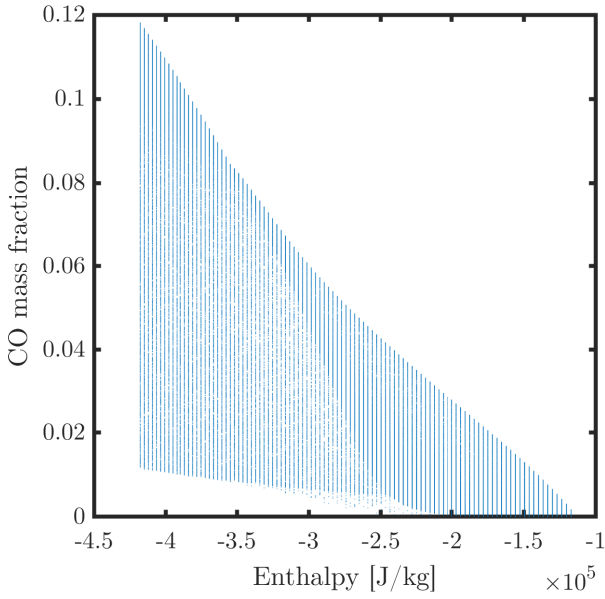
Number of flames	100
Equivalence ratio range	0.45 - 1.70
Domain length	0.01m
Grid nodes	500
Unburnt temperature	300 K
Time step	1×10^{-6} s
Simulation length	0.005s
Minimum reactive temperature	500K

4.2 Artificial neural network training for premixed flames

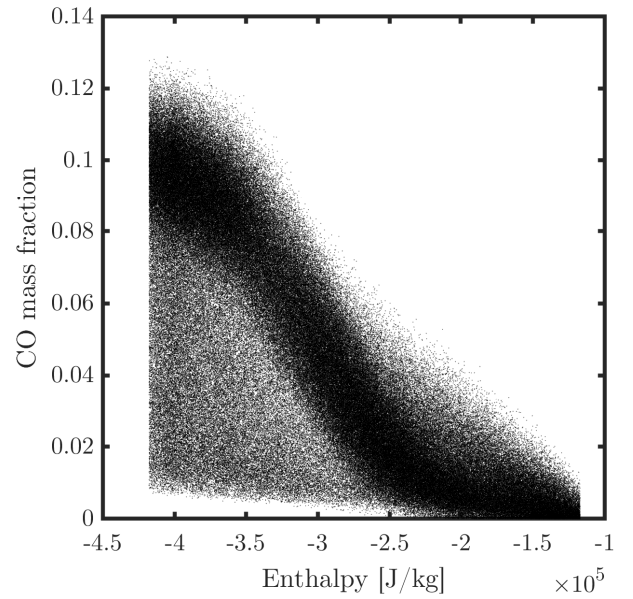
From the results of Chapter 3, no clustering of the composition space was performed and instead the species were predicted individually. However, to improve ANN accuracy compared to the simulation of the Sydney L flame, the multiple MLP (MMLP) method of Ding et al. [58] was used. The MMLP method achieves higher

Table 4.3: Parameters used for random data generation on the 1D premixed flame data.

Parameter	Value
b_1	8
b_2	8
b_3	10
Molar H/C window	3.8 - 4.2
Molar O/N window	0.254 - 0.274
ϕ window	0.45 - 1.70



(a) 1D premixed flame data.



(b) Randomly generated data.

Figure 4.1: Comparison of composition space coverage of original 1D premixed data and randomly generated data, showing vertical lines at constant enthalpy values in the premixed data.

accuracy by further dividing the data for each species based on the magnitude of the change in the input/change pair. In combustion applications, the magnitude of the change in a species over a given time step can vary significantly depending on the local combustion conditions. For example, for igniting compositions the change in some species can be very large, while for points near equilibrium the changes can be very small. With a loss function such as the mean squared error, for a fixed relative prediction error, the greatest absolute error (and so contribution to the loss function) comes from the points with the highest magnitudes. Therefore, during training the points with the highest output magnitude are naturally targeted and end up with a lower relative error, at the expense of points with a low output magnitude. Although the loss function could be weighted to enhance predictions

in points with lower output magnitudes, the MMLP method has been shown by Ding et al. [58] to outperform approaches using a weighted loss function, as well as approaches using non-linear variable scaling methods. In the MMLP method, to increase accuracy in points across the range of output magnitudes, the training data are split depending on the magnitude of their change, with separate MLPs trained for each species and each output range. This splitting is motivated by knowledge of combustion processes, but is based purely on the magnitude of the changes and does not use combustion knowledge directly, meaning it does not have to be tuned for different problems.

In contrast to the method of Ding et al. [58], where a manual selection of the output magnitudes was required, for the SWB flames the division of the training data was performed automatically, choosing three divisions for each considered species. First, the maximum absolute change of each species was calculated in the random data set. The exponent of this value, e , was recorded. The first subdivision contains all data with an output magnitude $10^{e-1} < |\Delta Y_i|$. The second subdivision contains all data with an output magnitude in the range $10^{e-2} < |\Delta Y_i| \leq 10^{e-1}$. This continued until all of the data was divided. With three subdivisions the final data set contains all data with an output magnitude in the range $10^{e-3} < |\Delta Y_i| \leq 10^{e-2}$. Table 4.4 shows the maximum mass fraction of H_2 in the random data, as well as the respective boundaries for division into three data sets, while Figure 4.2 shows the three divisions of the random data for H_2 according to the boundaries in Table 4.4.

Table 4.4: Maximum change in H_2 mass fraction over the randomly generated data, as well as the boundaries for division into data sets for MMLP training.

Species	Maximum $ \Delta Y $
H_2	3.20×10^{-4}
Data set	Output range
1	$10^{-5} < \Delta Y $
2	$10^{-6} < \Delta Y \leq 10^{-5}$
3	$10^{-7} < \Delta Y \leq 10^{-6}$

The GRI 1.2 mechanism was used, and as the random data was divided separately into three divisions for each species, there were 90 individual data sets in total. Once the random data was divided, the MLPs were trained. Each of the MLPs used had two hidden layers of thirty neurons each, with hyperbolic tangent activation functions on the hidden layers and linear activation functions on the output layer. The Levenberg-Marquardt algorithm with Bayesian regularisation was used for training the ANNs, with 1000 training iterations performed. At run time,

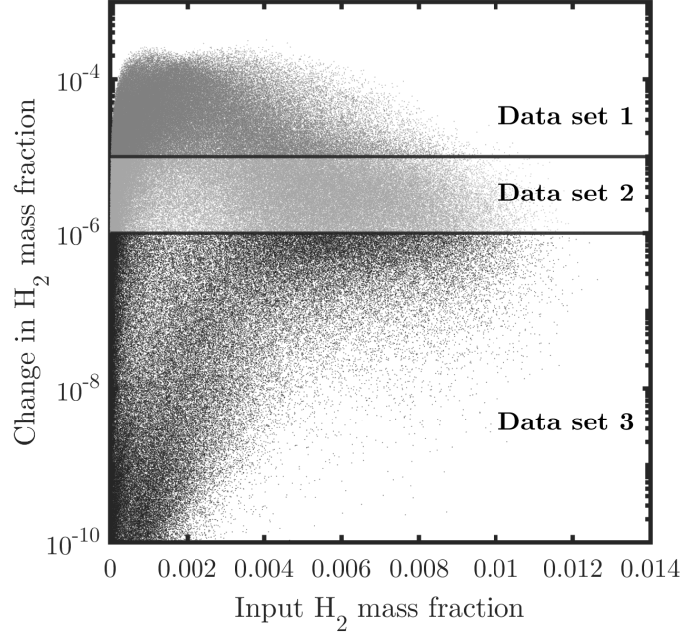


Figure 4.2: Randomly generated data divided into three data sets, based on the boundaries in Table 4.4.

the magnitude of the change in a given composition is unknown, as that is what is predicted by the ANNs. Therefore it is also unknown which MLP should be used for a given input point. To overcome this issue, the 'reverse' process described in Ding et al. [58] was used, where a given composition goes to the MLP with the smallest output range first. If the prediction by this MLP is outside of its own range then the point passes to the MLP with the next smallest range. This continues until the prediction is within the range of the MLP, or the MLP with the largest output range is reached.

4.3 Application to the Cambridge/Sandia swirl burner

4.3.1 Experimental and simulation details

The Cambridge/Sandia swirl burner (SWB) was studied experimentally by Sweeney et al. [108, 109] and Zhou et al. [110], who recorded velocity, temperature and species mass fraction measurements at various axial locations above the burner for a number of different flame configurations. The burner itself consists of a central ceramic bluff body of diameter $D = 12.7$ mm surrounded by two concentric fuel annuli, with outer diameters of $D = 25.4$ mm and $D = 38.1$ mm respectively. The inner annulus supplies premixed CH₄/air at a given equivalence ratio with a bulk axial velocity of $U_i = 8.1$ m/s and a temperature of 300 K, while the outer annulus

also supplies premixed CH_4/air at a given equivalence ratio with a temperature of 300 K and a bulk velocity of $U_o = 18.7 \text{ m/s}$, which can be split between axial and tangential to vary the swirl flow ratio (SFR). The Reynolds numbers based on these velocities and the annuli exits are $Re_i = 5960$ and $Re_o = 11500$ for the inner and outer annuli respectively. The fuel jets are themselves surrounded by an outer low velocity air co-flow of axial velocity $U_{cf} = 0.4 \text{ m/s}$ and temperature 300 K. Two flame configurations were considered: configurations five (SWB5) and six (SWB6), which both feature stratification. SWB6 includes a swirling outer fuel jet, with the operating conditions for SWB5 and SWB6 summarised in Table 4.5.

Table 4.5: Operating conditions for SWB5 and SWB6.

Configuration	ϕ_i	ϕ_o	SFR
SWB5	1.0	0.5	0.00
SWB6	1.0	0.5	0.25

Several previous LES studies of SWB5 have been conducted, while SWB6 has been less well examined. Nambully et al. [111] and Proch and Kempf. [112] both performed simulations of SWB5 using the GRI 3.0 mechanism [102] with fine grids of 50 million and 400 million cells used by Nambully et al. [111] and 100 million cells used by Proch and Kempf. [112]. Nambully et al. [111] used a filtered laminar flame PDF approach for the closure of the chemical source terms, with good agreement observed between mean and RMS radial profiles from the simulations and the experimental values for ϕ , temperature, CH_4 and CO_2 mass fractions. Proch and Kempf. [112] used an artificial flame thickening model with flamelet generated manifold look up tables for chemical source term modelling. Reasonable agreement was seen between the experiment and simulation mean and RMS radial profiles for ϕ , temperature, CH_4 , CO_2 , H_2O and O_2 mass fractions, with a slight worsening of agreement seen at higher axial locations. Both Nambully et al. [111] and Proch and Kempf. [112] reported an over prediction of the temperature in the recirculation zone directly above the bluff body, while Proch and Kempf. [112] also reported an under prediction of the equivalence ratio in this region. This under prediction of equivalence ratio was not observed by Nambully et al. [111] due to the inclusion of differential diffusion in their model which accounted for the accumulation of carbon above the bluff body. Additionally, Nambully et al. [111] and Proch and Kempf. [112] observed an under prediction of CO mass fraction peak values at all axial locations, while Proch and Kempf. [112] also reported an under prediction of H_2 peak values at all axial locations.

SWB5 simulations have also been performed using the LES-PDF method, with Turkeri et al. [113] using a Lagrangian stochastic particle solution scheme with 46 million stochastic particles, and Brauner et al. [84] using the stochastic fields method

with 8 stochastic fields. Turkeri et al. [113] used the 16 species ARM1 reduced mechanism in conjunction with a grid of 2.3 million cells and ISAT for chemical mechanism tabulation. Excellent agreement was seen between the mean radial profiles from the simulation and from experiment, with good agreement in the RMS radial profiles for temperature, equivalence ratio, CH_4 , CO_2 and O_2 mass fraction. Brauner et al. [84] used a reduced version of GRI 3.0 with 19 species and a grid of 4 million cells. Again, the mean and RMS radial profiles for temperature, CH_4 , O_2 and CO_2 from the experiment were well captured by the simulation, although the flame spreading rate was slightly over predicted. As in the studies by Nambully et al. [111] and Proch and Kempf. [112], Turkeri et al. [113] and Brauner et al. [84] both reported an over prediction of temperature directly above the bluff body, while Turkeri et al. [113] also reported an under prediction of the equivalence ratio in this region and an under prediction of CO mass fraction peak values.

To simulate SWB5 and SWB6, a cylindrical domain of axial length 250 mm and radius 100 mm was represented with a structured curvilinear mesh of approximately 2.28 million cells, which was split into 216 subdomains. Cell widths were approximately 0.35 mm immediately above the bluff body, with a smooth radial and axial expansion, with the grid spacing shown in Figure 4.3. At the inlet plane, adiabatic, Neumann, no-slip conditions with a wall function were prescribed at the bluff body, with the bulk velocities of the two premixed CH_4 /air jets and the co-flow matched to the experimental values. A zero-gradient boundary condition was applied over the outlet plane, with free-slip conditions applied over the cylindrical boundary. Unsteady velocity profiles were generated using the digital turbulence method of Di Mare et al. [104] and applied to the inlet streams, with the profiles based on the experimental velocity measurements taken from 2 mm above the bluff body. A constant time step of 1×10^{-6} s was used throughout all SWB simulations. As with the simulation of Sydney flame L, the GRI 1.2 mechanism consisting of 31 species was used.

As the SWB flames are non-piloted, they must be ignited. In the simulations, to fill the domain with flammable fuel/air mixture, the flames were run with a single stochastic field without reaction for two flow through times of the domain based on the bulk velocity of the inner fuel jet, which is approximately 60 000 steps at the selected time step. The flames were then ignited by setting all compositions within the CH_4 /air flammability limits to equilibrium for a single time step only. After this point, the flames were run with finite rate chemistry using the desired reaction source term solver for another two flow through times. Then, the flames were switched to eight stochastic fields and allowed to settle, before statistics were collected over another two flow through times.

To test the effectiveness of the ANN method in premixed flames, two simulations

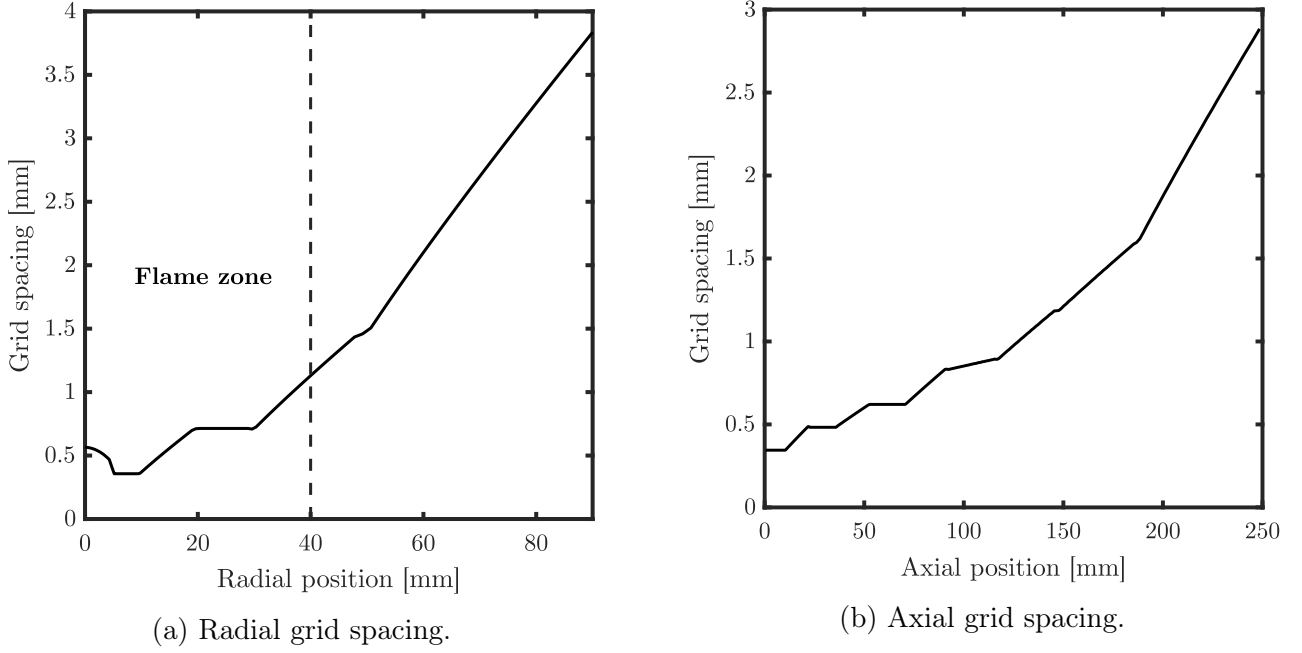


Figure 4.3: Radial and axial grid spacing at $x = 0$ and $r = 0$ respectively for the grid used in the SWB5 and SWB6 simulations.

of SWB6 were performed, which were identical beyond the method used to compute reaction source terms. One SWB6 simulation was performed where computation of the reaction source terms was performed solely with ANNs as described in the following section, and one where the VODE solver was used to evaluate to integrate the GRI 1.2 mechanism directly. The simulation of the SWB5 flame was undertaken using ANNs only.

4.3.2 Cambridge/Sandia swirl burner simulation results

Figures 4.4 and 4.5 show radial mean and RMS profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm from the ANN and DI simulations, as well as experimental values, for temperature, equivalence ratio and CH_4 , O_2 , H_2O , CO_2 , CO and H_2 mass fractions. In Figures 4.4 and 4.5 it can be seen that there is very good agreement between the mean and RMS radial profiles from the simulations using the ANNs and DI, showing that the random data generation method in conjunction with the MMLP method is able to accurately simulate turbulent premixed flames. Agreement is also generally good between the simulations and the experimental values, although as in Brauner et al. [84] the flame spreading rate is over predicted leading to differences in the major species at $x = 50$ mm. As reported by Nambully et al. [111], Proch and Kempf. [112] and Turkeri et al. in their studies of SWB5, under predictions of the CO mass fraction are seen near the centre line at all axial locations, with the H_2 mass fraction also underpredicted at all axial locations.

Figures 4.6, 4.7 and 4.8 show mean and RMS radial profiles for minor species for which experimental data is not available. These species are more difficult to predict than the major species, but it can be seen that generally agreement between the ANN and DI simulations is very good. Although some differences can still be seen between profiles for species such as H_2O_2 at $x = 50$ mm, the use of the MMLP method has increased the accuracy of predictions significantly compared to the ANNs used in the simulation of Sydney flame L.

As the random data set the MMLPs were trained on is quite general and the MMLPs have been shown to perform well in the SWB6 simulation, the same MMLPs can be applied to the SWB5 simulation without the need to check their performance by also doing a DI simulation. Figures 4.9 and 4.10 show radial mean and RMS profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm from the simulation using ANNs as well as experimental values, for temperature, equivalence ratio and CH_4 , O_2 , H_2O , CO_2 , CO and H_2 mass fractions. Generally, as in the SWB6 simulation, the agreement is very good between the simulation and experiment. As in the ANN and DI SWB6 simulations, and also in the simulations of Brauner et al. [84], the flame spreading rate is over predicted at the highest axial locations. This is possibly a consequence of the grid resolution and does not indicate a specific issue with the ANNs. Further, the temperature is over predicted near the centre line, which was also reported by Nambully et al. [111], Proch and Kempf. [112], Turkeri et al. [113] and Brauner et al. [84], so is likely to also be seen in a simulation using DI. The equivalence ratio is also under predicted near the centre line particularly at $x = 10$ mm which, as shown by Nambully et al. [111], is due to the lack of inclusion of differential diffusion in this simulation. Finally, the CO and H_2 mass fractions are generally under predicted at all axial locations, which again has been reported by Nambully et al. [111], Proch and Kempf [112] and Turkeri et al. [113] in their studies of SWB5 and also observed in the DI simulation of SWB6, and is more likely to be caused by the mechanism used than be a consequence of using the ANNs. These minor disagreements notwithstanding, the general agreement is very good between the ANN simulation and the experiment, showing that both the generality of the training data and the accuracy of the MMLP method such that the resulting ANNs can be successfully applied to families of similar flames.

Table 4.6 shows the speed-up obtained using the MMLPs in the SWB6 simulation. The inference time using the MMLPs is larger than using a single MLP, so the speed-up is 14x rather than 18x as found in the Sydney L flame simulations. However, a 14x speed-up is more than enough to remove the computational bottleneck present in the DI simulations. Table 4.7 shows the CPU hours required for the different parts of the SWB6 simulations. It can be seen from Table 4.7 that the training data generation and MMLP training time is insignificant compared to the

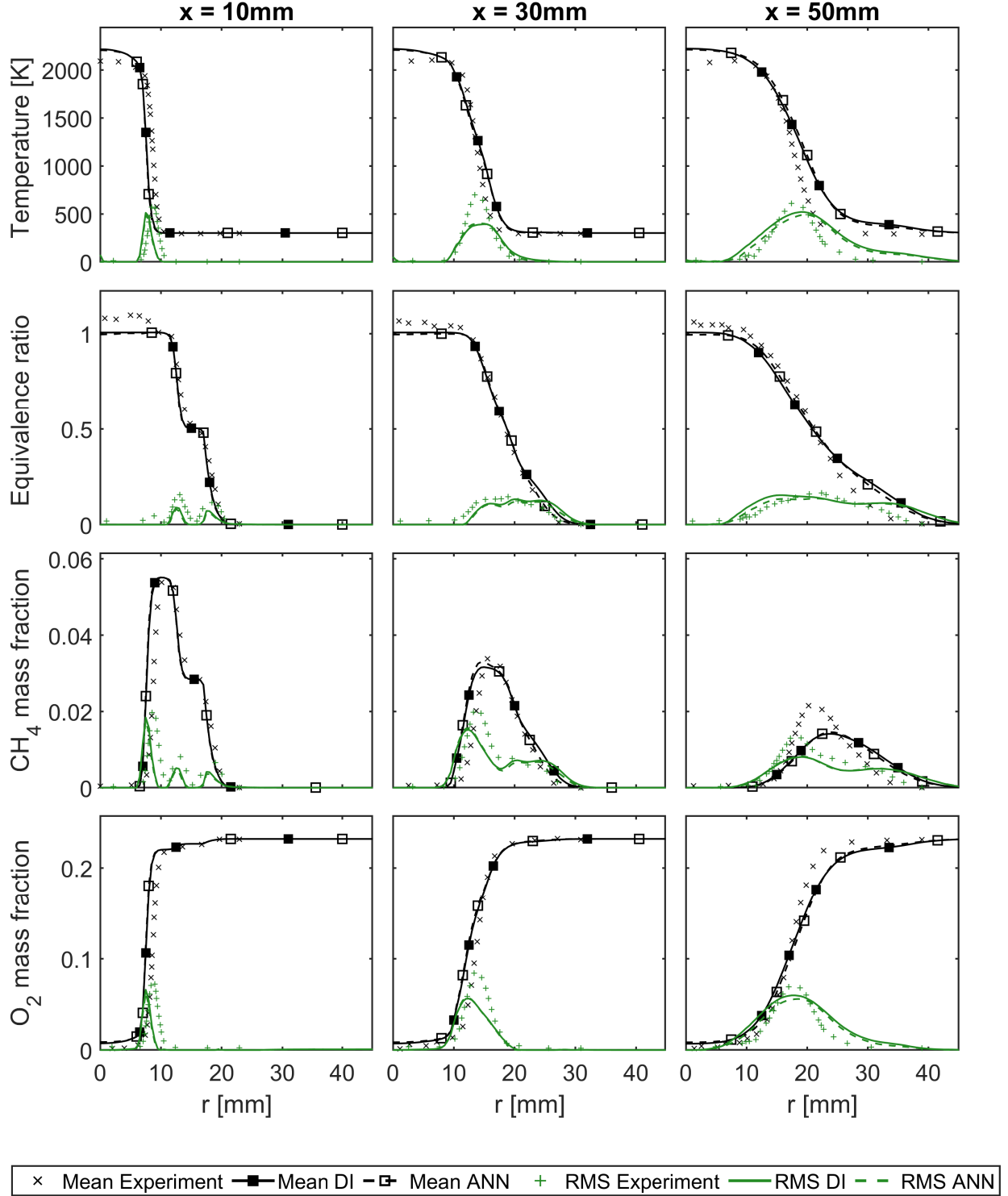


Figure 4.4: Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for temperature, equivalence ratio, CH₄ and O₂ mass fraction.

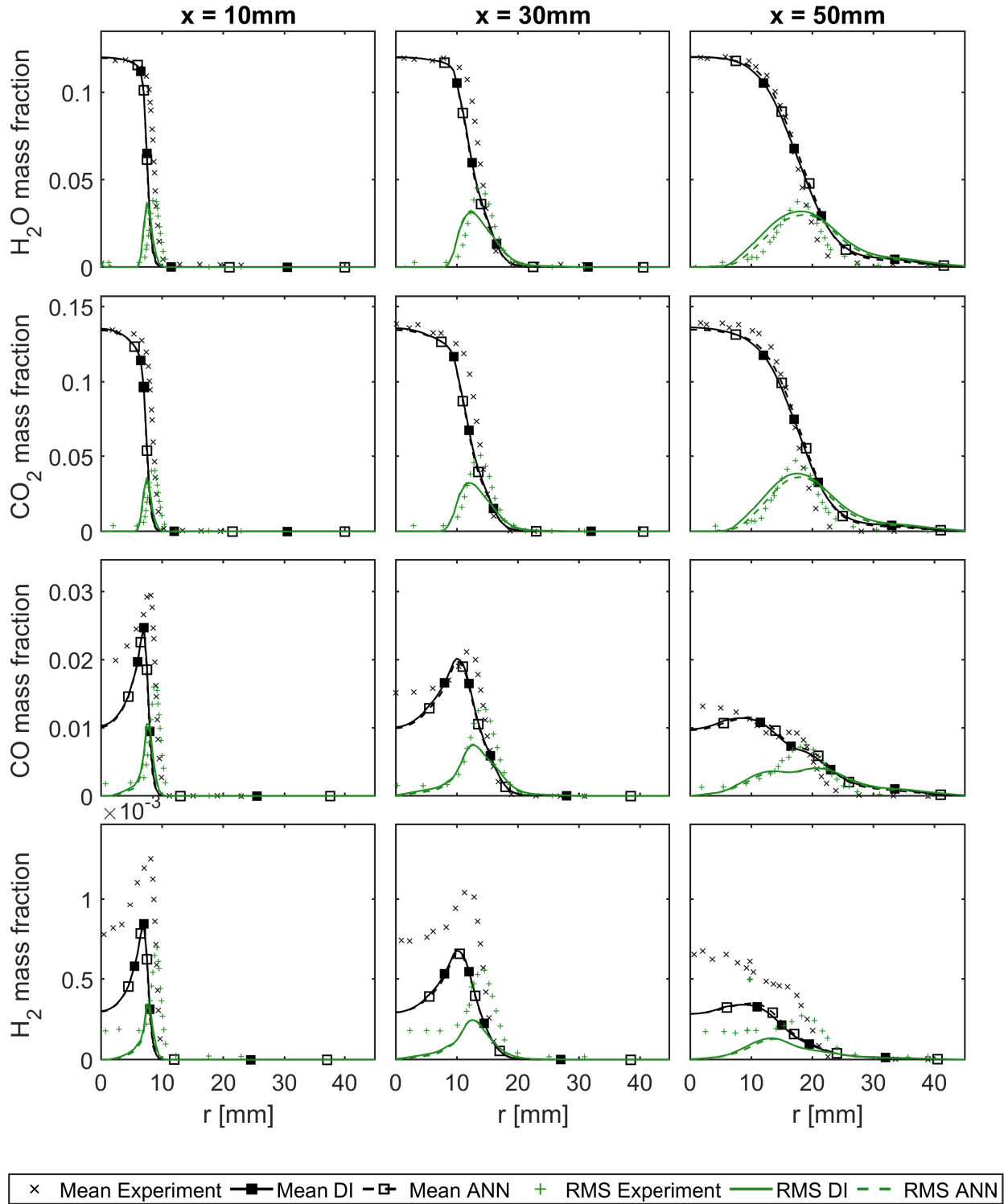


Figure 4.5: Radial mean and rms profiles at axial locations $x = 10\text{ mm}$, $x = 30\text{ mm}$ and $x = 50\text{ mm}$ for SWB6 simulations using ANNs and DI, as well as experimental values, for H_2O , CO_2 , CO and H_2 mass fraction.

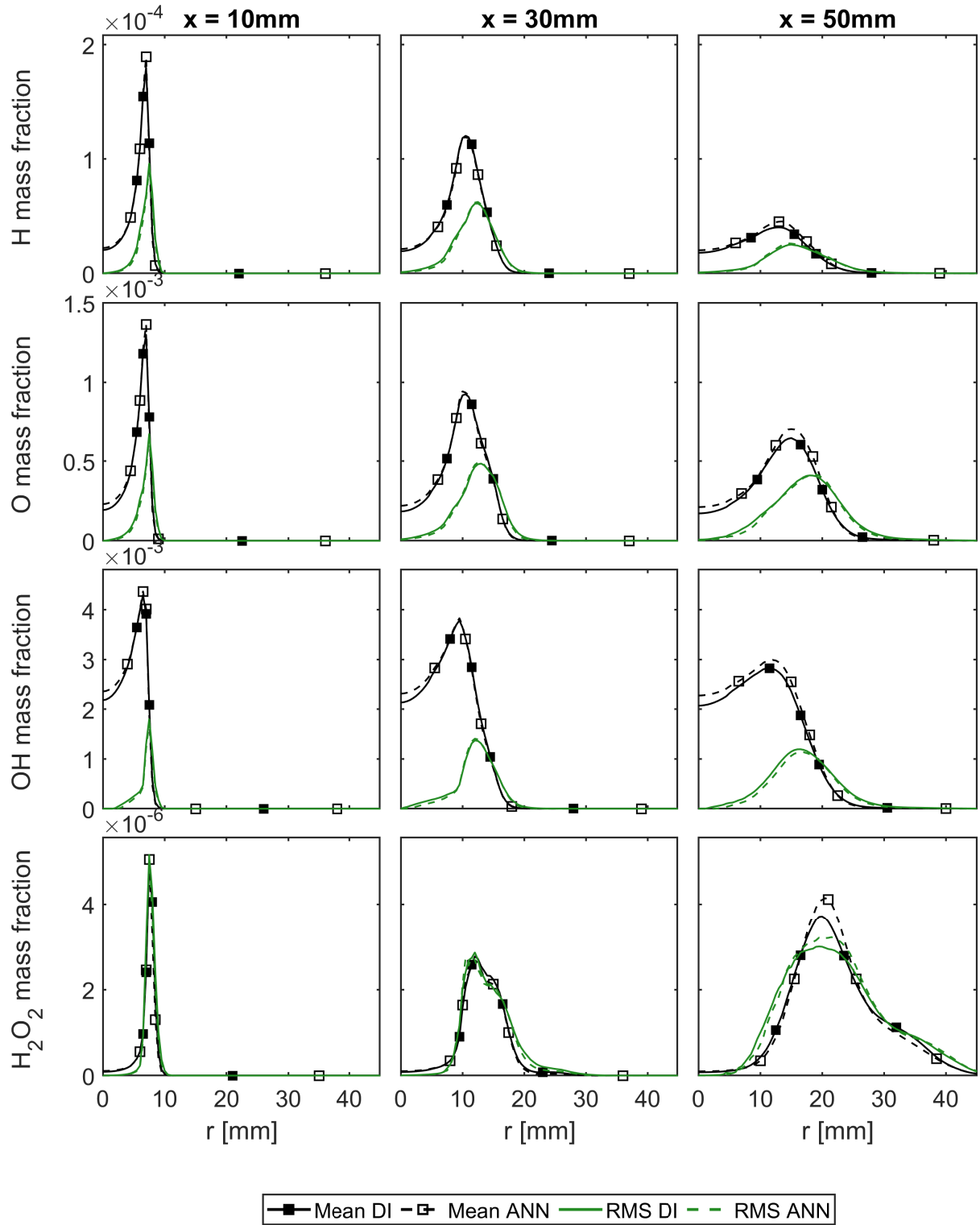


Figure 4.6: Radial mean and rms profiles at axial locations $x = 10\text{ mm}$, $x = 30\text{ mm}$ and $x = 50\text{ mm}$ for SWB6 simulations using ANNs and DI, as well as experimental values, for H, O, OH and H₂O₂ mass fraction.

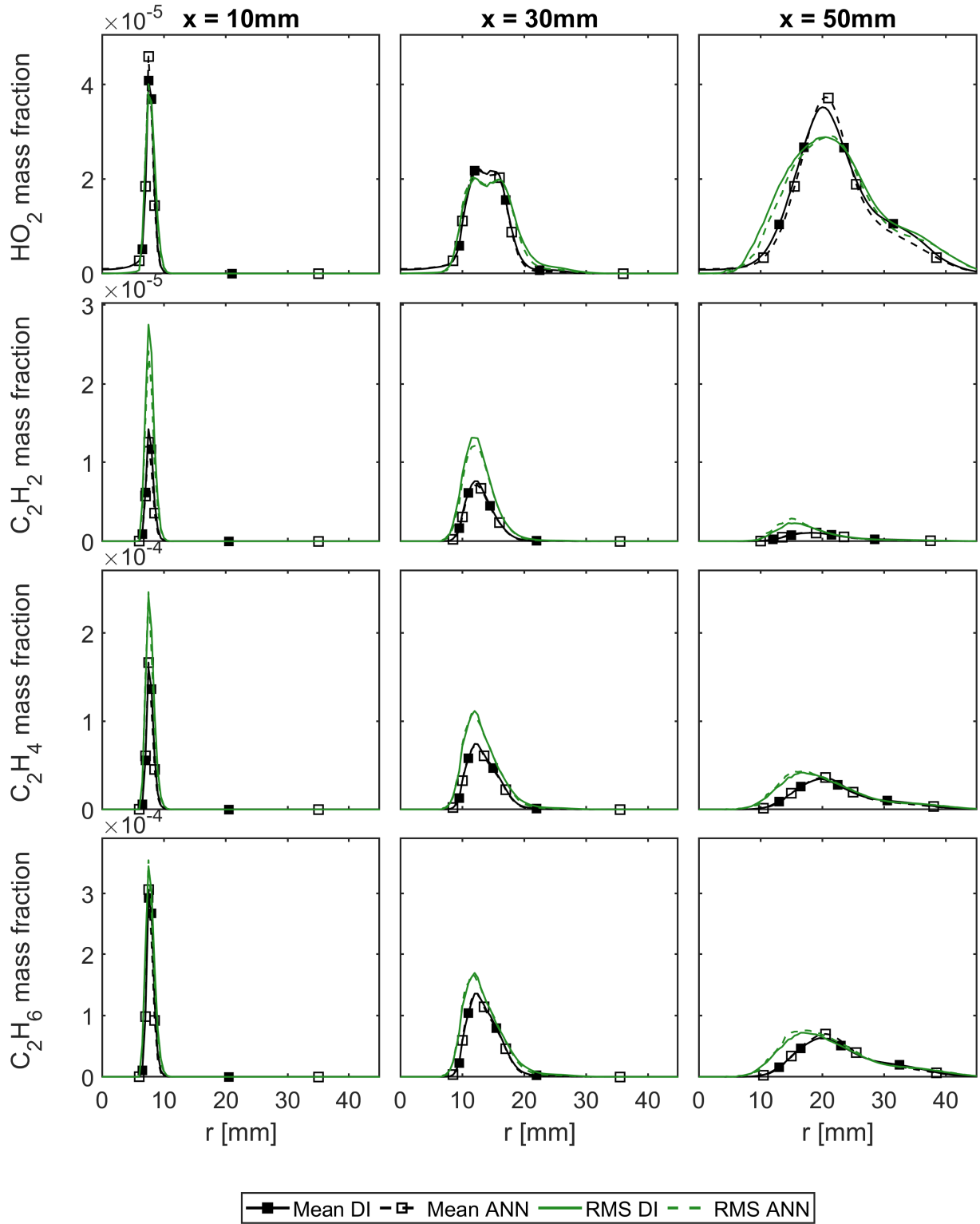


Figure 4.7: Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for HO₂, C₂H₂, C₂H₄ and C₂H₆ mass fraction.

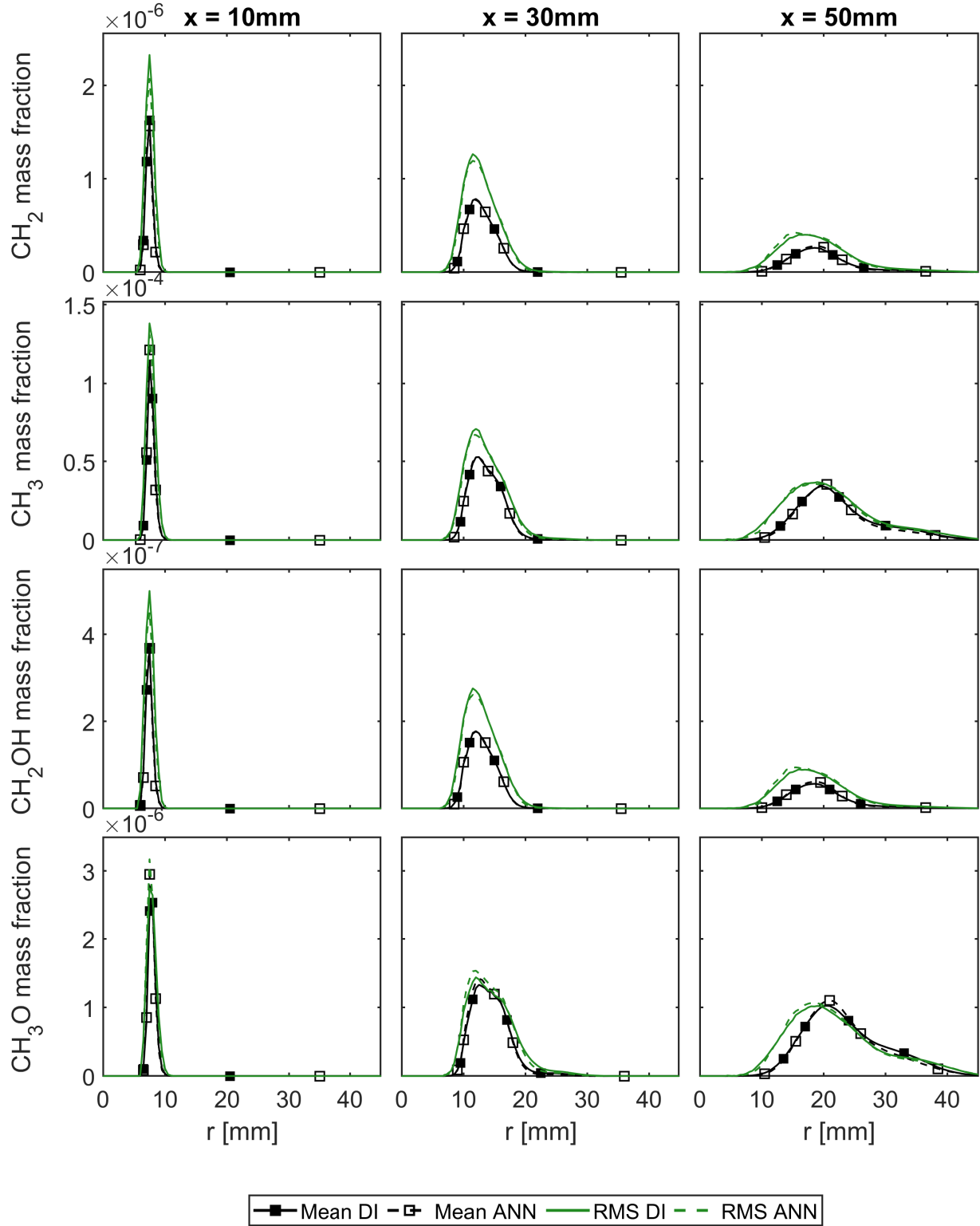


Figure 4.8: Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB6 simulations using ANNs and DI, as well as experimental values, for CH_2 , CH_3 , CH_2OH and CH_3O mass fraction.

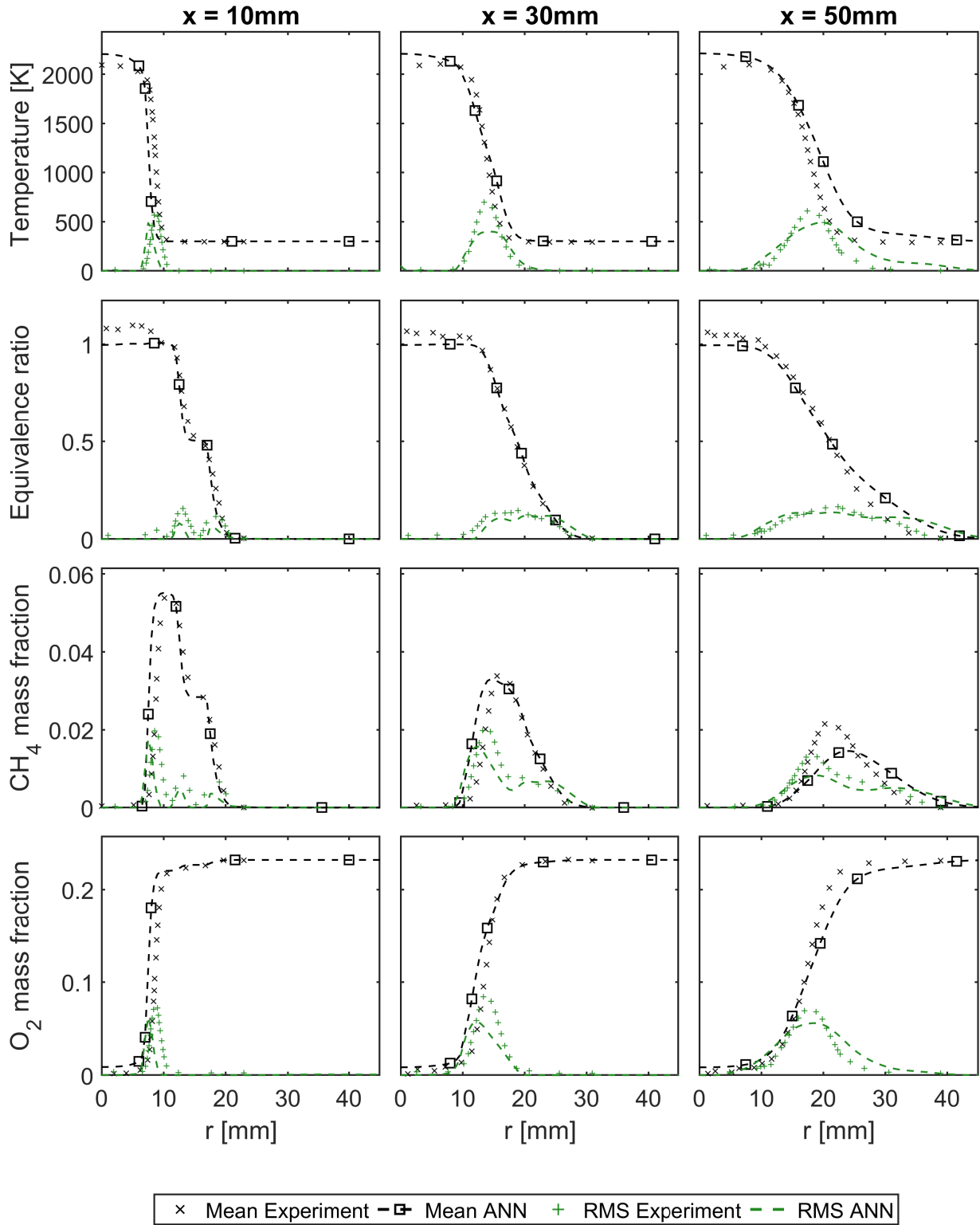


Figure 4.9: Radial mean and rms profiles at axial locations $x = 10\text{ mm}$, $x = 30\text{ mm}$ and $x = 50\text{ mm}$ for SWB5 simulations using ANNs and DI, as well as experimental values, for temperature, equivalence ratio, CH_4 and O_2 mass fraction.

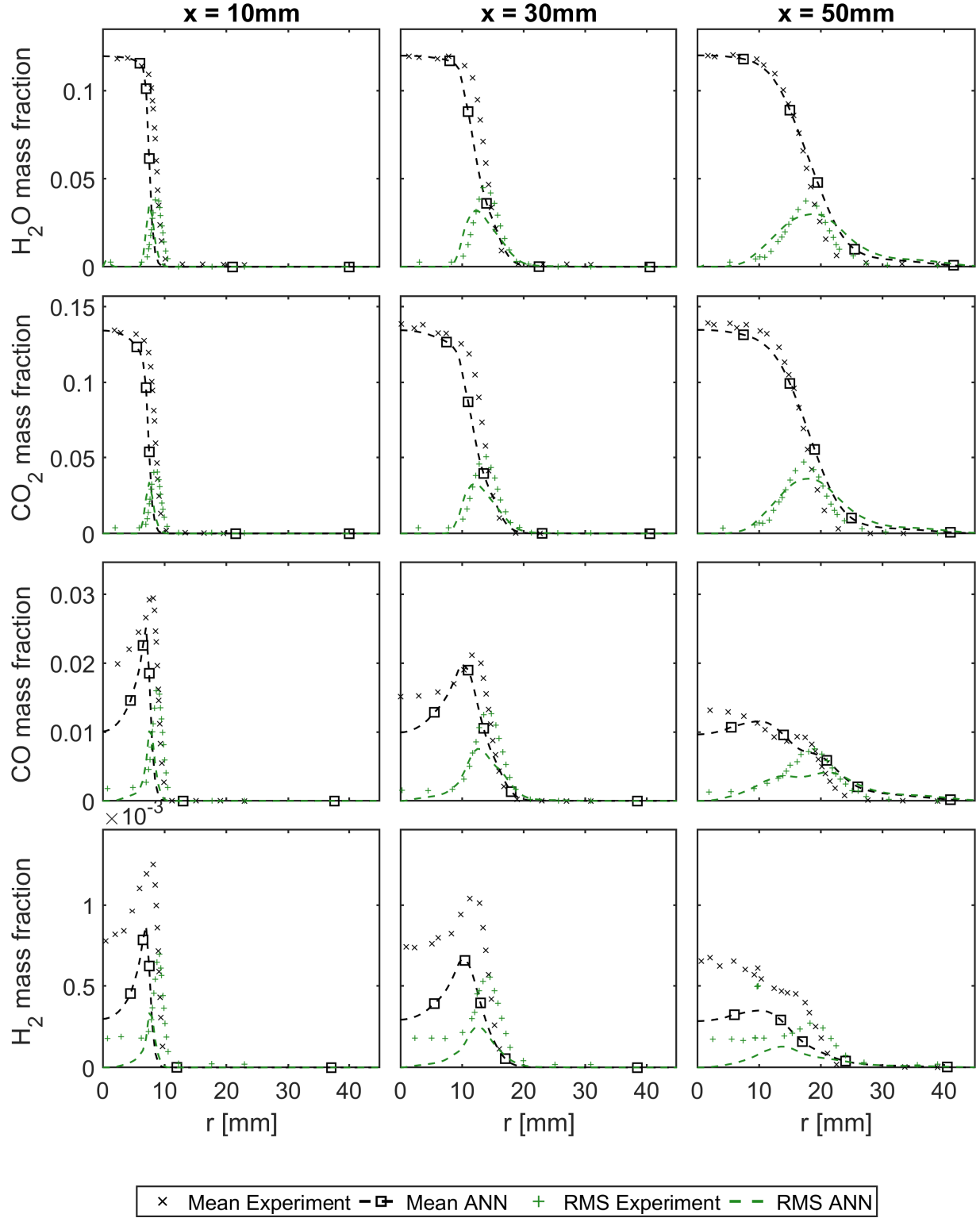


Figure 4.10: Radial mean and rms profiles at axial locations $x = 10$ mm, $x = 30$ mm and $x = 50$ mm for SWB5 simulations using ANNs and DI, as well as experimental values, for H₂O, CO₂, CO and H₂ mass fraction.

Table 4.6: Average time taken for reaction source term computations in SWB6 simulations using ANNs and DI, normalised to the time taken for reaction source term computations using DI.

Method	Reaction time	Speed up
DI	1.000	-
ANN	0.071	14x

Table 4.7: Time taken for different parts of the simulations of SWB6 using VODE and the MMLPs.

Stage	DI CPU time [hours]	ANN CPU time [hours]	Speed up
1D premixed flame simulations	-	490	-
ANN training	-	540	-
LES-PDF simulation	166000	44400	3.74x
Total	166000	45430	3.65x

CPU hours required for the SWB6 LES-PDF simulations. This means that even including the ANN tabulation time, it is still over 3x faster to perform simulations using MMLPs tabulated thermochemistry than it is to perform DI simulations.

Chapter 5

A method for incorporating conservation laws in combustion thermochemistry

Chapters 3 and 4 showed that the method of random data generation in conjunction with at least one MLP for each species is capable of simulating premixed and non-premixed flames with a high degree of accuracy. However, one of the main motivations for introducing the random data generation technique was that the ANN predictions do not obey mass and element conservation, or the mathematical bounds arising from the definition of mass fractions. The random data generation technique used to account for these element conservation errors requires the selection of allowable windows for the element ratios in the randomisation process. In Chapters 3 and 4, these windows were estimated based on prior knowledge. This is not ideal as the size of the windows is dependent on the ANN error in the particular application and so although it can be estimated based on experience it is unknown ahead of time. It is therefore desirable to formulate a method for rectifying element conservation errors in the ANN predictions. In the rest of this chapter such a method is proposed, which performs automatic selection and recalculation of certain species mass fractions for an arbitrary mechanism in order to satisfy the required bounds and conservation laws to machine precision.

The work in this chapter and in Appendix B has been submitted for publication in Readshaw et al. [114].

5.1 General principles of conservation method

In order to obey the conservation laws, a new composition, having been detransformed if necessary to yield Y_i^* , must satisfy certain constraints that consist of linear combinations of the species mass fractions. These constraints were given in Section 2.1.1 and are repeated here. The first constraint is that the species mass fractions must be bounded:

$$0 \leq Y_i^*(t + \Delta t) \leq 1. \quad (5.1)$$

The mass conservation constraint is given by:

$$\sum_{i=1}^{N_{sp}} Y_i^*(t + \Delta t) = 1, \quad (5.2)$$

while the element conservation constraints are given by:

$$\sum_{i=1}^{N_{sp}} A_{ji} Y_i^*(t + \Delta t) = Z_j(t), \quad (5.3)$$

where $j = 1, 2, \dots, N_e$ and N_e is the number of elements for which conservation is required in the reaction mechanism. Elements that do not participate in reactions (such as nitrogen if present only as N_2 in a mechanism that does not include nitrogen chemistry) can be ignored. A composition that satisfies Equation 5.3 will automatically satisfy Equation 5.2, so to obey both mass and element conservation it is sufficient to satisfy Equation 5.3.

The question then arises of how to enforce the conservation laws. If the predictions of the target outputs were perfect, then the conservation laws would be automatically satisfied. As such, techniques which reduce the loss by increasing the MLP accuracy, such as the use of a single MLP for each species [69, 57] or the multiple MLP method [58], naturally reduce the error in the conservation laws. In practice, however, the amount of constraint violation will never be zero and some points may even severely violate the constraints in search of reduction in the loss function. To mitigate this, one possibility is to introduce a term to the loss function that penalises constraint violation, as done, for example, in Bode et al. [115], where a penalty term for mass conservation was included. However, this does not guarantee that the constraints are not violated, and furthermore raises the issue of how to weight each term in the loss function in order to effectively impose the penalty. Modifying the loss function is also problematic when using separate MLPs for each species, as the constraints contain sums over species which the separate MLPs have

no knowledge of.

Instead, the composition can be manipulated in order to generate another new composition which automatically satisfies the constraints to machine precision. Using new values of all of the species in addition to the knowledge of the constraints over-determines the system. Therefore, rather than accepting the new composition and ignoring the constraints, mass fractions of certain species can be ignored and the constraints used instead to calculate these mass fractions.

Any process can be used to generate the new composition Y_i^* from a composition Y_i , before the conservation is enforced. The method for applying the conservation laws will be outlined in the context of using ANNs to make reaction step predictions in fractional step simulations, in which case the process can be expressed as:

$$Y_i^*(t + \Delta t) = \text{ANNs}[h(t), Y_i(t)]. \quad (5.4)$$

The basic procedure for correcting the ANN predictions in order to satisfy the conservation laws to machine precision will now be illustrated using a mixture consisting of only two species, with their mass fractions given by Y_1 and Y_2 . After the ANN predictions of both species are made any negative predictions are set to zero and any mass fractions greater than one are clipped to satisfy Equation 5.1. The mass fractions satisfying the conservation laws, Y_i^{con} , are initialised to these bounded predictions. Then, in order to satisfy an element conservation constraint, a correction is applied to one of the predicted species mass fractions. For example, if the prediction for Y_1 is ignored, its new value can be calculated using:

$$Y_1^{con} = \frac{Z_j - A_{2j}Y_2^{con}}{A_{1j}}. \quad (5.5)$$

The composition will now satisfy the conservation law for the element being considered. Alternatively, if the prediction for Y_2 is ignored, its new value can be calculated in a similar way using:

$$Y_2^{con} = \frac{Z_j - A_{1j}Y_1^{con}}{A_{2j}}. \quad (5.6)$$

It should be noted that only one of Equation 5.5 or Equation 5.6 is needed in order to calculate one species mass fraction to conserve the element under consideration. The species that is calculated thus will be referred to as the residual species. It is only the mass fractions of the residual species that will be altered to satisfy the

mass and element conservation laws, with the mass fractions of the other species left unchanged.

To ensure that Y_i^{con} remains bounded, the newly calculated residual species mass fractions must be also be bounded. In practice, the most common violation of the species mass fraction bounds occurs through negative mass fraction predictions. It is therefore worth examining how negative residual species mass fractions may arise. Looking at Equation 5.5, the residual species mass fraction would be negative if the total amount of element contained in Y_2^{con} was predicted to be greater than the available amount of that element at time t . Figure 5.1 illustrates how a negative residual prediction could arise by arbitrarily considering Y_1 as the residual species. From the top of Figure 5.1, the first two horizontal bars show the contributions of each species to the element under consideration in the initial composition and the target composition respectively. The third bar shows the contribution of each species to the element under consideration for fictitious ANN predictions. It can be seen in the third bar that there is a large overprediction in Y_2 compared to the target value, and therefore a large overprediction of the amount of the element under consideration contained in Y_2 . In fact, the predicted amount of the element contained in Y_2 is larger than its total amount available at time t . In order to conserve this element, the value of Y_1 is therefore calculated via Equation 5.5 to be negative. This is illustrated by the fourth bar in Figure 5.1, where the total amount of the element is now conserved, but the contribution of Y_1 to the element is negative.

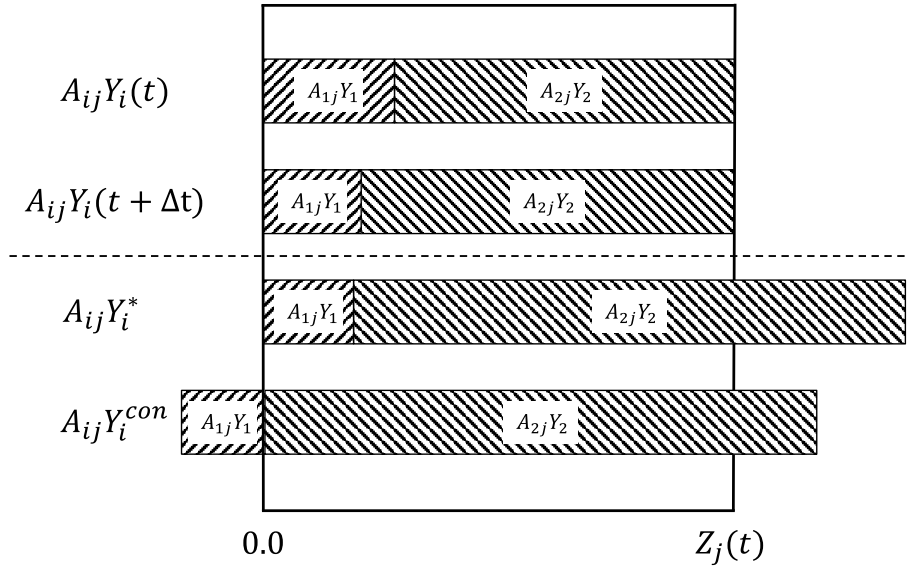


Figure 5.1: Illustration of a large overprediction of Y_2^* resulting in a mass fraction of Y_1^{con} which is calculated to be negative in order to satisfy the conservation law for the element under consideration.

Figure 5.1 raises two important points for the foundations of the proposed

method. The first is that the residual species should be chosen after the ANN predictions are made, and it should be the species that contains the highest possible amount of the element to be conserved in the predictions. This reduces the chance of the non-residual species containing more of the element than was available at time t , and so reduces the chance of negative values being calculated for residual species. This has the important implication that the choice of residual species should be different for each composition. Figure 5.2 shows the same initial and target composition as Figure 5.1, but now considers Y_2 as the residual species. Figure 5.2 shows that after using Equation 5.6 the composition Y_i^{con} remains bounded. Because the contribution of Y_2^* to the element under consideration is so much larger than that of Y_1^* , the error in Y_1 would have to be extremely large to calculate Y_2^{con} as negative.

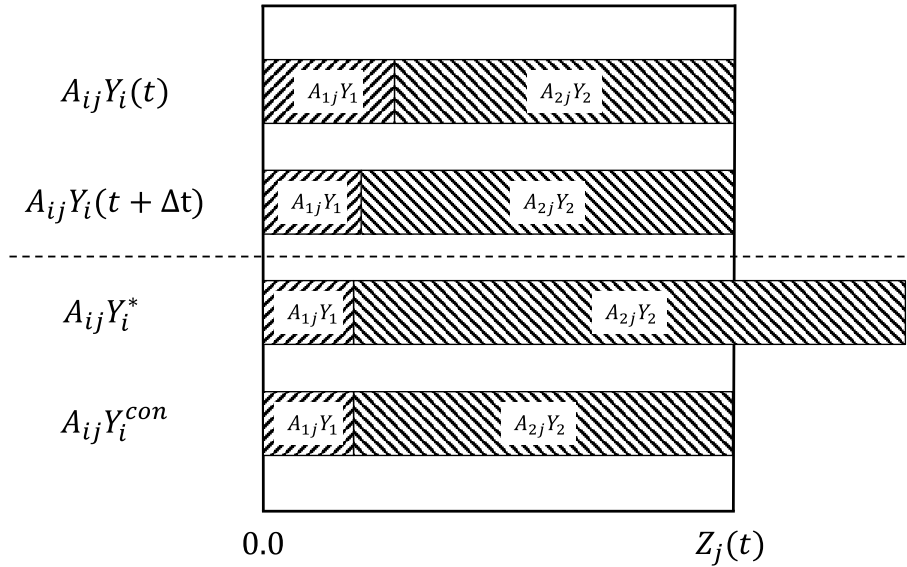


Figure 5.2: Illustration of a large overprediction of Y_2^* which can be corrected in order to satisfy the conservation law for the element under consideration by calculating Y_2^{con} .

The second important point raised by Figures 5.1 and 5.2 is that the chance of negative residual species mass fractions occurring will naturally decrease with increasing prediction accuracy. Wildly inaccurate predictions from a correctly implemented ANN method, such as those which are significantly out-of-bounds, are extremely uncommon and should be screened out. Moderate prediction errors are easily absorbed by the proposed method.

The expansion of the method illustrated so far to the enforcement of Equation 5.3 for a mixture containing an arbitrary number of species is straightforward. ANN predictions of all the species are made. A separate residual species is chosen to enforce each of the N_e element constraints. To proceed, it is simplest to eliminate the elements one at a time. For example, if the elements are C, H and O, a possible choice of residual species is CH_4 (for C), H_2O (for H without changing C) and O_2

(for O without changing C or H). Another possibility is CH_4 (for C), H_2O (for H) and both of CO_2 and CO (for O without altering C). For simplicity, we restrict ourselves to using a single residual species for each element, hence there are a total of N_e residual species. This has the consequence that the final residual species must contain a single element only to avoid disrupting the already conserved elements. This means the number of species which can be considered as the final residual species is limited, as is the number of possible orders in which to eliminate the elements.

Once the element solution order is determined, the first residual species is chosen as the species predicted to contribute the most to the mass fraction of the first element to be eliminated. The second residual species is the species which does not contain the first element and is predicted to contribute the most to the mass fraction of the second element. This process continues until all of the residual species have been determined. At that point, the new residual species mass fractions can be calculated to satisfy the mass and element conservation laws.

Depending on which species can be considered as the final residual species and the ANN accuracy, it may not be possible to avoid a negative final residual species. As an example, the GRI 1.2 mechanism has 31 species, but there are only 6 which contain a single element: H_2 , H, O, O_2 , C and N_2 . The conservation of N is unnecessary in GRI 1.2, so N_2 is ignored. Of the remaining species, it is possible that none will occur in large enough amounts to prevent the error in the non-residual species from overwhelming the amount of the element under consideration at time t . Therefore, an algorithm is proposed to rectify a negative value in the final residual species, in the instances where it arises, without altering any of the other elements.

The rest of this section details the computational implementation of the procedure for enforcing the conservation laws in ANNs, starting from the input composition, $[h(t), Y_i(t)]$, and finishing with a predicted composition which is bounded and obeys the conservation laws, $Y_i^{\text{con}}(t + \Delta t)$. The procedure can be broken down into the following four steps:

1. Generate initial predictions (Algorithm 1)
2. Select residual species (Algorithm 2)
3. Calculate residual species mass fractions (Algorithm 3)
4. Rectify negative final residual species if necessary (Algorithm 4)

Each of these steps is described in the following sections. A worked example of the solution procedure using Algorithms 1 - 4 is also provided in Appendix B.

5.1.1 Step 1: Generate initial predictions

The first step in the process is to generate the initial ANN predictions. If the ANNs predict the species, $Y_i^*(t + \Delta t)$, then these can be used directly. If the ANNs predict the changes in the species over a given time step then predictions of the change, ΔY_i^* , can be made and added to the input composition to retrieve $Y_i^*(t + \Delta t)$. Any compositions with extremely unphysical predictions should be screened out. Any $Y_i^*(t + \Delta t)$ that lie moderately outside of the mass fraction bounds are set to lie within them, to give the bounded predictions $Y_i^b(t + \Delta t)$. If the ANNs predict $Y_i^*(t + \Delta t)$ then predictions can be guaranteed to be in bounds by using modified rectified linear unit activation functions on the output layer.

Algorithm 1 Generate initial ANN prediction

Require: $h(t), Y_i(t)$ ▷ Input composition

if ANNs predict outputs **then**

$Y_i^*(t + \Delta t) = \text{ANNs}[h(t), Y_i(t)]$ ▷ Predicted outputs

else if ANNs predict changes **then**

$\Delta Y_i^* = \text{ANNs}[h(t), Y_i(t)]$ ▷ Predicted changes

$Y_i^*(t + \Delta t) = Y_i(t) + \Delta Y_i^*$ ▷ Predicted outputs

end if

for $i = 1, N_{sp}$ **do** ▷ Remove out of bounds predictions

if $Y_i^*(t + \Delta t) < 0$ **then**

$Y_i^*(t + \Delta t) = 0$

end if

if $Y_i^*(t + \Delta t) > 1$ **then**

$Y_i^*(t + \Delta t) = 1$

end if

end for

$Y_i^b(t + \Delta t) = Y_i^*(t + \Delta t)$

return $Y_i^b(t + \Delta t)$ ▷ Bounded predictions

5.1.2 Step 2: Select residual species

The second step is to choose which of the predicted species to consider as residual species for the given composition. As explained, the final residual species is restricted to species which are made up of a single element only. The number of candidate final residuals is denoted by N_C , while the index of each candidate final residual in the mechanism is stored in $\mathcal{C}_c$. The index of the element contained in each of the candidate final residuals are denoted by $\mathcal{E}_c^{\mathcal{C}}$. Because the number of element solution orders is limited by the candidate final residuals, it is preferable to select the final

residual species first in order to ensure that a valid order is chosen. To do this, the predicted element mass fractions are calculated using the bounded predictions. As explained previously, to try and avoid negative residual predictions, the final residual species, $\mathcal{S}_{N_e}$, is chosen to be the candidate final residual species which contributes the highest proportion of the mass of its respective element. The element solution order, $\mathcal{E}_j$, for the N_e elements is then determined. The final element in the solution order, $\mathcal{E}_{N_e}$, will be the one contained by the final residual species $\mathcal{S}_{N_e}$. The order for the remaining elements has been found to be unimportant and can be determined as desired. Once the element solution order has been fixed, the remaining residual species can be determined.

Algorithm 2 Select residual species

Require: $Y_i^b(t + \Delta t)$ $\triangleright$ Non-negative ANN predictions**Require:** $\mathcal{C}_c$ $\triangleright$ Indices of N_c candidate final residuals**Require:** $\mathcal{E}_c^c$ $\triangleright$ Indices of the elements contained in the candidates

Calculate predicted element mass fractions:

for $j = 1, N_e$ **do**

$$Z_j^b(t + \Delta t) = \sum_{i=1}^{N_{sp}} A_{ji} Y_i^b$$

end for

Find the index of the final residual species in the candidates by calculating how much each candidate contributes to its element:

$$\mathcal{T} = \text{maxloc} \left(\frac{1 \times Y_{\mathcal{C}_c}^b}{Z_{\mathcal{E}_c^c}^b} \right)$$

 $\mathcal{S}_{N_e} = \mathcal{C}_{\mathcal{T}}$ $\triangleright$ Index of final residual species in mechanism $\mathcal{E}_{N_e} = \mathcal{E}_{\mathcal{T}}^c$ $\triangleright$ Index of element contained in final residual $\mathcal{E}_j = (\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{N_e})$ $\triangleright$ Element solution order

Determine the other residual species:

for $k = 1, N_e - 1$ **do**

Find indices of possible residual species for the current element, depending on the elements which have already been eliminated:

 $\mathcal{Q}_q = \mathcal{S}_i^{\text{tot}}$ which contain $\mathcal{E}_k$ and may contain any of $\mathcal{E}_l$, with $l = k + 1, N_e$ Index of residual species for the k th element in the possible species:

$$\mathcal{T} = \text{maxloc} \left(\frac{A_{\mathcal{E}_k} \mathcal{Q}_q Y_{\mathcal{Q}_q}^b}{Z_{\mathcal{E}_k}^b} \right)$$

Index of residual species for the k th element in the mechanism: $\mathcal{S}_k = \mathcal{Q}_{\mathcal{T}}$ **end for****return** $\mathcal{S}_j, \mathcal{E}_j$ $\triangleright$ Indices of residual species and element solution order

5.1.3 Step 3: Calculate mass fractions of residual species

Once the residual species have been determined, the conservation of mass and elements can be enforced. The target element mass fractions, $Z_j(t)$, are calculated using the input composition. The conserved species mass fractions, $Y_i^{\text{con}}(t + \Delta t)$, are initialised to the bounded ANN predictions. A switch for the species, $\mathcal{B}_i$, is introduced and set to one for all species. The switch for the first residual species, $\mathcal{B}_{\mathcal{S}_1}$, is set to zero. The predicted mass fraction of the first element to be conserved

is calculated without the mass fraction of the first residual species, as follows:

$$Z_{\mathcal{E}_1}^{con} = \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_1 i} Y_i^{con}. \quad (5.7)$$

The new mass fraction of the first residual species can then be calculated in order to conserve the first element:

$$Y_{\mathcal{S}_1}^{con} = \frac{Z_{\mathcal{E}_1} - Z_{\mathcal{E}_1}^{con}}{A_{\mathcal{E}_1 \mathcal{S}_1}}. \quad (5.8)$$

The switch for the first residual species, $\mathcal{B}_{\mathcal{S}_1}$, is then set to one again, and the process is repeated until all of the new residual species mass fractions are determined. The calculated species mass fractions now conserve all of the elements, as well as the total mass. As explained, it is possible that a negative value may be encountered in the residual species, although this is unlikely owing to the choice of the choice of residual species as the one contributing the most to its corresponding element. If the residual species with the negative value is not the final one, a new solution order can be tried, and if negative residuals are still encountered, the composition can be rejected while the new state can be calculated with direct integration. As long as such instances are rare, this will not impact on the time saving of the ANN tabulation. If a negative value in the final residual species is encountered, then it can be rectified using the algorithm to be described in Step 4.

Algorithm 3 Calculate residual species mass fractions

Require: $Y_i(t)$ ▷ Original species composition
Require: $Y_i^b(t + \Delta t)$ ▷ Bounded ANN predictions
Require: S_j ▷ Indices of residual species from Step 2
Require: $\mathcal{E}_j$ ▷ Element solution order from Step 2
 Calculate element mass fractions at time t :
for $j = 1, N_e$ **do**

$$Z_j(t) = \sum_{i=1}^{N_{sp}} A_{ji} Y_i(t)$$

end for
 $Y_i^{con}(t + \Delta t) = Y_i^b(t + \Delta t)$ ▷ Initialise conserved mass fractions
 $\mathcal{B}_i = 1$ ▷ Set residual species switch
for $j = 1, N_e$ **do**
 $\mathcal{B}_{S_j} = 0$ ▷ Set switch for current residual species
 Calculate element mass in all but the current residual species:

$$Z_{\mathcal{E}_j}^{con} = \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_j i} Y_i^{con}$$

 Calculate new residual species mass fraction:

$$Y_{S_j}^{con} = \frac{Z_{\mathcal{E}_j} - Z_{\mathcal{E}_j}^{con}}{A_{\mathcal{E}_j S_j}}$$

 $\mathcal{B}_{S_j} = 1$ ▷ Reset switch
end for
return $Y_i^{con}(t + \Delta t)$ ▷ Composition obeying conservation laws

5.1.4 Step 4: Rectify final residual species if needed

If a negative value is encountered in the final residual species, it can be removed by modifying a pair of rectifying species, $\mathcal{R}_1$ and $\mathcal{R}_2$, to make up for the required change in the final element without altering any of the other elements. For example, if O_2 is chosen to be the final residual species and found to be negative, its mass fraction can be set to zero. This introduces extra O element, which must be removed to satisfy element conservation. To do this, the mass fraction of CO_2 can be reduced and the mass fraction of CO increased to account for the O element without changing the C element. Algorithm 4 shows how a negative value in the mass fraction of the final residual species can be rectified.

Algorithm 4 Rectify final residual species if needed**Require:** $Y_i^{con}(t + \Delta t)$ $\triangleright$ Conserved ANN predictions with negative final residual**Require:** S_{N_e} $\triangleright$ Index of final residual species**Require:** $\mathcal{R}_1, \mathcal{R}_2$ $\triangleright$ Indices of pairs of rectifying species for the final residual**Require:** $\mathcal{E}^{\mathcal{R}}$ $\triangleright$ Other element pair of rectifying species contain

$$a = -A_{\mathcal{E}_{N_e} S_{N_e}} Y_{S_{N_e}}^{con}$$

$$b = A_{\mathcal{E}_{N_e} \mathcal{R}_1}$$

$$c = A_{\mathcal{E}_{N_e} \mathcal{R}_2}$$

$$d = \frac{A_{\mathcal{E}^{\mathcal{R}} \mathcal{R}_1}}{A_{\mathcal{E}^{\mathcal{R}} \mathcal{R}_2}}$$

$$\Delta Y_{\mathcal{R}_1}^{con} = \frac{-a}{b - cd} \quad \triangleright \text{Change in first rectifying species mass fraction}$$

$$\Delta Y_{\mathcal{R}_2}^{con} = -d \Delta Y_{\mathcal{R}_1}^{con} \quad \triangleright \text{Change in second rectifying species mass fraction}$$

$$Y_{\mathcal{R}_1}^{con} = Y_{\mathcal{R}_1}^{con} + \Delta Y_{\mathcal{R}_1}^{con} \quad \triangleright \text{New mass fraction of first rectifying species}$$

$$Y_{\mathcal{R}_2}^{con} = Y_{\mathcal{R}_2}^{con} + \Delta Y_{\mathcal{R}_2}^{con} \quad \triangleright \text{New mass fraction of second rectifying species}$$

$$Y_{S_{N_e}}^{con} = 0 \quad \triangleright \text{Negative final residual mass fraction zeroed}$$

return $Y_i^{con}(t + \Delta t)$ $\triangleright$ Bounded conserved ANN predictions

After using Algorithms 1 - 3, and Algorithm 4 if necessary, the predicted composition now satisfies the bounds and the conservation laws. A worked example of the method is given in Appendix B. In the following section the conservation method is applied to 1D premixed flame simulations to evaluate its effectiveness.

5.2 Application to reaction source term predictions

To test the performance of the conservation method, simulations of 1D premixed flames with non-equal diffusivities (NEDs), as described in Chapter 4, are used as a test case. Although the ultimate aim is to apply ANNs to turbulent combustion problems, here the 1D premixed flames are deliberately chosen as the test case. This is because the effect of enforcing the conservation will be more easily isolated in 1D premixed flames, where as in turbulent cases a number of factors influence the error and it would be more difficult to directly see the effect of enforcing the conservation laws.

Reaction source term computations in the simulations were performed with three approaches: direct integration with the VODE solver, ANNs without enforced conservation and ANNs with enforced conservation. Two sets of flames were computed:

CH₄/air flames, computed with the GRI 1.2 mechanism, and C₃H₈/air flames, computed with the USC Mech II mechanism [116]. Details of the simulations are given in Table 4.2. A number of simulations were performed with a range of equivalence ratios chosen within the flammable ranges of each fuel. To initialise the simulations, the domain was filled with unburnt fuel/air mixture at the chosen equivalence ratio. To ignite the flame, half of the domain was initialised to equilibrium conditions, while the other half was initialised as unburnt. A minimum temperature of 500K was set for a point in the domain to be considered reactive.

Table 5.1: Premixed flame simulation details.

Domain length	10mm
Unburnt temperature	300K
Time step	1×10^{-6} s
Simulation length	0.02s

The GRI 1.2 mechanism contains four elements (O, H, C and N) and 31 species, while USC Mech II contains five elements (O, H, C, N and Ar) and 111 species. Neither GRI 1.2 or USC Mech II includes nitrogen chemistry, with N₂ the only species in each mechanism containing the N element, so the conservation of N is unnecessary in both mechanisms. Additionally, for USC Mech II Ar is neglected, meaning that in both mechanisms there are three elements to be conserved: O, H and C. The constraint on mass conservation for both mechanisms is that the sum of the reacting species must equal $1 - Y_{N_2}$, which will be automatically satisfied if the other elements are conserved. Both mechanisms share the same five candidate final residual species: C, H₂, H, O and O₂. The rectifying species for H₂ and H were chosen as H₂O and OH, while the rectifying species for O and O₂ were chosen as CO₂ and CO. Although it is very unlikely that C will ever be used as the final residual species due to its general small contribution to the C element, its rectifying species are chosen as CO₂ and O₂. Finally, NEDs are used for both sets of flames.

Because the intent of this test is to assess the effect of enforcing the conservation laws on the general predictive performance of the ANNs, the training data is not collected from a distinct abstract problem, but is also collected from 1D premixed flame simulations. The premixed flames used for data collection shared the same parameters as Table 5.1, but used equal diffusivities (EDs). To generate points with variable element ratios (which are required for simulations using differential diffusion), the random data generation method explained in Chapter 3 was used, with the original 1D premixed flame data discarded. The randomly generated data was integrated using VODE over a time step of 1×10^{-6} s to form the ANN training data sets. Single output MLPs were used for each species, with different targets used for GRI 1.2 and USC Mech II to show that the conservation method works whether

predicting the change in the composition over the time step or the composition directly. Table 5.2 gives the training information for the GRI 1.2 mechanism, while Table 5.3 gives the training information for USC Mech II.

Table 5.2: GRI 1.2 training set information and ANN architecture.

Parameter	Value
ϕ range	0.45 - 1.70
Number of data	370,000
Targets	$Y_i(t + \Delta t)$
Number of layers	4
Neurons per layer	32 30 30 1
Scaling	Linear to [-1 1]
Activation functions	n/a - $\tanh(x)$ - $\tanh(x)$ - $\max(-1, x)$
Training method	Levenberg-Marquardt [98] with Bayesian regularisation [96]
Training iterations	1500

Table 5.3: USC Mech II training set information and ANN architecture.

Parameter	Value
ϕ range	0.50 - 2.50
Number of data	250,000
Targets	ΔY_i
Number of layers	4
Neurons per layer	112 20 20 1
Scaling	Linear to [-1 1]
Activation functions	n/a - $\tanh(x)$ - $\tanh(x)$ - x
Training method	Levenberg-Marquardt [98] with Bayesian regularisation [96]
Training iterations	1000

The effect of enforcing the conservation in the simulations using the ANNs is assessed in three ways for each reaction mechanism. The first is by recording the mass and element conservation error in the simulations using the ANNs with and without enforced conservation. The second way that the conservation method is assessed is by comparing the steady state 1D premixed flame profiles obtained using the ANNs with and without enforced conservation with the profiles obtained using DI. The final way in which the conservation method is assessed is by comparing the

time taken for ANN reaction source term computations with and without enforced conservation. As the overarching aim of using the ANNs is to alleviate the reaction source term computation bottleneck, the conservation method must be sufficiently lightweight that it does not negate the overall speed benefits of the ANNs.

5.2.1 Results for CH₄/air flames with GRI 1.2

Table 5.4 shows the root mean square error in each of the quantities that should be conserved over all of the reacting points throughout simulations with NEDs at $\phi = 1.0$ for GRI 1.2, when using the ANNs with and without conservation. It can be seen that, the ANN predictions are subject to an error in the conserved quantities. Although this error is small, it can accumulate as a simulation progresses. In contrast, constraining the ANN predictions using Algorithms 1-4 shows that all of the quantities are conserved to machine precision for every single reacting point. Results are only shown for $\phi = 1.0$, but the application of conservation has the same effect in the other cases as well.

Table 5.4: Root mean square error in conserved quantities across the reaction step over all reacting points throughout simulations with non-equal diffusivities at $\phi = 1.0$ and GRI 1.2.

Quantity	ANNs RMSE	ANNs with conservation RMSE
Y_O^e	1.97×10^{-7}	1.51×10^{-18}
Y_H^e	6.91×10^{-7}	1.75×10^{-18}
Y_C^e	1.39×10^{-7}	3.57×10^{-19}
$1 - Y_{N_2}$	2.38×10^{-9}	3.60×10^{-20}

Figures 5.3 and 5.4 show CO and H₂ mass fraction profiles at the end of 1D premixed flame simulations with NEDs at various equivalence ratios using GRI 1.2 with the three methods for reaction source term computations. It can be seen in Figures 5.3 and 5.4 that there is a significant increase in accuracy when using ANNs and enforcing the conservation laws compared to using the ANNs to predict all species directly. Because the conservation method is not influenced by the ANN architecture, it can be easily appended to architectures that improve ANN accuracy, such as the MMLP method.

Figures 5.5 and 5.6 show the molar H/C and O/N profiles at the end of the GRI 1.2 simulations with the different reaction source integration methods. The deviation of the composition space from the DI simulation is seen when using the ANNs without the conservation method, while the profiles with the ANNs with

5.2 Application to reaction source term predictions

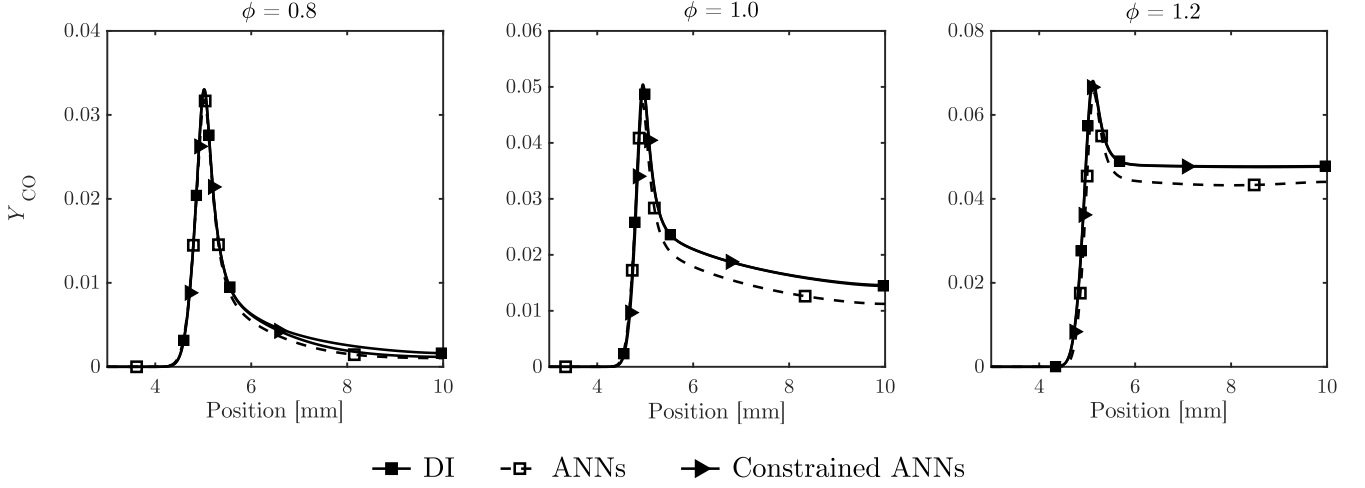


Figure 5.3: CO mass fraction profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

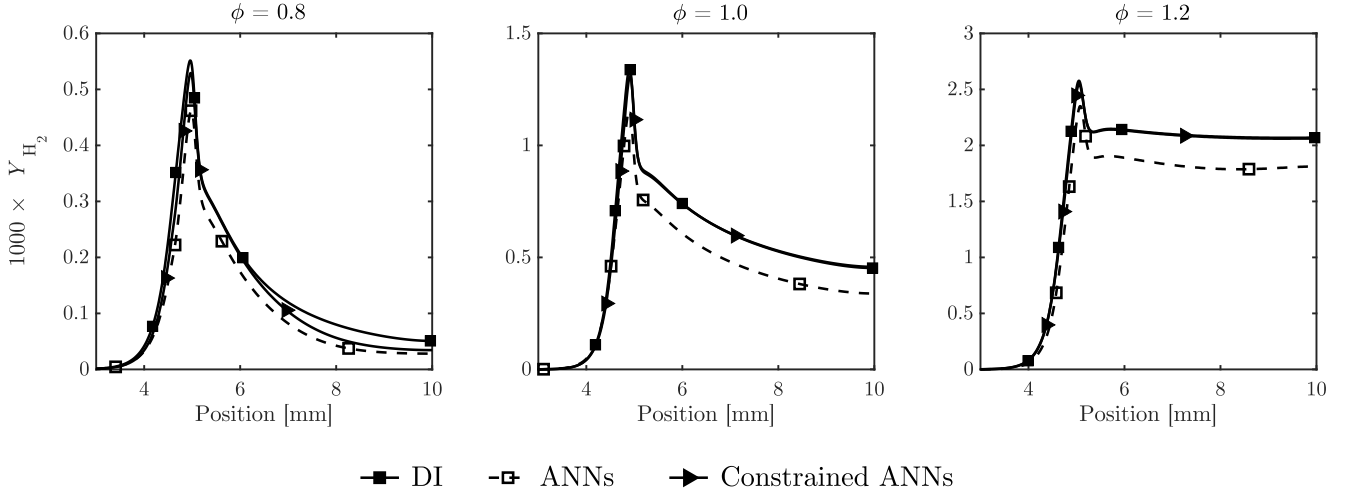


Figure 5.4: H₂ mass fraction profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

conservation are in much closer agreement to the DI profiles. Although the mass and elements are conserved to machine precision at every reacting point in the ANN simulation with the conservation method, the error in the ANN predictions of the species means there are some slight differences between the DI and conserved ANN species profiles. The action of differential diffusion then means that the element ratio profiles are not exactly reproduced by the ANNs with conservation, even though the mass and elements are conserved over the reaction step. Nevertheless, it is clear from Figures 5.5 and 5.6 that by enforcing the conservation laws the ANN prediction accuracy is greatly improved compared to predicting all of the species directly.

Table 5.5 shows the frequency of the residual species permutations recorded throughout the GRI 1.2 premixed flame simulations with NEDs at various equivalence ratios. At fuel lean and stoichiometric equivalence ratios the most frequent

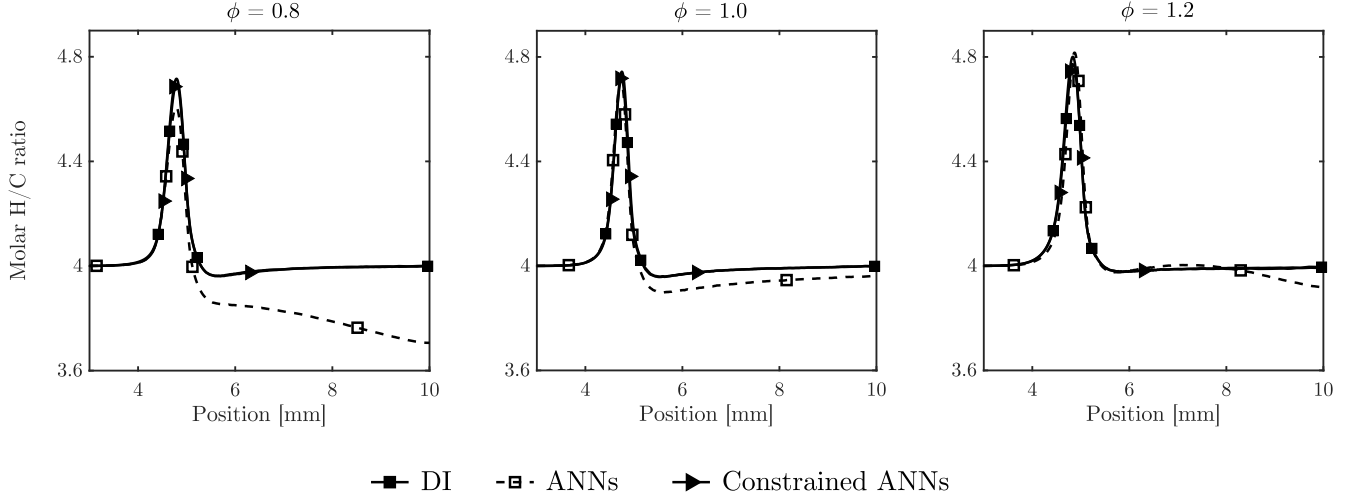


Figure 5.5: Molar H/C ratio profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

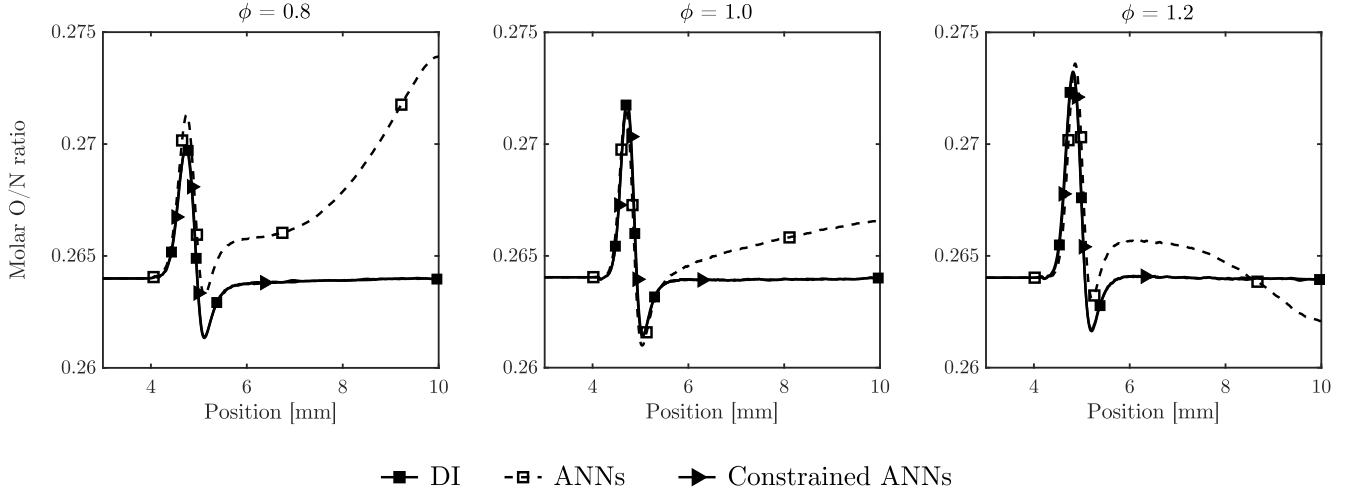


Figure 5.6: Molar O/N ratio profiles at the end of GRI 1.2 1D premixed flame simulations with NEDs when using different methods for reaction source term integration.

permutation is (CO_2 , H_2O , O_2), with no occurrences of H_2 being selected as the final residual species. As the equivalence ratio increases permutations with O_2 as the final residual become less common, with (CO_2 , H_2O , H_2) and then (CO , H_2O , H_2) becoming the dominant permutation as the amount of O_2 available decreases. Although the residual species permutations are chosen automatically using the method outlined in Algorithms 1 - 4 it is still reassuring to see that there is a general reflection of the physics in the residual species permutations. Although the candidate final residuals included O, H and C, it is interesting to see that none of them are ever chosen to be the final residual, meaning that they always contribute less to their respective elements than O_2 or H_2 . No negative residual species mass fractions (final or otherwise) are ever encountered, which is testament to the high accuracy

of using a single MLP for each species, even if predicting the output directly as in this case.

Table 5.5: Percentage frequency of residual species permutations at different equivalence ratios for 1D premixed flames with NEDs for GRI 1.2.

			Percentage frequency [%]				
Permutation			$\phi = 0.6$	$\phi = 0.8$	$\phi = 1.0$	$\phi = 1.2$	$\phi = 1.4$
CH ₄	H ₂ O	O ₂	9.4	10.0	8.2	10.3	18.3
CO	H ₂ O	O ₂	11.1	3.8	6.5	5.1	5.1
CO	H ₂ O	H ₂	-	-	-	8.1	76.6
CO ₂	H ₂ O	O ₂	79.4	86.2	85.3	0.2	-
CO ₂	H ₂ O	H ₂	-	-	-	76.3	-
C ₂ H ₂	H ₂ O	O ₂	0.1	-	-	-	-

Table 5.6 shows the average CPU time to compute the reaction step across the domain over simulations at $\phi = 1.0$ with each mechanism for each reaction source term computation method. Reaction source term computations with the ANNs are on average 30x faster than DI with the VODE solver for GRI 1.2. This is faster than the speed-up reported in Chapters 3 and 4 for the GRI 1.2 mechanisms in the turbulent flame cases, which is a consequence of the 1D premixed flame simulations having a large proportion of the domain at a high temperature. Enforcing the conservation laws increases the required CPU time by a very small amount, meaning that the ANNs with the conservation method are 29x faster than the VODE solver. This shows that the time required for the conservation is very small compared to the time required for the ANN inference. USC Mech II is more complex than the GRI 1.2 mechanism meaning that the DI simulations are significantly slower, and so the speed-up obtained from using the ANNs is larger at approximately 56x. Again, the time required to enforce the conservation laws for USC Mech II is very small, meaning that the simulations using ANNs with conservation are approximately 54x faster than the DI simulations.

Table 5.6 shows the average CPU time required to perform reaction step calculations in the 1D premixed flame simulations with each integration method for the GRI 1.2 mechanism. The time for training data generation or ANN training is not considered in Table 5.6 because the focus is not on the generalisation capability of the ANNs but is instead on assessing the effects of enforcing the conservation laws on ANN performance. For GRI 1.2, reaction step calculations in the 1D premixed flames with ANNs are on average 30x faster than using the VODE solver. Enforcing the conservation laws with Algorithms 1-4 incurs a negligible computational cost compared to the ANN inference time, and remains on average 29x faster than the VODE solver.

Table 5.6: Average time taken to perform one reaction step in GRI 1.2 1D premixed flame simulations with NEDs at $\phi = 1.0$ with each integration method using a single Intel Xeon 6132 2.6 GHz CPU.

Method	CPU time [s]	Speed up
DI	0.7228	-
ANNs	0.0245	30x
ANNs with conservation	0.0251	29x

5.2.2 Results for C_3H_8 /air flames with USC Mech II

Table 5.7 shows the root mean square error in each of the quantities that should be conserved across all of the reacting points throughout simulations with NEDs at $\phi = 1.0$ for USC Mech II when using the ANNs with and without conservation. As in Table 5.4 for the GRI 1.2 mechanism, the conservation laws are satisfied to machine precision for every reacting point throughout simulations at any equivalence ratio.

Table 5.7: Root mean square error in conserved quantities across the reaction step over all reacting points throughout simulations with non-equal diffusivities at $\phi = 1.0$ for USC Mech II.

Quantity	ANNs RMSE	ANNs with conservation RMSE
Y_O^e	5.74×10^{-7}	1.99×10^{-18}
Y_H^e	8.08×10^{-6}	3.35×10^{-18}
Y_C^e	3.06×10^{-7}	9.60×10^{-19}
$1 - Y_{N_2}$	4.95×10^{-9}	8.65×10^{-20}

Figures 5.7 and 5.8 show CO and H_2 mass fraction profiles at the end of the USC Mech II simulations with the different reaction source term integration methods. The agreement between the profiles from the ANN simulations with conservation and the DI profiles is not as good as in the GRI 1.2 simulations, as USC Mech II is a more complicated mechanism than GRI 1.2 and so the overall ANN accuracy is lower. However, the agreement between the accuracy of the ANNs without conservation is much lower than the accuracy of the ANNs with conservation, which is particularly noticeable in Figures 5.7 and 5.8 at $\phi = 0.75$, where the ANNs without conservation do not accurately reproduce the peak value, peak location or the general shape of the DI profiles. Again, this shows that regardless of the accuracy level of the ANNs themselves, enforcing the conservation increases the overall ANN simulation accuracy. As with the GRI 1.2 mechanism, if better agreement between the ANN

simulations and the DI was desired then architectures which increase ANN accuracy like the MMLP method could be used.

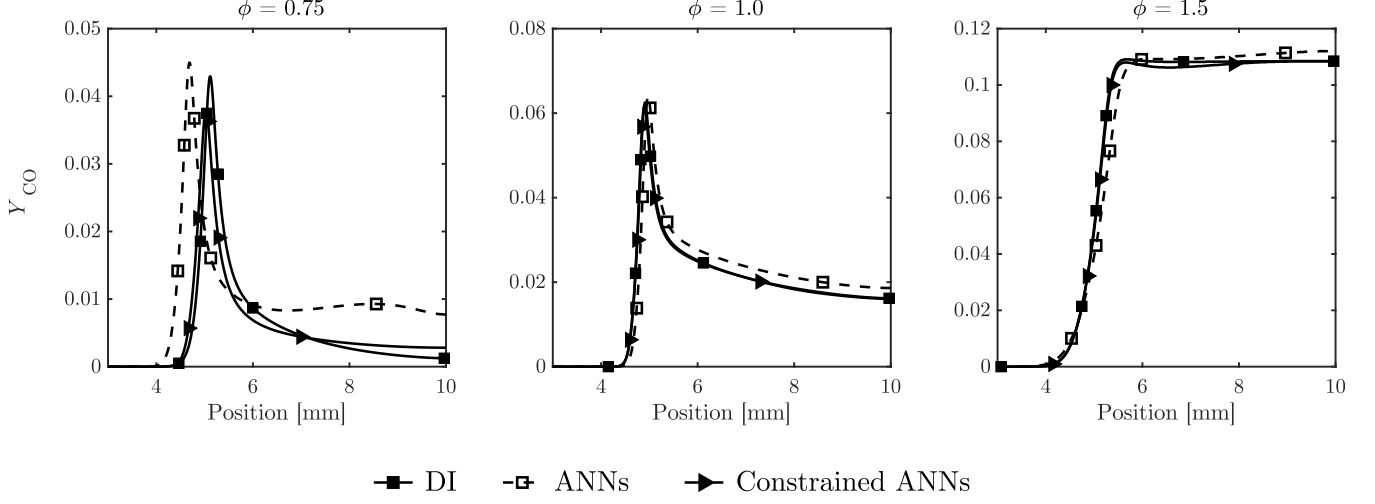


Figure 5.7: CO mass fraction profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

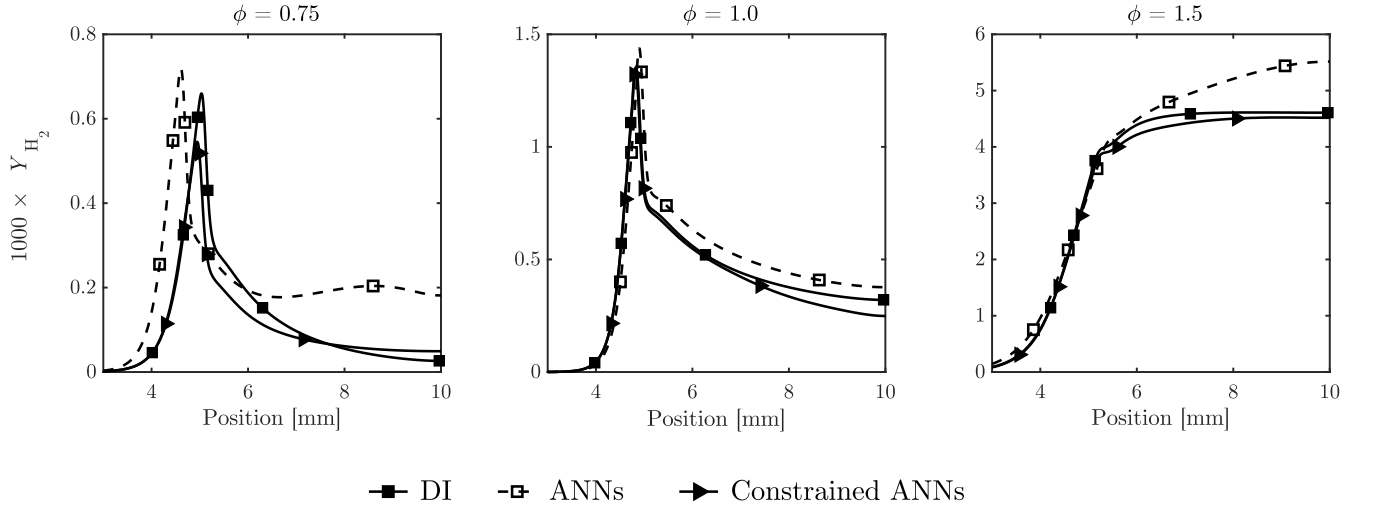


Figure 5.8: H₂ mass fraction profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

The molar H/C and O/N profiles at the end of the USC Mech II simulations with the different reaction source integration methods are shown in Figures 5.9 and 5.10. As the the general agreement between the ANN and DI species profiles is worse than in the GRI 1.2 simulations, differential diffusion means there are more noticeable differences between the element ratio profiles. However, it is again clear in Figures 5.9 and 5.10 that enforcing the conservation increases the simulation accuracy generally and prevents the largest deviations from the DI composition space, for example as seen in the O/N profile at $\phi = 1.5$.

Table 5.5 shows the frequency of the residual species permutations recorded

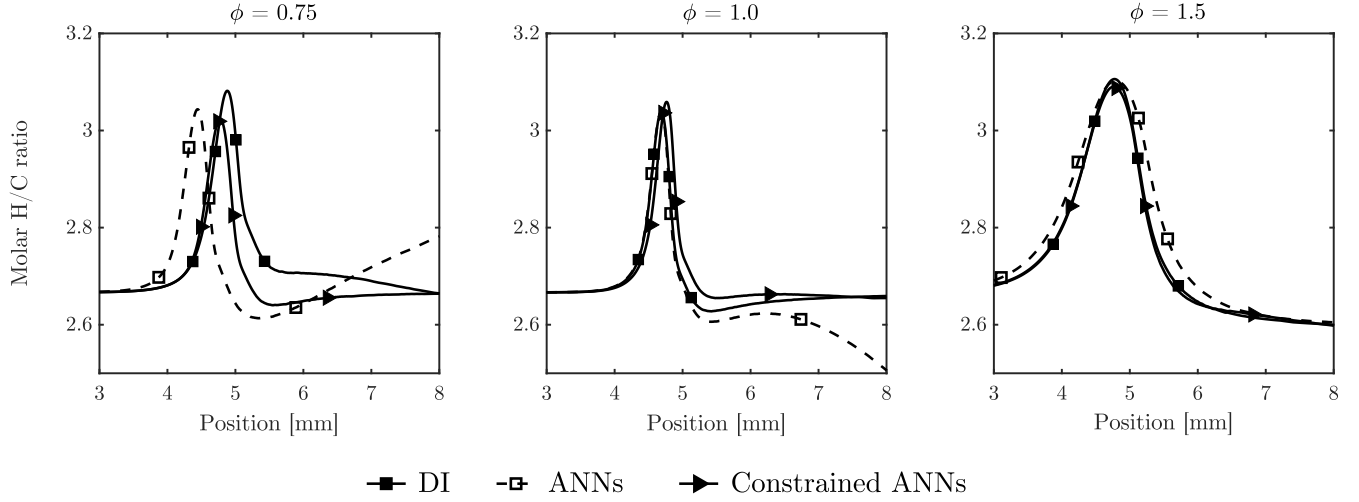


Figure 5.9: Molar H/C ratio profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

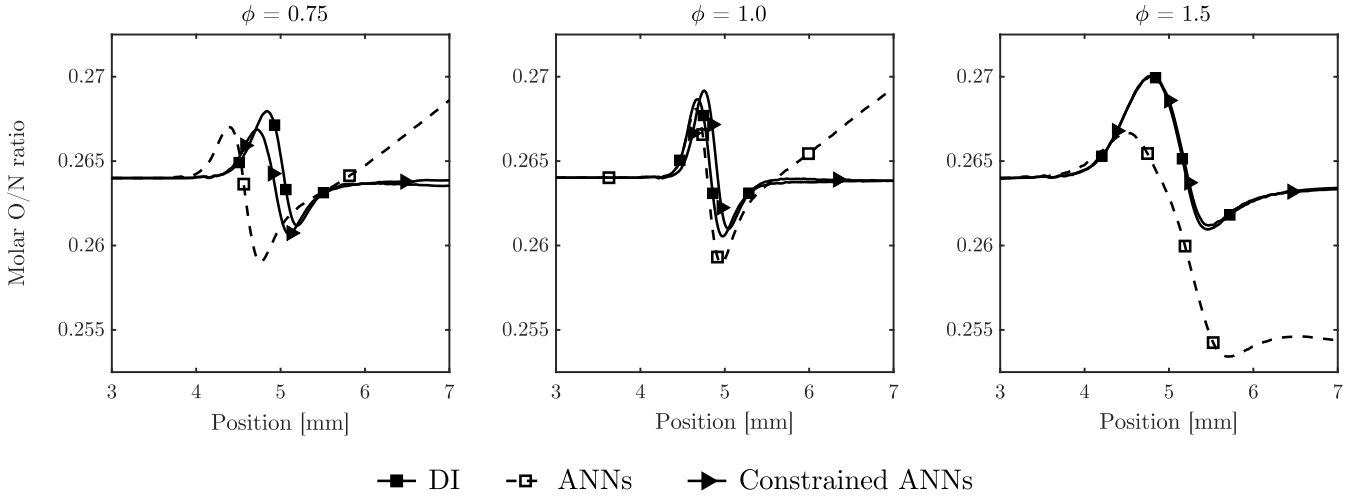


Figure 5.10: Molar O/N ratio profiles at the end of USC Mech II 1D premixed flame simulations with NEDs using different methods for reaction source term integration.

throughout the USC Mech II premixed flame simulations with NEDs at various equivalence ratios. Although there are roughly 4x as many species in USC Mech II, all of the residual species permutations use species which were also used in the GRI 1.2 mechanism (with C_3H_8 as the fuel instead of CH_4). This agrees with prior knowledge of major and minor species, which is that generally the major species account for a large majority of the elemental mass while the minor species have low mass fractions and so contribute little to the element mass fractions. As in the GRI 1.2 case, as the mixture moves from fuel lean to stoichiometric to fuel rich, permutations using O_2 become less common and permutations using H_2 become dominant.

At no point in any of the USC Mech II simulations were negative residual species predicted. This means that in these simulations Algorithm 4 was never required.

Table 5.8: Percentage frequency of residual species permutations at different equivalence ratios for 1D premixed flames with NEDs for USC Mech II.

			Percentage frequency [%]				
Permutation			$\phi = 0.7$	$\phi = 1.0$	$\phi = 1.5$	$\phi = 2.0$	$\phi = 2.5$
C ₃ H ₈	H ₂ O	O ₂	2.6	0.3	1.3	9.2	-
C ₃ H ₈	H ₂	O ₂	-	-	-	3.4	-
C ₃ H ₈	O ₂	H ₂	-	-	-	-	1.5
CO	H ₂ O	O ₂	3.7	7.2	6.1	5.8	-
CO	H ₂	O ₂	-	-	-	0.8	-
CO	H ₂ O	H ₂	-	-	92.6	71.2	78.2
CO	O ₂	H ₂	-	-	-	9.5	20.3
CO ₂	H ₂ O	O ₂	93.7	89.5	-	-	-
CO ₂	H ₂ O	H ₂	-	3.0	-	-	-

This is partly due to the way the method is formulated, as choosing the residual species which contribute the most to their respective elements reduces the change of encountering negative residuals. However depending on the species available in the mechanism and the ANN accuracy it is still possible to encounter negative residuals, particularly in the final residual species. Therefore, the negative final species rectification proposed in Algorithm 4 should still be kept as part of the method, even though it has not been used in this case.

Table 5.9 shows the average CPU time required to perform reaction step calculations in the 1D premixed flame simulations with each integration method for the USC Mech II mechanism. The USC Mech II mechanism is more complex than GRI 1.2 so the speed-up when using ANNs is larger: on average 56x faster when using ANNs compared to the VODE solver. Again the time for the conservation is very small compared to the ANN inference time, and reaction source term computations using ANNs with conservation are still 54x faster than those with the VODE solver. The results from this section show that applying the conservation laws to the ANN predictions gives significant accuracy benefits with very little additional computational cost.

Table 5.9: Average time taken to perform one reaction step in USC Mech II 1D premixed flame simulations with NEDs at $\phi = 1.0$ with each integration method using a single Intel Xeon 6132 2.6 GHz CPU.

Method	CPU time [s]	Speed up
DI	5.924	-
ANNs	0.106	56x
ANNs with conservation	0.110	54x

5.3 Application to random data generation

Although the conservation method has been explained and applied in the context of tabulation of reaction mechanisms with ANNs, it can be applied to any composition point which undergoes some process. For example, the method can be readily applied to data randomly generated from canonical combustion data. In this case the random data generation procedure based on a composition can be expressed as:

$$Y_i^r(t) = \text{RANDGEN}[h(t), Y_i(t)], \quad (5.9)$$

where RANDGEN refers to the random data generation method outlined in Chapter 2. The conservation method can then be applied to Y_i^r to generate data which satisfies the constraints which are present in the canonical combustion problem data. In this way, random data can be generated which shares the element mass fractions of the original data. This also means that the element ratios present in the original data will remain. The advantage of this is that the compositions generated still contain randomness and so do not exactly obey the equations of the canonical combustion data, but do satisfy constraints on the composition space which may be encountered in the application.

The specifics of the method used depends on which elements are considered. In combustion applications the main point of difference is whether nitrogen chemistry is included or not. If nitrogen chemistry is not included and N_2 is the only nitrogen containing species, then the mass fraction of diatomic nitrogen must remain unchanged as changing it would change the element mass fractions. If nitrogen chemistry is included, then the mass fraction of N_2 can be changed, as long as the mass fractions of the other nitrogen containing species change to conserve the nitrogen element. For hydrocarbon fuels burning in air, it is likely that the mass fraction of N_2 will be by far the largest of the nitrogen containing species and altering any

other nitrogen containing species on the same scale as N_2 will lead to unrealistic compositions. Therefore the method proposed alters the other nitrogen containing species mass fractions first, before using N_2 to ensure that the nitrogen element mass fraction is conserved. The method can be readily adapted if burning fuels with a significant N content, such as ammonia.

The random data generation procedure with conservation of the element mass fractions from the original data for hydrocarbon fuel/air mechanisms is outlined in Algorithm 5. The calculation of the random enthalpy value is identical to the random data generation procedure without conservation. Then, the random mass fractions of the species which are not N_2 are generated (including the other nitrogen containing species, if present). The 'randomly generated' mass fraction of N_2 is taken as the mass fraction of N_2 in the original data. The normalisation of the species mass fractions in the original random data generation method is unnecessary as the conservation method will guarantee mass and element conservation. Algorithms 2 - 4 are then applied to the randomly generated composition. The target element mass fractions are the element mass fractions from the original data. In Algorithm 2 the predicted element mass fractions are not calculated but are replaced with the target element mass fractions.

Algorithm 5 is applied to two mechanisms: GRI 1.2, which does not include nitrogen chemistry, and GRI 3.0, which does include nitrogen chemistry. The data for the GRI 1.2 mechanism is the 1D premixed flame data with EDs collected in Chapter 4 for the SWB simulations, while the data for the GRI 3.0 mechanism is generated from an ensemble of laminar flamelets in an analogous manner to the canonical combustion problem data generated for the Sydney L simulations in Chapter 3. Figure 5.11 shows examples of the resulting random data after Algorithm 5 is applied to the GRI 1.2 mechanism data. The data is visually indistinguishable from the random data generated for the SWB simulations, but shares the same mass fractions as the original premixed flame data, meaning that the element ratios are practically exact and acceptable element ratio windows do not have to be predetermined. Table 5.10 shows the root mean square error between the element mass fractions in the randomly generated data and the canonical combustion data for the GRI 1.2 mechanism. The nitrogen element mass fraction is exact as the N_2 mass fraction is unchanged by the conservation process, while the other elements are conserved to machine precision.

Figure 5.12 shows the random data generated by applying Algorithm 5 to the GRI 3.0 laminar flamelet data. As GRI 3.0 includes nitrogen chemistry the nitrogen element must be included in the conservation process and, as Table 5.11 shows, is conserved to machine precision. Because the mass fraction of N_2 is much higher than any of the other nitrogen containing species, it is always chosen (automatically)

as the residual species to conserve the nitrogen element. This means its value is modified and so the N_2 mass fraction in the random data is not identical to the N_2 mass fraction in the canonical combustion data, as it was with the GRI 1.2 mechanism.

Algorithm 5 Random data generation with conservation of element mass fractions

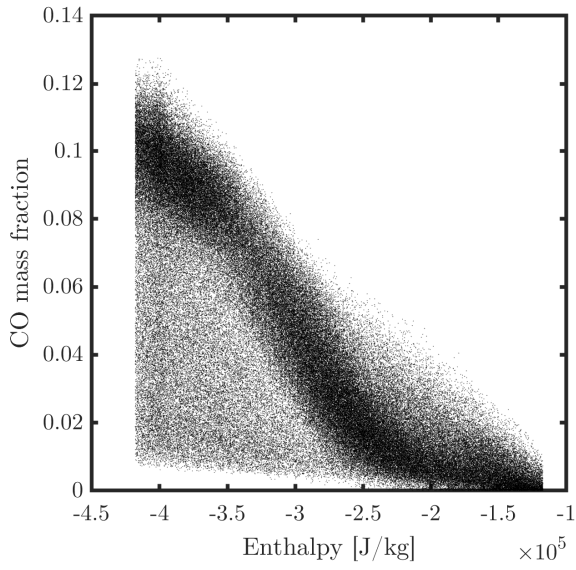
Require: h, Y_i ▷ Canonical combustion problem data point
 $h^r = h + a_1 \frac{\max(h) - \min(h)}{b_1}$ ▷ Random enthalpy generated as normal
for $i = 1, \dots, N_{sp}$ **do** ▷ Perform randomisation
 if Y_i is not Y_{N_2} **then** ▷ If not N_2
 if $Y_i > 1 \times 10^{-20}$ **then**
 $Y_i^r = 10^{(L + \frac{a_{3,i}L}{b_3})}$ where $L = \log_{10}(Y_i)$
 else
 $Y_i^r = Y_i$
 end if
 end if
 $Y_{N_2}^r = Y_{N_2}$
end for
Use Algorithm 2 ▷ Replace Z_j^b with Z_j
Use Algorithm 3
Use Algorithm 4 (if necessary)
return $h^r, Y_i^{r,con}$ ▷ Random data point with conserved element mass fractions

Table 5.10: Root mean square error between element mass fractions of randomly generated data and canonical combustion problem data for the GRI 1.2 mechanism.

Quantity	RMSE
Z_O	4.23×10^{-17}
Z_H	4.94×10^{-18}
Z_C	1.16×10^{-17}
Z_N	0

Table 5.11: Root mean square error between element mass fractions of randomly generated data and canonical combustion problem data for the GRI 3.0 mechanism.

Quantity	RMSE
Z_O	5.53×10^{-18}
Z_H	5.65×10^{-17}
Z_C	1.22×10^{-17}
Z_N	7.16×10^{-17}



(a) Enthalpy against CO mass fraction.

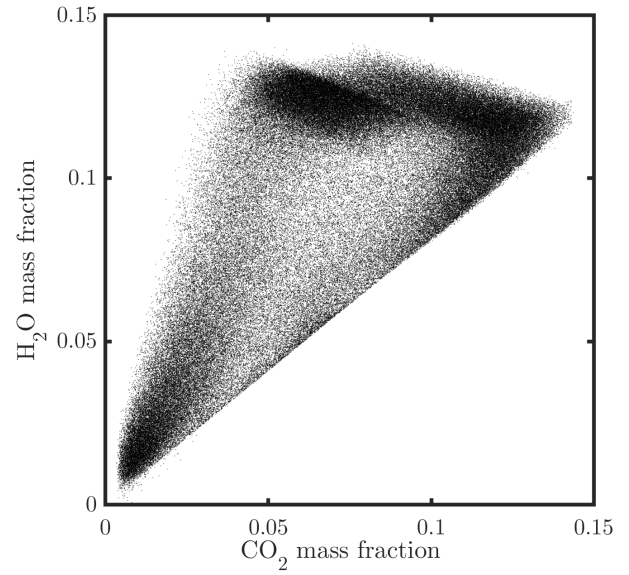

 (b) CO₂ mass fraction against H₂O mass fraction.

Figure 5.11: Composition space coverage of GRI 1.2 randomly generated data which shares the same element mass fractions as the 1D premixed flame EDs data.

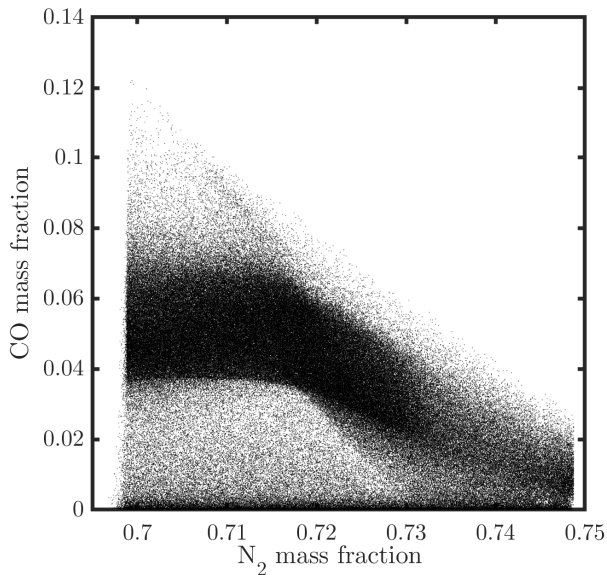
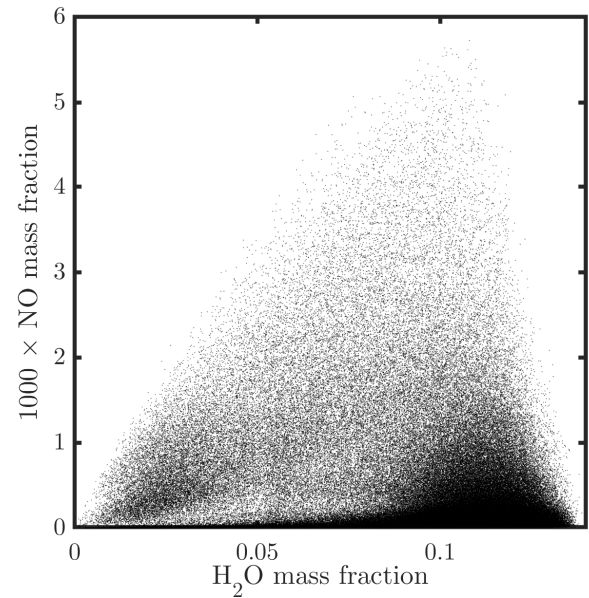

 (a) N₂ mass fraction against CO mass fraction.

 (b) H₂O mass fraction against NO mass fraction.

Figure 5.12: Composition space coverage of GRI 3.0 randomly generated data which shares the same element mass fractions as the 1D laminar flamelet data.

5.4 Application to reaction rate predictions

In Section 5.2 the focus was on applying the conservation laws to composition points predicted by ANNs. This is because the ultimate aim of the method is to apply ANNs with enforced conservation to transported PDF simulations using the fractional step method, where reaction source terms can be isolated and treated as a system of ODEs. However, in other applications (for example Owoyele and Pal [38]) ANNs are used to directly predict instantaneous reaction rates, which are then integrated. If rates are to be predicted by ANNs the constraint arising from mass conservation is given by Equation 5.10, while the constraints arising from element conservation are given by Equation 5.11:

$$\sum_{i=1}^{N_{sp}} \dot{\omega}_i^*(t) = 0 \quad (5.10)$$

$$\sum_{i=1}^{N_{sp}} A_{ji} \dot{\omega}_i^*(t) = 0_j \quad (5.11)$$

In contrast to the species mass fraction, the instantaneous reaction rates are unbounded. This makes it difficult to ensure that species mass fractions computed after integration of the predicted rates remain bounded. Therefore, there are two parts to conservation when predicting rates. The first part is to enforce the element conservation laws in Equation 5.11 (and so automatically Equation 5.10) in an analogous manner to those in Equation 5.3. The residual species for which the reaction rates are calculated can be selected in a very similar way to that outlined in Algorithm 2, except now the species which have the largest contributions to the rates of change of their respective elements should be selected.

Algorithm 3 can then be followed, except it is the rate of change of the chosen residual species which is calculated to make sure Equation 5.11 is satisfied. The second part of the conservation process when predicting rates is that if negative species mass fractions are encountered in a composition point after the integration of the rates, then Algorithms 2 - 4 can be used directly on the newly found composition point to satisfy the conservation laws.

Chapter 6

Conclusions

Tabulating thermochemistry with artificial neural networks has the potential to alleviate the computational bottleneck of combustion simulations by reducing the time required to perform the integration of the chemical kinetics of the reacting system. Despite this potential, a number of research challenges remain, relating to the generality, accuracy, and physical consistency of ANN tabulation approaches. The present work has aimed to tackle these research challenges, and in this chapter the main points of advancement are summarised along with suggestions for further work which have been unlocked by the work performed.

6.1 Summary of thesis achievements

A method has been developed which is capable of producing ANNs which are general enough to be applied to families of combustion problems with high prediction accuracy. This method first uses canonical combustion problems to generate data covering the approximate region of composition space to be accessed in a simulation. The generality of this canonical combustion problem data is improved using a random data generation method. This random data generation method expands the composition space coverage of the data, particularly for the hard to anticipate minor species, while still producing combustion relevant data. If desired, this random data generation method is capable of modifying certain physical properties of the data such as the ratios between element mass fractions. This increases the robustness of ANNs used in simulations where mass and element conservation are not enforced and therefore increases the accuracy of predictions.

To further increase the accuracy of ANN predictions without undue specialisation to a given test case, it has been shown that subdivision of the composition space

covered by the training data is unnecessary. Instead, it has been found that a significant increase in prediction accuracy can be obtained by performing predictions of each species separately. This has the added benefit of keeping the number of ANN parameters required low, and therefore renders quasi-second order and second order training algorithms, such as the Levenberg-Marquardt algorithm, feasible. The Levenberg-Marquardt algorithm is capable of producing more accurate ANNs than first order training algorithms but is only feasible for small to medium sized ANNs.

The accuracy of the developed method using random data generation and separate MLPs for each species was first demonstrated by performing LES-PDF simulations of the non-premixed CH_4 /air jet flame, Sydney flame L with the full GRI 1.2 mechanism. The high prediction accuracy of the ANNs produced in this way allowed for the full transient behaviour of Sydney flame L to be simulated, from the first injection of the jets into the domain to statistical steady state. This is in contrast to Franke et al. [55], where the limited accuracy meant that it was only possible to simulate the flame with ANNs once it had reached statistical steady state. Statistics obtained from LES-PDF simulations using the ANNs and DI respectively for reaction source term calculations were found to be in excellent agreement for major species and good agreement for minor species, while a reaction source term calculation speed-up of 18x was obtained.

After the successful demonstration of the ANN method in the context of non-premixed flames, the range of applications of ANN thermochemistry tabulation was expanded by simulating a turbulent premixed jet flame. The obvious canonical combustion problem choice of 1D laminar flamelets means that there are numerous applications of ANN thermochemistry tabulation to non-premixed flames. However, for premixed flames the canonical combustion problem is more difficult to select, meaning that there are few examples of premixed flame simulations using ANNs. The sampling deficiencies present in 1D laminar premixed flames with equal diffusivities were overcome using the random data generation method, allowing the application of ANNs, again to the full GRI 1.2 mechanism, for the simulation of the stratified swirl-stabilised bluff-body CH_4 /air Cambridge/Sandia swirl burner.

Recirculation zones near the Cambridge/Sandia swirl burner bluff-body require very high prediction accuracy to prevent the accumulation of errors affecting downstream predictions, particularly for minor species. This was achieved using the multiple MLP method, where as well as predicting each species individually, the output ranges for each species are automatically divided and a separate MLP is used for each. The MMLP method allows excellent prediction accuracy for states from ignition up to equilibrium, even for minor species. This is reflected in the excellent agreement between statistics obtained from LES-PDF simulations of SWB6 when using the ANNs and DI respectively for reaction source term computations for

both major and minor species, while obtaining a reaction source term computation speed-up of approximately 14x. Having been validated in the SWB6 flame, the same ANNs were applied to an LES-PDF simulation of the SWB5 flame, with excellent agreement obtained between statistics from the simulation and those recorded by experiment, showing that the ANNs generated using this method are general enough to be applied to families of problems.

While the simulations of Sydney flame L and SWB5 and SWB6 showed the capabilities of the proposed ANN method, in none of the simulations were the ANNs physically consistent with the chemical mechanism they were imitating. To remedy this, a method of enforcing the mass and chemical element conservation laws as well as the composition space bounds has been formulated. This method enforces the conservation laws to machine precision by calculating new values of certain species once ANN predictions have been made.

The effects of enforcing the conservation laws on ANN prediction accuracy have been assessed through the simulation of 1D premixed CH_4/air and $\text{C}_3\text{H}_8/\text{air}$ flames at equivalence ratios across the flammable range. These simulations show that as well as ensuring physical consistency, enforcing the conservation laws improves ANN accuracy compared to ANN simulations without enforced conservation. For the CH_4/air flames using the full GRI 1.2 mechanism a reaction source term computation speed-up without enforced conservation of 30x relative to DI is obtained, while the speed-up with enforced conservation is 29x. For the $\text{C}_3\text{H}_8/\text{air}$ flames using the full USC Mech II mechanism a reaction source term computation speed-up without enforced conservation of 56x relative to DI is obtained, while the speed-up with enforced conservation is 54x. This demonstrates that the conservation method computationally inexpensive and so physical consistency, and the increase in accuracy that goes with it, can be obtained at very little cost. The conservation method can be easily appended to any existing or new ANN architectures and further, it can actually be applied to any process manipulating compositions, including the random data generation method.

Through the formulation of the random data generation method, the multiple MLP method and the method for enforcing the conservation laws, this thesis has detailed how the generality, accuracy and physical consistency of current ANN approaches has been advanced. This advancement means that ANN tabulated thermochemistry can be applied more reliably to alleviate the computational bottleneck present in simulation methods requiring the real time integration of the chemical kinetics, which includes DNS, the LES-PDF method, laminar flame computations, the conditional moment closure method, the multiple mapping conditioning method, thickened flame methods, the linear eddy model and partially stirred reactors. Additionally, these advancements pave the way for future work which can be performed,

which is detailed in the next section.

6.2 Suggestions for future work

The first opportunity for further work is the application of the current method to large-scale reacting flow simulations using complex mechanisms. The turbulent flame simulations in this work both used the GRI 1.2 mechanism, consisting of 31 species and 175 reactions. The largest mechanism used in this work was the USC Mech II mechanism, consisting of 111 species and 784 reactions, which was used in 1D premixed flame simulations. The simulation of large-scale turbulent flames with mechanisms of the size of USC Mech II is prohibitively expensive when using DI, but can be realistically performed with the proposed ANN method.

This means that simulations of industrially relevant combustion equipment are feasible with complex chemistry. This is of importance to the aviation sector where the fuel used is kerosene, with typical mechanisms for kerosene surrogates containing several hundred species and thousands of reactions. Such simulations often occur at pressures up to 20 bar, while the simulations performed in this work were at atmospheric pressure, meaning that the ANN method will need to be expanded to include pressure dependence. It may also be desirable to use variable time steps in such simulations, in which case several sets of ANNs will need to be trained, each able to predict the reaction behaviour over a different time interval.

With the proposed ANN method, other complex mechanisms, such as those which are designed for the prediction of pollutants such as soot can be simulated at modest computational cost. The physical consistency afforded by the conservation method means that reduced mechanisms from which the full composition can be generated, such as those obtained when using RCCE, can also be reliably simulated.

Although the random data generation method is capable of improving composition space coverage from canonical combustion problem data, applications for which it is difficult to identify a suitable canonical combustion problem remain. Therefore, on-the-fly training has the potential to expand the range of ANN applications further. While Chi et al. [74] have performed an initial exploration of on-the-fly training, their approach suffers from the drawback that training is required every few hundred iterations during a simulation. This is because they were unable to enforce the conservation laws, and so properties of the composition space being accessed were also changing as the simulation progressed. With the conservation method detailed here, these properties can be guaranteed to remain unchanged even when using the ANNs. This means that on-the-fly training can occur only once at the

start of the simulation, increasing the potential speed-up compared to the method of Chi et al.. Alternatively if the anticipated composition space is approximately known, a combination of ahead of time and on-the-fly training could be performed, to tune the ANNs for the specific application with modest computational cost.

Appendix A

Comparison of the moments arising from the modelled probability density function equation and the stochastic fields

In this appendix, the moments arising from the modelled PDF equation and from the equation describing the stochastic fields are derived.

A.1 The modelled probability density function equation

The PDF, $\bar{\mathcal{P}}_{\text{sgs}}(\psi; \mathbf{x}, t)$, is considered as a good generalised function with the following properties:

$$\lim_{\psi \rightarrow \pm\infty} \bar{\mathcal{P}}_{\text{sgs}}(\psi; \mathbf{x}, t) = 0; \quad \lim_{\psi \rightarrow \pm\infty} \frac{\partial \bar{\mathcal{P}}_{\text{sgs}}(\psi; \mathbf{x}, t)}{\partial \psi} = 0, \quad (\text{A.1})$$

which implies that in physical space the problem is bounded. The n^{th} moment of the scalar $\varphi(\mathbf{x}, t)$ is:

$$M_n(\mathbf{x}, t) = \int_{-\infty}^{\infty} \psi^n \bar{\mathcal{P}}_{\text{sgs}}(\psi; \mathbf{x}, t) d\psi. \quad (\text{A.2})$$

To calculate the equation for the moments for a constant density fluid, Equation

2.66 can be multiplied term by term by ψ^n and then integrated between the limits $\pm\infty$. The integration for each of the terms in Equation 2.66 is as follows:

Term 1:

$$\rho \frac{\partial \bar{P}_{\text{sgs}}}{\partial t} \rightarrow \int_{-\infty}^{\infty} \psi^n \rho \frac{\partial \bar{P}_{\text{sgs}}}{\partial t} d\psi = \rho \frac{\partial}{\partial t} \int_{-\infty}^{\infty} \psi^n \bar{P}_{\text{sgs}} d\psi = \rho \frac{\partial M_n}{\partial t}$$

Term 2:

$$\rho \bar{u}_i \frac{\partial \bar{P}_{\text{sgs}}}{\partial x_i} \rightarrow \int_{-\infty}^{\infty} \psi^n \rho \bar{u}_i \frac{\partial \bar{P}_{\text{sgs}}}{\partial x_i} d\psi = \rho \bar{u}_i \frac{\partial}{\partial x_i} \int_{-\infty}^{\infty} \psi^n \bar{P}_{\text{sgs}} d\psi = \rho \bar{u}_i \frac{\partial M_n}{\partial x_i}$$

Term 3:

$$\begin{aligned} \frac{\partial^2}{\partial \psi^2} \left(\bar{\mathcal{P}}_{\text{sgs}} \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \right) &\rightarrow \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \int_{-\infty}^{\infty} \psi^n \frac{\partial^2}{\partial \psi^2} \bar{\mathcal{P}}_{\text{sgs}} d\psi = \\ &\frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \left[\psi^n \frac{\partial}{\partial \psi} \bar{\mathcal{P}}_{\text{sgs}} \right]_{-\infty}^{\infty} - \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \int_{-\infty}^{\infty} n \psi^{n-1} \frac{\partial}{\partial \psi} \bar{\mathcal{P}}_{\text{sgs}} d\psi = \\ &-\frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \left[\psi^{n-1} \bar{\mathcal{P}}_{\text{sgs}} \right]_{-\infty}^{\infty} + \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 \int_{-\infty}^{\infty} n(n-1) \psi^{n-2} \bar{\mathcal{P}}_{\text{sgs}} d\psi = \\ &\frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x} \right)^2 n(n-1) M_{n-2} \end{aligned}$$

Term 4:

$$\begin{aligned} \frac{\partial}{\partial x_i} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial \bar{\mathcal{P}}_{\text{sgs}}}{\partial x_i} \right) &\rightarrow \int_{-\infty}^{\infty} \psi^n \frac{\partial}{\partial x_i} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial \bar{\mathcal{P}}_{\text{sgs}}}{\partial x_i} \right) d\psi = \\ \frac{\partial}{\partial x_i} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial}{\partial x_i} \left(\int_{-\infty}^{\infty} \psi^n \bar{\mathcal{P}}_{\text{sgs}} d\psi \right) \right) &= \frac{\partial}{\partial x_i} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial M_n}{\partial x_i} \right) \end{aligned}$$

Term 5:

A Comparison of the moments arising from the modelled probability density function equation and the stochastic fields

$$\begin{aligned}
\frac{C_d}{\tau_{\text{sgs}}} \frac{\partial}{\partial \psi} [\rho(\psi - \bar{\varphi}) \bar{\mathcal{P}}_{\text{sgs}}] &\rightarrow \frac{C_d}{\tau_{\text{sgs}}} \int_{-\infty}^{\infty} \psi^n \frac{\partial}{\partial \psi} [\rho(\psi - \bar{\varphi}) \bar{\mathcal{P}}_{\text{sgs}}] d\psi = \\
&[\psi^n \rho(\psi - \bar{\varphi}) \bar{\mathcal{P}}_{\text{sgs}}]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} n\psi^{n-1} (\psi - \bar{\varphi}) \bar{\mathcal{P}}_{\text{sgs}} d\psi = \\
&- \int_{-\infty}^{\infty} n\psi^n \bar{\mathcal{P}}_{\text{sgs}} d\psi + \int_{-\infty}^{\infty} n\psi^{n-1} \bar{\varphi} \bar{\mathcal{P}}_{\text{sgs}} d\psi = -nM_n + nM_{n-1}\bar{\varphi}
\end{aligned}$$

Term 6:

$$\begin{aligned}
\frac{\partial}{\partial \psi} \rho \bar{\mathcal{P}}_{\text{sgs}} \dot{\omega} &\rightarrow \int_{-\infty}^{\infty} \psi^n \frac{\partial}{\partial \psi} \rho \bar{\mathcal{P}}_{\text{sgs}} \dot{\omega} = \\
&[\rho \psi^n \dot{\omega} \bar{\mathcal{P}}_{\text{sgs}}]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} n\rho \psi^{n-1} \dot{\omega} \bar{\mathcal{P}}_{\text{sgs}} d\psi = -n\rho \overline{\dot{\omega} \psi^{n-1}}
\end{aligned}$$

Collecting all of the terms, the following equation for the n^{th} moment results:

$$\begin{aligned}
\rho \frac{\partial M_n}{\partial t} + \rho \bar{u}_i \frac{\partial M_n}{\partial x_i} + n(n-1)M_{n-2} \frac{\mu}{\sigma} \left(\frac{\partial \bar{\varphi}}{\partial x_i} \right)^2 \\
= \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial M_n}{\partial x_j} \right] - n\rho \frac{C_d}{\tau_{\text{sgs}}} (M_n - M_{n-1}\bar{\varphi}) + n\rho \overline{\dot{\omega} \psi^{n-1}}.
\end{aligned}$$

A.2 The stochastic fields

The moments arising from the stochastic fields can be obtained using standard methods. The increment of a stochastic field $\xi(\mathbf{x}, t)$, given by Equation 2.68, and repeated here is:

$$\begin{aligned}
\rho d\xi = & \underbrace{-\rho \bar{u}_j \frac{\partial \xi}{\partial x_j} dt + \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial \xi}{\partial x_j} \right] dt - \frac{\rho C_d}{\tau_{\text{sgs}}} (\xi - \bar{\varphi}) dt + \rho \dot{\omega} dt}_{A(\xi)dt} \\
& + \underbrace{\sqrt{2 \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}}} \frac{\partial \xi}{\partial x_j} dW_j}_{B(\xi)dW}
\end{aligned} \tag{A.3}$$

The stochastic term involves the increment of a Wiener process, which is represented as the product of η (a dichotomic variable with $\mathcal{N} \sim [-1, 1]$) and the square root of the time-step. It is thus of $\sim \mathcal{O}(dt^{1/2})$. The only constraint on η is its property $\mathcal{N} \sim [-1, 1]$. The stochastic process so defined is not differentiable in time and it is therefore necessary to resort to a time expansion around a time instance, t . The derivative of the n^{th} moment is calculated assuming an infinite number of stochastic fields N_{inf} :

$$\frac{dM_n}{dt} = \lim_{N_{\text{inf}} \rightarrow \infty} \frac{1}{N_{\text{inf}}} \sum_{i=1}^{N_{\text{inf}}} \frac{(\xi^i(\mathbf{x}, t + dt))^n - (\xi^i(\mathbf{x}, t))^n}{dt}. \tag{A.4}$$

While a single realisation of the scalar field does not have any physical significance, in an averaged formulation the stochastic fields should share the same one-point moments as those given by the modelled PDF equation.

To calculate $(\xi(\mathbf{x}, t + dt))^n$ it is necessary to use a Taylor-Itô expansion [87] of the function $f(\xi) = (\xi(\mathbf{x}, t))^n$. The Itô formula/Martingale representation theorem in differential form allows the expansion of the function and yields:

$$\rho df(\xi) = \left[A(\xi) \frac{\partial f(\xi)}{\partial \xi} + \frac{1}{2} B(\xi)^2 \frac{\partial^2 f(\xi)}{\partial \xi^2} \right] dt + \left[B(\xi) \frac{\partial f(\xi)}{\partial \xi} \right] dW + R, \tag{A.5}$$

where R is the remainder containing higher order terms. Substituting $f(\xi) = (\xi(\mathbf{x}, t))^n$ into Equation A.5 and neglecting $R \sim \mathcal{O}(dt^{3/2})$:

$$\rho d\xi^n = \left[nA(\xi)\xi^{n-1} + \frac{1}{2}n(n-1)B(\xi)^2\xi^{n-2} \right] dt + \left[nB(\xi)\xi^{n-1} \right] dW, \tag{A.6}$$

where the superscript denotes the power n . Once the increment is calculated, $\xi^n(\mathbf{x}, t + dt)$ can be obtained from $\xi^n(\mathbf{x}, t + dt) = \xi^n(\mathbf{x}, t) + d\xi^n$. On substitution into Equation A.4:

$$\rho \frac{dM_n(\mathbf{x}, t)}{dt} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \frac{1}{dt} \left\{ \underbrace{\left[nA(\xi)(\xi^i)^{n-1} + \frac{1}{2}n(n-1)B(\xi)^2(\xi^i)^{n-2} \right] dt}_I + \underbrace{\left[n(\xi^i)^{n-1}B^i(\xi) \right] dW^i}_{II} \right\}. \quad (\text{A.7})$$

On substitution of $A(\xi)$ and $B(\xi)$ into Equation A.7, term I then becomes:

$$\begin{aligned} n \left[-\rho \bar{u}_j \frac{\partial \xi^i}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial \xi^i}{\partial x_j} \right) - \frac{\rho C_d}{\tau_{\text{sgs}}} (\xi^i - \bar{\varphi}) + \rho \dot{\omega}^i(\xi) \right] (\xi^i)^{(n-1)} dt \\ + \left[n(n-1) \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \left(\frac{\partial \xi^i}{\partial x_j} \right)^2 \right] (\xi^i)^{(n-2)} dt. \end{aligned} \quad (\text{A.8})$$

In the stochastic contribution, the quantities B and dW are uncorrelated so that term II is zero:

$$\lim_{N_{\text{inf}} \rightarrow \infty} \frac{1}{N_{\text{inf}}} \sum_{i=1}^{N_{\text{inf}}} B^i(\xi) dW^i = 0.$$

Substituting Equation A.8 back into Equation A.7 results in:

$$\begin{aligned} \rho \frac{dM_n}{dt} = \lim_{N_{\text{inf}} \rightarrow \infty} \frac{1}{N_{\text{inf}}} \sum_{i=1}^{N_{\text{inf}}} \left[-n \rho \bar{u}_j \frac{\partial \xi^i}{\partial x_j} (\xi^i)^{n-1} + n \frac{\partial}{\partial x_j} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial \xi^i}{\partial x_j} \right) (\xi^i)^{n-1} \right. \\ \left. + n(n-1) \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \frac{\partial \xi^i}{\partial x_j} \frac{\partial \xi^i}{\partial x_j} (\xi^i)^{n-2} - n \frac{\rho C_d}{\tau_{\text{sgs}}} (\xi^i - \bar{\varphi}) (\xi^i)^{n-1} + n \rho \dot{\omega}^i(\xi^i)^{n-1} \right], \end{aligned}$$

and averaging over the stochastic fields gives:

$$\begin{aligned}
 \rho \frac{dM_n}{dt} = & \underbrace{-n\rho\bar{u}_j \overline{\frac{\partial\varphi}{\partial x_j} \varphi^{n-1}}}_{\text{Term 1}} + \underbrace{n \frac{\partial}{\partial x_j} \left(\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \right) \varphi^{n-1}}_{\text{Term 2}} \\
 & + \underbrace{n(n-1) \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \overline{\frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi}{\partial x_j} \varphi^{n-2}}}_{\text{Term 3}} - \underbrace{n \frac{\rho C_d}{\tau_{\text{sgs}}} (\bar{\varphi}^n - \overline{\varphi \varphi^{n-1}})}_{\text{Term 4}} \\
 & + \underbrace{n\rho\dot{\omega} \overline{\varphi^{n-1}}}_{\text{Term 5}}.
 \end{aligned}$$

The equation for the moments can be found from here through extensive use of the chain rule. Considering each term individually:

Term 1:

$$-n\rho\bar{u}_j \overline{\frac{\partial\varphi}{\partial x_j} \varphi^{n-1}} \rightarrow -\rho\bar{u}_j \overline{\frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi^n}{\partial\varphi}} = -\rho\bar{u}_j \overline{\frac{\partial\varphi^n}{\partial x_j}} = -\rho\bar{u}_j \frac{\partial M_n}{\partial x_j}$$

Term 2:

$$\begin{aligned}
 n \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \right] \varphi^{n-1} & \rightarrow \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \right] \frac{\partial\varphi^n}{\partial\varphi} = \\
 \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi^n}{\partial\varphi} \right] & - \left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \frac{\partial}{\partial x_j} \left(\frac{\partial\varphi^n}{\partial\varphi} \right) = \\
 \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi^n}{\partial x_j} \right] & - n \left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi^{n-1}}{\partial x_j} = \\
 \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial M_n}{\partial x_j} \right] & - n \left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi^{n-1}}{\partial\varphi} = \\
 \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial M_n}{\partial x_j} \right] & - n(n-1) \left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \overline{\frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi}{\partial x_j} \varphi^{n-2}}
 \end{aligned}$$

Term 3:

$$n(n-1) \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \overline{\frac{\partial\varphi}{\partial x_j} \frac{\partial\varphi}{\partial x_j} \varphi^{n-2}} \rightarrow n(n-1) \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \varphi^{n-2} \overline{\left(\frac{\partial\varphi}{\partial x_j} \right)^2}$$

A Comparison of the moments arising from the modelled probability density function equation and the stochastic fields

Term 4:

$$n \frac{\rho C_d}{\tau_{\text{sgs}}} (\overline{\varphi^n} - \overline{\bar{\varphi}} \overline{\varphi^{n-1}}) \rightarrow n \rho \frac{C_d}{\tau_{\text{sgs}}} (M_n - M_{n-1} \bar{\varphi})$$

Term 5:

$$n \rho \overline{\dot{\omega} \varphi^{n-1}} \rightarrow n \rho \overline{\dot{\omega} \varphi^{n-1}}$$

Collecting all of the terms, the moments arising from the stochastic fields are given by:

$$\begin{aligned} \rho \frac{\partial M_n}{\partial t} &+ \rho \bar{u}_i \frac{\partial M_n}{\partial x_i} + n(n-1) \frac{\mu}{\sigma} \overline{\varphi^{n-2} \left(\frac{\partial \varphi}{\partial x_i} \right)^2} \\ &= \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{\sigma} + \frac{\mu_{\text{sgs}}}{\sigma_{\text{sgs}}} \right) \frac{\partial M_n}{\partial x_j} \right] - n \rho \frac{C_d}{\tau_{\text{sgs}}} (M_n - M_{n-1} \bar{\varphi}) + n \rho \overline{\dot{\omega} \varphi^{n-1}} \end{aligned}$$

Appendix B

Example of satisfying conservation laws for a composition

The conservation procedure proposed is demonstrated via an example of methane combustion with the 7-species 4-step mechanism of Jones and Lindstedt [117]. This mechanism consists of the elements C, O, H and N, and contains the species CH₄, O₂, CO, CO₂, H₂, H₂O and N₂, with the indices given in Table B.1. It should be noted that no ANN predictions were actually carried out. Instead, an artificial initial set of values was constructed for the sake of the demonstration, so as to incur a significant conservation error that will be remedied by the proposed algorithm. In practice, a good ANN procedure would not incur so severe conservation errors in the first place. Furthermore, the fictitious composition has been chosen so that the procedure calculates a negative value in a residual species, so that the use of Algorithm 4 can also be demonstrated, although this did not occur in the 1D premixed flames due to the accuracy in the ANN predictions.

Table B.1: Elements and species contained in the mechanism and their respective indices.

Index	Element	Index	Species
1	C	1	CH ₄
2	H	2	O ₂
3	O	3	CO
4	N	4	CO ₂
		5	H ₂
		6	H ₂ O
		7	N ₂

B Example of satisfying conservation laws for a composition

In this mechanism, N_2 is inert and is the only nitrogen containing species, meaning it is not necessary to approximate it with ANNs or to enforce conservation of the N element. This means that $N_e = 3$. The candidate final residual species are those which only contain a single element, hence O_2 and H_2 , meaning that $N_c = 2$, $\mathcal{C}_c = (2, 5)$ and $\mathcal{E}_c^{\mathcal{C}} = (3, 2)$. The rectifying species for O_2 are CO_2 and CO , meaning that $\mathcal{R}_1 = 4$ and $\mathcal{R}_2 = 3$ and $\mathcal{E}^{\mathcal{R}} = 1$. In this mechanism, there are no pairs of rectifying species available for H_2 and, therefore, although it can be used as a final residual if calculated to be negative, the composition should be re-calculated with O_2 used as the final residual instead.

Step 1:

Column 2 of Table B.2 gives an example input composition. Following Algorithm 1, the required ANN predictions should be carried out although, as mentioned, fictitious predictions have been generated for the purpose of illustration. These fictitious predictions are shown in column 3 of Table B.2. Any unbounded predictions in $Y_i^*(t + \Delta t)$ are returned within bounds, giving column 4 of Table B.2.

Table B.2: Mass fractions of species at various stages in Algorithm 1.

Species	$Y_i(t)$	$Y_i^*(t + \Delta t)$	$Y_i^b(t + \Delta t)$
CH_4	0.011	0.008	0.008
O_2	0.031	0.081	0.081
CO	0.045	0.025	0.025
CO_2	0.141	0.143	0.143
H_2	0.007	0.006	0.006
H_2O	0.147	0.148	0.148
N_2	(0.618)	(0.618)	(0.618)

Step 2:

The next step is to use Algorithm 2 to select the residual species. The predicted element mass fractions are calculated and shown in Table B.3.

Table B.3: Target and predicted element mass fractions, calculated from species in Table B.2.

Quantity	$Y_i(t)$	$Y_i^b(t + \Delta t)$
$\sum_{i=1}^{N_{sp}} Y_i$	1.0000	1.0290
Z_C	0.0660	0.0557
Z_H	0.0262	0.0246
Z_O	0.2898	0.3307
Z_N	(0.6180)	(0.6180)

Then, the predicted contribution of each candidate final residual species to its respective element is calculated. For the first candidate final residual (O_2):

$$\frac{Y_{C_1}^b}{Z_{\mathcal{E}_1^C}^b} = \frac{Y_2^b}{Z_3^b} = \frac{0.0810}{0.3307} = 0.2449,$$

and for the second candidate final residual (H_2):

$$\frac{Y_{C_2}^b}{Z_{\mathcal{E}_2^C}^b} = \frac{Y_5^b}{Z_2^b} = \frac{0.0060}{0.0246} = 0.2442.$$

For this composition, O_2 is predicted to contribute the most to its respective element mass fraction, so $\mathcal{T} = 1$ and $\mathcal{S}_{N_e} = \mathcal{C}_1 = 2$. This means that the final element in the solution order is O, as $\mathcal{E}_{N_e} = \mathcal{E}_1^C = 3$. The rest of the element solution order is unimportant and is chosen as $\mathcal{E}_1 = 1$ and $\mathcal{E}_2 = 2$ to give the element solution order as $\mathcal{E}_j = (1, 2, 3)$ or (C, H, O). The remaining residual species are now determined. The possible species from which the first residual species is determined are:

$$\mathcal{Q}_q = \mathcal{S}_i, \text{ which contain } \mathcal{E}_1 \text{ and may contain } \mathcal{E}_2 \text{ and/or } \mathcal{E}_3,$$

which are $\mathcal{Q}_q = (1, 3, 4)$ or (CH_4 , CO , CO_2). The contributions of each of these species to element $\mathcal{E}_1$ are now calculated. For CH_4 :

$$\frac{A_{\mathcal{E}_1 \mathcal{Q}_1} Y_{\mathcal{Q}_1}^b}{Z_{\mathcal{E}_1}^b} = \frac{A_{11} Y_1^b}{Z_1^b} = \frac{0.7487 \times 0.0080}{0.0557} = 0.1075,$$

while for CO :

B Example of satisfying conservation laws for a composition

$$\frac{A_{\mathcal{E}_1 \mathcal{Q}_2} Y_{\mathcal{Q}_2}^b}{Z_{\mathcal{E}_1}^b} = \frac{A_{13} Y_3^b}{Z_1^b} = \frac{0.4288 \times 0.0250}{0.0557} = 0.1923,$$

and for CO₂:

$$\frac{A_{\mathcal{E}_1 \mathcal{Q}_3} Y_{\mathcal{Q}_3}^b}{Z_{\mathcal{E}_1}^b} = \frac{A_{14} Y_4^b}{Z_1^b} = \frac{0.2729 \times 0.1430}{0.0557} = 0.7002.$$

Therefore, $\mathcal{T} = 3$ and $\mathcal{S}_1 = \mathcal{Q}_{\mathcal{T}} = \mathcal{Q}_3 = 4$, meaning CO₂ is the first residual species. For the second residual species:

$$\mathcal{Q}_q = \mathcal{S}_i, \text{ which contain } \mathcal{E}_2 \text{ and may contain } \mathcal{E}_3,$$

so the possible residual species for the second element are $\mathcal{Q}_q = (5, 6)$, or (H₂, H₂O). The contributions of the possible residual species for $\mathcal{E}_2$ to the mass fraction of $\mathcal{E}_2$ are now calculated. For H₂:

$$\frac{A_{\mathcal{E}_2 \mathcal{Q}_1} Y_{\mathcal{Q}_1}^b}{Z_{\mathcal{E}_2}^b} = \frac{A_{25} Y_5^b}{Z_2^b} = \frac{1.0000 \times 0.0060}{0.0246} = 0.2442,$$

and for H₂O:

$$\frac{A_{\mathcal{E}_2 \mathcal{Q}_2} Y_{\mathcal{Q}_2}^b}{Z_{\mathcal{E}_2}^b} = \frac{A_{26} Y_6^b}{Z_2^b} = \frac{0.1119 \times 0.148}{0.0246} = 0.6740.$$

Therefore, $\mathcal{T} = 2$ and $\mathcal{S}_2 = \mathcal{Q}_{\mathcal{T}} = \mathcal{Q}_2 = 6$, meaning H₂O is the second residual species. As the final residual species has already been determined, all of the residual species are now determined and are: $\mathcal{S}_j = (4, 6, 2)$ or (CO₂, H₂O, O₂).

Step 3:

With the residual species determined, Algorithm 3 is now used to calculate the new values of the residual species which will make the overall composition satisfy element and mass conservation. The target element mass fractions are calculated and shown in Table B.3. The conserved mass fractions are initialised to the non-negative mass fractions from Step 1. The new values of residual species are calculate in turn using:

$$Y_{S_j}^{con} = \frac{Z_{\mathcal{E}_j} - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_j i} Y_i^{con}}{A_{\mathcal{E}_j S_j}}.$$

The mask $\mathcal{B}_i$ is set to one for all species. Then, to calculate the new mass fraction of the first residual species, $\mathcal{B}_{S_1}$ is set to zero, i.e. $\mathcal{B}_4 = 0$. For the first residual species:

$$Y_{S_1}^{con} = \frac{Z_{\mathcal{E}_1} - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_1 i} Y_i^{con}}{A_{\mathcal{E}_1 S_1}} = \frac{Z_1 - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{1 i} Y_i^{con}}{A_{1 4}} = 0.1807.$$

Next, $\mathcal{B}_{S_1}$ is set to one, and then $\mathcal{B}_{S_2}$ is set to zero: $\mathcal{B}_4 = 1$ and $\mathcal{B}_6 = 0$. The new value of the second residual species is calculated as follows:

$$Y_{S_2}^{con} = \frac{Z_{\mathcal{E}_2} - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_2 i} Y_i^{con}}{A_{\mathcal{E}_2 S_2}} = \frac{Z_2 - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{2 i} Y_i^{con}}{A_{2 6}} = 0.1627.$$

Finally, $\mathcal{B}_{S_2}$ is set to one, while $\mathcal{B}_{S_3}$ is set to zero: $\mathcal{B}_6 = 1$ and $\mathcal{B}_2 = 0$. The new value of the final residual is calculated as follows:

$$Y_{S_3}^{con} = \frac{Z_{\mathcal{E}_3} - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{\mathcal{E}_3 i} Y_i^{con}}{A_{\mathcal{E}_3 S_6}} = \frac{Z_3 - \sum_{i=1}^{N_{sp}} \mathcal{B}_i A_{3 i} Y_i^{con}}{A_{3 2}} = -0.00033.$$

Now all of the new residual species mass fractions have been determined and the composition satisfies the conservation laws. However, the mass fraction of O_2 has been predicted to be negative. This needs to be rectified.

Step 4:

The pair of species used to rectify this final residual are CO_2 and CO , i.e. $\mathcal{R}_1 = 4$ and $\mathcal{R}_2 = 3$, while the other element contained in these species is C , i.e. $\mathcal{E}^{\mathcal{R}} = 1$. Following Algorithm 4:

$$a = -A_{\mathcal{E}_{N_e} S_{N_e}} Y_{S_{N_e}}^{con} = -A_{3 2} Y_2^{con} = 0.00033$$

$$b = A_{O \ CO_2} = 0.7271$$

$$c = A_{O \ CO} = 0.5712$$

B Example of satisfying conservation laws for a composition

$$\begin{aligned}
 d &= \frac{A_{\text{C CO}_2}}{A_{\text{C CO}}} = 0.6365 \\
 \Delta Y_{\mathcal{R}_1}^{\text{con}} &= \frac{-a}{b - cd} = -0.00091 \\
 \Delta Y_{\mathcal{R}_2}^{\text{con}} &= -d \Delta Y_{\mathcal{R}_1}^{\text{con}} = 0.00058 \\
 Y_{\mathcal{R}_1}^{\text{con}} &= Y_{\mathcal{R}_1}^{\text{con}} + \Delta Y_{\mathcal{R}_1}^{\text{con}} = 0.1797 \\
 Y_{\mathcal{R}_2}^{\text{con}} &= Y_{\mathcal{R}_2}^{\text{con}} + \Delta Y_{\mathcal{R}_2}^{\text{con}} = 0.0256 \\
 Y_{\mathcal{S}_{N_e}}^{\text{con}} &= 0
 \end{aligned}$$

The final calculated composition Y_i^{con} now satisfies all of the conservation laws with no negative predictions. The composition is summarised in Table B.4, along with the initial composition.

Table B.4: Initial and final compositions after using Algorithms 1-4.

Species	$Y_i(t)$	$Y_i^{\text{con}}(t + \Delta t)$
CH ₄	0.011	0.0080
O ₂	0.031	0.0000
CO	0.045	0.0256
CO ₂	0.141	0.1797
H ₂	0.007	0.0060
H ₂ O	0.147	0.1627
N ₂	(0.618)	(0.618)
$\sum_{i=1}^{N_{sp}} Y_i$	1.0000	1.0000
Z_{C}	0.0660	0.0660
Z_{H}	0.0262	0.0262
Z_{O}	0.2898	0.2898
Z_{N}	(0.6180)	(0.6180)

References

- [1] E. E. O'Brien. The probability density function (PDF) approach to reacting turbulent flows. In *Turbulent Reacting Flows*, pages 185–218. Springer, 1980.
- [2] S. B. Pope. PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science*, 11(2):119–192, 1985.
- [3] C Dopazo. Recent developments in PDF methods. In *Turbulent Reacting Flows*, pages 375–474. Academic Press London, 1994.
- [4] L. Valiño. Field Monte Carlo formulation for calculating the probability density function of a single scalar in a turbulent flow. *Flow, Turbulence and Combustion*, 60(2):157–172, 1998.
- [5] V. Sabel'nikov and O. Soulard. Rapidly decorrelating velocity-field model as a tool for solving one-point Fokker-Planck equations for probability density functions of turbulent reactive scalars. *Physical Review E - Statistical, Non-linear, and Soft Matter Physics*, 72(1), 2005.
- [6] W. P. Jones and S. Navarro-Martinez. Large eddy simulation of autoignition with a subgrid probability density function method. *Combustion and Flame*, 150(3):170–187, 2007.
- [7] R. Mustata, L. Valiño, C. Jiménez, W.P. Jones, and S. Bondi. A probability density function Eulerian Monte Carlo field method for large eddy simulations: Application to a turbulent piloted methane/air diffusion flame (Sandia D). *Combustion and Flame*, 145:88–104, 2006.
- [8] W.P. Jones and V.N. Prasad. Large eddy simulation of the Sandia flame series (D-F) using the Eulerian stochastic field method. *Combustion and Flame*, 157(9):1621–1636, 2010.
- [9] V. N. Prasad, M. Juddoo, A. R. Masri, W. P. Jones, and K Luo. Investigation of extinction and re-ignition in piloted turbulent non-premixed methane–air flames using LES and high-speed OH-LIF. *Combustion Theory and Modelling*, 17(3):483–503, 2013.

- [10] D. Fredrich, W. P. Jones, and A. J. Marquis. The stochastic fields method applied to a partially premixed swirl flame with wall heat transfer. *Combustion and Flame*, 205:446–456, 2019.
- [11] R.W. Bilger. Conditional moment closure for turbulent reacting flow. *Physics of Fluids*, 5:327–334, 1993.
- [12] A.Y. Klimenko and R.W. Bilger. Conditional moment closure for turbulent combustion. *Progress in Energy and Combustion Science*, 25:595–687, 1999.
- [13] A.Y. Klimenko and S.B. Pope. The modeling of turbulent reactive flows based on multiple mapping conditioning. *Physics of Fluids*, 15:1907–1925, 2003.
- [14] O. Colin, F. Ducros, D. Veynante, and T. Poinso. A thickened flame model for large eddy simulations of turbulent premixed combustion. *Physics of Fluids*, 12:1843–1863, 2000.
- [15] A.R. Kerstein. A linear-eddy model of turbulent scalar transport and mixing. *Combustion Science and Technology*, 60:391–421, 1988.
- [16] F. A. Williams. *Combustion Theory: The Fundamental Theory of Chemically Reacting Flow Systems*. The Benjamin/Cummings Publishing Company, Inc., 2nd edition, 1985.
- [17] S. H. Lam and D. A. Goussis. The CSP method for simplifying kinetics. *International Journal of Chemical Kinetics*, 26(4):461–486, 1994.
- [18] A. Massias, D. Diamantis, E. Mastorakos, and D. A. Goussis. An algorithm for the construction of global reduced mechanisms with CSP data. *Combustion and Flame*, 117(4):685–708, 1999.
- [19] P. Koniavitis, S. Rigopoulos, and W. P. Jones. A methodology for derivation of RCCE-reduced mechanisms via CSP. *Combustion and Flame*, 183:126–143, 2017.
- [20] P. Koniavitis, S. Rigopoulos, and W. P. Jones. Reduction of a detailed chemical mechanism for a kerosene surrogate via RCCE-CSP. *Combustion and Flame*, 194:85–106, 2018.
- [21] N. Peters. *Turbulent Combustion*. Cambridge University Press, 2000.
- [22] C. D. Pierce and P. Moin. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *Journal of Fluid Mechanics*, 504:73–97, 2004.

- [23] U. Maas and S.B. Pope. Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds. *Symposium (International) on Combustion*, 24(1):103–112, 1992.
- [24] U. Maas and S.B. Pope. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combustion and Flame*, 88(3-4):239–264, 1992.
- [25] O. Gicquel, N. Darabiha, and D. Thevenin. Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion. *Proceedings of the Combustion Institute*, 28(2):1901–1908, 2000.
- [26] D.A. Goussis and M. Valorani. An efficient iterative algorithm for the approximation of the fast and slow dynamics of stiff systems. *Journal of Computational Physics*, 214:316–346, 2006.
- [27] H. Niemann, D. Schmidt, and U. Maas. An efficient storage scheme for reduced chemical kinetics based on orthogonal polynomials. *Journal of Engineering Mathematics*, 31:131–142, 1997.
- [28] S.R. Tonse, Moriarty N.W., N.J. Brown, and M. Frenklach. PRISM: Piecewise reusable implementation of solution mapping. An economical strategy for chemical kinetics. *Israel Journal of Chemistry*, 39:97–106, 1999.
- [29] G. Ribert, O. Gicquel, N. Darabiha, and D. Veynante. Tabulation of complex chemistry based on self-similar behaviour of laminar premixed flames. *Combustion and Flame*, 146:649–664, 2006.
- [30] S.B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combustion Theory and Modelling*, 1(1):41–63, 1997.
- [31] M. Ihme, W. T. Chung, and A. A. Mishra. Combustion machine learning: Principles, progress and prospects. *Progress in Energy and Combustion Science*, 91:101010, 2022.
- [32] L. Zhou, Y. Song, W. Ji, and H. Wei. Machine learning for combustion. *Energy and AI*, 7:100128, 2022.
- [33] J. C. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinso. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.
- [34] Z.M. Nikolaou, C. Chrysostomou, L. Vervisch, and S. Cant. Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods. *Flow, Turbulence and Combustion*, 103:485–501, 2019.

- [35] F.C. Christo, A.R. Masri, E.M. Nebot, and S.B. Pope. An integrated PDF/neural network approach for simulating turbulent reacting systems. *Symposium (International) on Combustion*, 26(1):43–48, 1996.
- [36] J. An, G. He, K. Luo, F. Qin, and B. Liu. Artificial neural network based chemical mechanisms for computationally efficient modeling of hydrogen/carbon monoxide/kerosene combustion. *International Journal of Hydrogen Energy*, 2020.
- [37] J.A. Blasco, N. Fueyo, C. Dopazo, and J-Y. Chen. A self-organizing-map approach to chemistry representation in combustion applications. *Combustion Theory and Modelling*, 4:61–76, 2000.
- [38] O. Owoyele and P. Pal. Chemnode: A neural ordinary differential equations framework for efficient chemical kinetic solvers. *Energy and AI*, 7:100118, 2022.
- [39] J-Y. Chen, J.A. Blasco, N. Fueyo, and C. Dopazo. An economical strategy for storage of chemical kinetics: Fitting in situ adaptive tabulation with artificial neural networks. *Proceedings of the Combustion Institute*, 28(1):115–121, 2000.
- [40] L.F. Große and F. Joos. Approximation of chemical reaction rates in turbulent combustion simulation. In *Proceedings of the 18th European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning*, 2010.
- [41] L.F. Große and F. Joos. Neural networks in combustion simulations. In *Proceedings of the International Conference on Fuzzy Computation and International Conference on Neural Computation*, 2010.
- [42] F.C. Christo, A.R. Masri, E.M. Nebot, and T. Turanyi. Utilizing artificial neural network and repro-modelling in turbulent combustion. In *IEEE International Conference on Neural Networks - Conference Proceedings*, 1995.
- [43] F.C. Christo, A.R. Masri, and E.M. Nebot. Artificial neural network implementation of chemistry with PDF simulation of H_2/CO_2 flames. *Combustion and Flame*, 106:406–427, 1996.
- [44] 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. *Combustion and Flame*, 113:38–52, 1998.
- [45] J.A. Blasco, N. Fueyo, J.C. Larroya, C. Dopazo, and J-Y. Chen. A single-step time-integrator of a methane-air chemical system using artificial neural networks. *Computers and Chemical Engineering*, 23(9):1127–1133, 1999.

- [46] N.S. Mehdizadeh, P. Sinaei, and A. Nichkoohi. Modeling Jones’ reduced chemical mechanism of methane combustion with artificial neural network. In *ASME 3rd Joint US-European Fluids Engineering Summer Meeting*, 2010.
- [47] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.
- [48] J. An, F. Qin, J. Zhang, and Z. Ren. Explore artificial neural networks for solving complex hydrocarbon chemistry in turbulent reactive flows. *Fundamental Research*, 2:595–603, 2022.
- [49] B.A. Sen and S. Menon. Turbulent premixed flame modeling using artificial neural networks based chemical kinetics. *Proceedings of the Combustion Institute*, 32:1605–1611, 2009.
- [50] B.A. Sen and S. Menon. Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames. *Combustion and Flame*, 157:62–74, 2010.
- [51] B.A. Sen, E.R. Hawkes, and S. Menon. Large eddy simulation of extinction and reignition with artificial neural networks based chemical kinetics. *Combustion and Flame*, 157:566–578, 2010.
- [52] A.K. Chatzopoulos and S. Rigopoulos. A chemistry tabulation approach via Rate-Controlled Constrained Equilibrium (RCCE) and Artificial Neural Networks (ANNs), with application to turbulent non-premixed $\text{CH}_4/\text{H}_2/\text{N}_2$ flames. *Proceedings of the Combustion Institute*, 34(1):1465–1473, 2013.
- [53] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.
- [54] W. Meier, R.S. Barlow, Y.L. Chen, and J.Y. Chen. Raman/Rayleigh/LIF measurements in a turbulent $\text{CH}_4/\text{H}_2/\text{N}_2$ jet diffusion flame: experimental techniques and turbulence–chemistry interaction. *Combustion and Flame*, 123:326–343, 2000.
- [55] L.L.C. Franke, A.K. Chatzopoulos, and S. Rigopoulos. Tabulation of combustion chemistry via Artificial Neural Networks (ANNs): Methodology and application to LES-PDF simulation of Sydney Flame L. *Combustion and Flame*, 185:245–261, 2017.
- [56] A.R. Masri, R.W. Bilger, and R.W. Dibble. Turbulent nonpremixed flames of methane near extinction: Mean structure from Raman measurements. *Combustion and Flame*, 71(3):245–266, 1988.

- [57] T. Readshaw, T. Ding, S. Rigopoulos, and W.P. Jones. Modeling of turbulent flames with the large eddy simulation–probability density function (LES–PDF) approach, stochastic fields, and artificial neural networks. *Physics of Fluids*, 33:035154, 2021.
- [58] T. Ding, T. Readshaw, S. Rigopoulos, and W. P. Jones. Machine learning tabulation of thermochemistry in turbulent combustion: An approach based on hybrid flamelet/random data and multiple multilayer perceptrons. *Combustion and Flame*, 231:111493, 2021.
- [59] R. S. Barlow, J. H. Frank, A.N. Karpetis, and J.Y. Chen. Piloted methane/air jet flames: Scalar structure and transport effects. *Combustion and Flame*, 143:433–449, 2005.
- [60] R. Laubscher and J.H. Hoffman. 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. *Journal of Thermal Engineering*, 4:1828–1846, 2018.
- [61] K. Wan, C. Barnaud, L. Vervisch, and P. Domingo. Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects. *Combustion and Flame*, 220:119–129, 2020.
- [62] P. Nguyen, H. Domingo, L. Vervisch, and D. Nguyen. Machine learning for integrating combustion chemistry in numerical simulations. *Energy and AI*, 5:100082, 2021.
- [63] G. Cerri, V. Michelassi, S. Monacchia, and S. Pica. Kinetic combustion neural modelling integrated into computational fluid dynamics. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 217:185–192, 2003.
- [64] Majid Haghshenas, Peetak Mitra, Niccolò Dal Santo, and D.P. Schmidt. Acceleration of chemical kinetics computation with the learned intelligent tabulation (LIT) method. *Energies*, 14:7851, 2021.
- [65] A. Kempf, F. Flemming, and J. Janicka. Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by LES. *Proceedings of the Combustion Institute*, 30:557–565, 2005.
- [66] M. Ihme and H. Pitsch. Modeling of radiation and nitric oxide formation in turbulent nonpremixed flames using a flamelet/progress variable formulation. *Physics of Fluids*, 20(5), 2008.

- [67] M. Ihme, C. Schmitt, and H. Pitsch. Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame. *Proceedings of the Combustion Institute*, 32:1527–1535, 2009.
- [68] M.D. Emami and A.E. Fard. Laminar flamelet modeling of a turbulent $\text{CH}_4/\text{H}_2/\text{N}_2$ jet diffusion flame using artificial neural networks. *Applied Mathematical Modelling*, 36:2082–2093, 2012.
- [69] O. Owoyele, P. Kundu, M. Ameen, T. Echekeki, and S. Som. Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames. *International Journal of Engine Research*, 2019.
- [70] M. Hansinger, Y. Ge, and M. Pfitzner. Deep residual networks for flamelet/progress variable tabulation with application to a piloted flame with inhomogeneous inlet. *Combustion Science and Technology*, pages 1–27, 2020.
- [71] R. S. Barlow, S. Meares, G. Magnotti, H. Cutcher, and A.R. Masri. Local extinction and near-field structure in piloted turbulent CH_4 /air jet flames with inhomogeneous inlets. *Combustion and Flame*, 162:3516–3540, 2015.
- [72] R. Ranade, G. Li, S. Li, and T. Echekeki. An efficient machine-learning approach for PDF tabulation in turbulent combustion closure. *Combustion Science and Technology*, 193:1258–1277, 2021.
- [73] R. Kapoor, A. Lentat, and S. Menon. Simulations of methane-air flames using ISAT and ANN. In *37th Joint Propulsion Conference and Exhibit*, 2001.
- [74] C. Cheng, G. Janiga, and D. Thévenin. On-the-fly artificial neural network for chemical kinetics in direct numerical simulations of premixed combustion. *Combustion and Flame*, 226:467–477, 2021.
- [75] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, 126(9):10797114, 2021.
- [76] T. Bolton and L. Zanna. Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, 11:376–399, 2019.
- [77] K. Wan, C. Barnaud, L. Vervisch, and P. Domingo. Machine learning for detailed chemistry reduction in DNS of a syngas turbulent oxy-flame with side-wall effects. *Proceedings of the Combustion Institute*, 38:2825–2833, 2021.
- [78] T. S. J. Brauner. *Large Eddy Simulation of Premixed and Stratified Turbulent Combustion*. PhD thesis, Imperial College London, 2016.

- [79] J. Smagorinsky. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3):99–164, 1963.
- [80] U. Piomelli and J. Liu. Large-eddy simulation of rotating channel flows using a localized dynamic model. *Physics of fluids*, 7(4):839–848, 1995.
- [81] F. Gao and E. E. O’Brien. A large-eddy simulation scheme for turbulent reacting flows. *Physics of Fluids A: Fluid Dynamics*, 5:1282–1284, 1993.
- [82] P. J. Colucci, F. A. Jaberi, P. Givi, and S. B. Pope. Filtered density function for large eddy simulation of turbulent reacting flows. *Physics of Fluids*, 10(2):499–515, 1998.
- [83] F. A. Jaberi, P. J. Colucci, S. James, P. Givi, and S. B. Pope. Filtered mass density function for large-eddy simulation of turbulent reacting flows. *Journal of Fluid Mechanics*, 401:85–121, 1999.
- [84] T. Brauner, W. P. Jones, and A. J. Marquis. LES of the Cambridge stratified swirl burner using a sub-grid PDF approach. *Flow, Turbulence and Combustion*, 96:965–985, 2016.
- [85] C. Dopazo and E. E. O’Brien. An approach to the autoignition of a turbulent mixture. *Acta Astronautica*, 1(9-10):1239–1266, 1974.
- [86] J. Villersmaux and J. C. Devillon. Représentation de la coalescence et de la redispersion des domaines de ségrégation dans un fluide par un modèle d’interaction phénoménologique. In *Proceedings of the 2nd International Symposium on Chemical Reaction Engineering*, volume 26, pages 1–13. New York:, Elsevier, 1972.
- [87] C. Gardiner. *Handbook of Stochastic Methods*. Springer-Verlag, 2009.
- [88] LES-BOFFIN: Users Guide, 2002. Imperial College London.
- [89] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with training edge separation. *AIAA Journal*, 21:1525–1532, 1983.
- [90] V.N. Prasad. *Large eddy simulation of partially premixed turbulent combustion*. PhD thesis, Imperial College London, 2011.
- [91] W.P. Jones, A.J. Marquis, and V.N. Prasad. LES of a turbulent premixed swirl burner using the Eulerian stochastic field method. *Combustion and Flame*, 159:3079–3095, 2012.
- [92] J.H. Ferziger and M. Peric. *Computation methods for fluid dynamics*. Springer Science & Business Media, 2012.

- [93] Y. P. Almeida. *Large Eddy Simulation of Supersonic Combustion using a Probability Density Function method*. PhD thesis, Imperial College London, 2019.
- [94] H. Wang, P. Zhang, and T. Pant. Consistency and convergence of Eulerian Monte Carlo field method for solving transported probability density function equation in turbulence modeling. *Physics of Fluids*, 30:115106, 2018.
- [95] S. Gallot-Lavallée. *Liquid Fuel Spray Atomisation, Ignition and Combustion Dynamics*. PhD thesis, Imperial College London, 2017.
- [96] M.T. Hagan, H.B. Demuth, M.H. Beale, and O. De Jess. *Neural network design*. Martin Hagan, 2014.
- [97] S. Ruder. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.
- [98] D. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11:431–441, 1963.
- [99] B. M. Wilamowski and H. Yu. Improved computation for Levenberg-Marquardt training. *IEEE Transactions on Neural Networks*, 21(6):930–937, 2010.
- [100] D.J.C. MacKay. Bayesian Interpolation. *Neural Computation*, 4:415–447, 1992.
- [101] F. Dan Foresee and M.T. Hagan. Gauss-Newton approximation to Bayesian learning. In *IEEE International Conference on Neural Networks - Conference Proceedings*, volume 3, pages 1930–1935, 1997.
- [102] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, Jr. W. C. Gardiner, V. V. Lissianski, and Z. Qin. GRI 3.0 mechanism. http://www.me.berkeley.edu/gri_mech/. [Online; accessed 21-July-2022].
- [103] H. Wang, M. Juddoo, S. H. Starner, A. R. Masri, and S. B. Pope. A novel transient turbulent jet flame for studying turbulent combustion. *Proceedings of the Combustion Institute*, 34(1):1251–1259, 2013.
- [104] L. di Mare, M. Klein, W. P. Jones, and J. Janicka. Synthetic turbulence inflow conditions for large eddy simulation. *Physics of Fluids*, 18(2):025107, 2006.
- [105] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: A Variable-Coefficient ODE solver. *SIAM Journal on Scientific and Statistical Computing*, 10:1038–1051, 1989.

- [106] T. Readshaw, L.L.C. Franke, W.P. Jones, and S. Rigopoulos. Simulation of turbulent premixed flames with machine learning - tabulated thermochemistry. *Under Review*, 2022.
- [107] T. Poinso and D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2nd edition, 2005.
- [108] M.S. Sweeney, S. Hochgreb, M.J. Dunn, and R.S. Barlow. The structure of turbulent stratified and premixed methane/air flames I: Non-swirling flows. *Combustion and Flame*, 159:2896–2911, 2012.
- [109] M.S. Sweeney, S. Hochgreb, M.J. Dunn, and R.S. Barlow. The structure of turbulent stratified and premixed methane/air flames II: Swirling flows. *Combustion and Flame*, 159:2912–2929, 2012.
- [110] R. Zhou, S. Balusamy, M.S. Sweeney, R.S. Barlow, and S. Hochgreb. Flow field measurements of a series of turbulent premixed and stratified methane/air flames. *Combustion and Flame*, 160:2017–2028, 2013.
- [111] S. Nambully, P. Domingo, V. Moureau, and L. Vervisch. A filtered-laminar-flame PDF sub-grid-scale closure for LES of premixed turbulent flames: II. Application to a stratified bluff-body burner. *Combustion and Flame*, 161:1775–1791, 2014.
- [112] F. Proch and A. M. Kempf. Numerical analysis of the Cambridge stratified flame series using artificial thickened flame LES with tabulated premixed flame chemistry. *Combustion and Flame*, 161:2627–2646, 2014.
- [113] H. Turkeri, X. Zhao, S. B. Pope, and M. Muradoglu. Large eddy simulation/probability density function simulations of the Cambridge turbulent stratified flame series. *Combustion and Flame*, 199:24–45, 2019.
- [114] T. Readshaw, S. Rigopoulos, and W.P. Jones. On the incorporation of conservation laws in artificial neural networks for chemical kinetics tabulation. *Under Review*, 2022.
- [115] M. Bode, M. Guading, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute*, 38:2617–2625, 2021.
- [116] Hai Wang, Xiaoqing You, Ameya V. Joshi, Scott G. Davis, Alexander Laskin, Fokion Egolopoulos, and Chung K. Law. USC Mech Version II. High-temperature combustion reaction model of H₂/CO/C1-C4 Compounds. http://ignis.usc.edu/USC_Mech_II.htm, 2007. Accessed: 27/09/2022.

- [117] W.P. Jones and R.P. Lindstedt. Global reaction schemes for hydrocarbon combustion. *Combustion and Flame*, 73:233–249, 1988.