

ACCELERATION TECHNIQUES FOR EFFICIENT AND ACCURATE PARTICLE PDF SIMULATIONS OF LARGE-SCALE TURBULENT COMBUSTION

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Pushan Sharma

August 2022

© 2022 Pushan Sharma
ALL RIGHTS RESERVED

ACCELERATION TECHNIQUES FOR EFFICIENT AND ACCURATE
PARTICLE PDF SIMULATIONS OF LARGE-SCALE TURBULENT
COMBUSTION

Pushan Sharma, Ph.D.

Cornell University 2022

In this time of severe climate change, there is an increasing need for sophisticated simulation tools to facilitate more efficient fossil-fuel based combustion devices with low pollutant and greenhouse gas emissions. In particular, to simulate a turbulent reacting flow, a proper prediction of the interactions between turbulence and chemistry is extremely important. Probability density function (PDF) methods have been shown to capture this strong turbulence chemistry interaction accurately. However, one of the biggest disadvantages of PDF methods is its significantly higher computational cost of solving the chemistry in its exact form compared to other simpler methods, such as flamelet-based models. This necessitates the development of strategies to reduce the cost of PDF type approaches without losing the level of accuracy. Two different categories of techniques to accelerate the process are explored in this thesis.

The use of analytical Jacobian is observed to be a promising step to accelerate the chemistry source term integration compared to using numerical Jacobian. Here, a generalized projection-based analytical Jacobian framework is provided that considers all species mass fractions and temperature in the state vector while satisfying the mass conservation constraint consistently. The use of a projection

matrix with the analytical Jacobian ensures that the mass fraction vector never goes off its realizable simplex, defined by the constraint, at any time step. This approach provides an accurate solution with different types of solvers, and also predicts the spectral properties of the corresponding dynamical system within round-off errors. Next, the algorithm for generating the analytical Jacobian is combined with a dimension reduction technique, the quasi-steady species (QSS) assumption. An automated algorithm package is developed, which provides the analytical expression of the Jacobian with QSS species, properly integrating all the additional derivatives corresponding to the QSS-derived algebraic expressions, and can be readily implemented in any high-fidelity turbulent simulation. Combining these two techniques (analytical Jacobian and QSS) has shown a significant reduction in the computational cost of a partially stirred reactor (PaSR) simulation.

Adaptive chemistry approaches, which tailor the fidelity and size of the kinetic models used for reaction integration to the local flame conditions, have reduced the cost of PDF-type methods due to their ability to accurately describe the relevant combustion kinetics with significantly fewer variables and equations. In this work, two new adaptive chemistry algorithms are developed. The first addresses a key challenge of efficiently capturing the secondary chemistry pathways, such as pollutant formation, in an existing pre-partitioned adaptive chemistry (PPAC) approach. This new algorithm, PPAC-Additive, decouples the secondary chemistry from primary chemistry pathways (fuel oxidation), thus developing smaller reduced kinetic models compared to PPAC. In PPAC-Additive, the reduced mod-

els are generated first for main oxidation targets only, and then the important secondary pathways, demonstrated with NO_x prediction, are dealt with afterwards in a separate stage. Both PPAC and PPAC-Additive generate the reduced models based on a sample set of compositions in an offline stage, which are then used adaptively during the simulation of interest. In contrast to the current adaptive frameworks, which require extensive pre-processing analysis and assumption, the second algorithm, dynamically partitioned adaptive chemistry (DPAC), is a completely stand-alone adaptive approach with limited need for user input. DPAC generates the reduced models during the adaptive simulation based on the encountered compositions at runtime. DPAC updates the reduced kinetic models continuously as the flame and its compositions evolve in time, making it more flexible and efficient than other existing adaptive chemistry approaches. Both PPAC-Additive and DPAC have shown significant gain in CPU cost and memory requirement compared to a detailed simulation and PPAC in the context of large-scale LES-PDF simulations.

BIOGRAPHICAL SKETCH

Pushan Sharma was born and grew up near Kolkata, India. He completed his Bachelor of Engineering (B.E.) degree in Mechanical Engineering from Jadavpur University, Kolkata, India. Then he pursued his Masters in Science (M.S.) degree from Indian Institute of Technology Delhi. During his Masters, Pushan worked under the guidance of Prof. Anjan Ray on flame structure and stability of low calorific gases burnt in an oxygen enhanced environment. In the spring of 2017, Pushan joined Cornell University, Ithaca, to pursue his PhD. In his first year of his PhD, he worked with Prof. C. Thomas Avedisian on film boiling of glycerol. In his second year, Pushan moved to Prof. Perrine Pepiot's research group, where his work focused on developing algorithms to reduce the cost of integrating the chemistry source term in particle PDF type simulations of turbulent combustion.

To those we have lost in Covid

ACKNOWLEDGEMENTS

I have a great pleasure in expressing my indebtedness to my supervisor, Prof. Perrine Pepiot for his guidance and encouragement extended to me, without which the present work would not have been possible. Her continuous feedback not only helped me to grow as a researcher, but also as a writer and a presenter. I am eternally grateful to her for helping me out during a very difficult year of my life, and providing me with the opportunity to conduct research in her lab. Perrine has consistently answered my questions, sometimes very stupid ones, compassionately and with unparalleled availability. Apart from research, it was also an amazing experience to work as a teaching assistant with her. She is a brilliant teacher, and I only hope I can be even half the teacher she has been to her students. Overall, it has been a great pleasure working with her for almost 5 years.

I would also like to thank the members in my research committee, Prof. Elizabeth Fisher and Prof. Peter Diamessis for providing valuable comments and suggestions pertaining to my thesis. Both of them are great teachers, and attending their classes provided a great learning experience. And, of course how can I forget those little chitchats with Peter everyday at Gimme! I would also like to acknowledge Prof. Olivier Desjardins for providing the NGA code that is extensively used in this work. All the numerous academic discussions with him during group meetings were inspiring, and so equally refreshing were the non-academic ones at the coffee hour or at liquid state. I have also been lucky to serve as a teaching assistant with Prof. Charles Williamson, whose teaching demos in the fluids class were simply jaw-dropping. Finally, I was privileged to work with one of

the stalwarts in the field of combustion or turbulent flow simulation in general: Prof. Stephen B. Pope. The four meetings I had with him regarding the analytical Jacobian framework and the concept of projection matrix were awe-inspiring and something that I will always reminisce in the future. I am incredibly grateful to Dr. Marcia J. Sawyer for supporting me during the worst point of my PhD, and helping me to take one of the most important career changing (possibly life changing) decision of working with Perrine.

I would be failing in my duties if I don't acknowledge Dr. Ashish Newale. Ashish helped me with writing the analytical Jacobian code, understanding the subtle details of the PDF code, and provided me several files (and insights!) from his previous Sandia Flame D runs. His obsession with verifying and validating every single detail of the smallest of changes in any code was annoying, but provided the "rigor to truly infer something with confidence", in his own words. I am also grateful to Dr. Himanshu Goyal for helping me understand the PaSR code at the very beginning of my PhD, and showing me how to debug a large code. He did not know how I wrote the analytical Jacobian including QSS species, but without him I would not be able to do it within a couple of months. Dr. Lara Backer-Heberle, you had to deal with the unfortunate fate of sitting next to me, and therefore bearing the burnt of my questions. Thank you for being very patient and answering all the questions. I always enjoyed the discussions regarding the building algorithm you developed, which I eventually used pretty much everywhere! And how can I forget the great Mr. Nick Kincaid! I shared the longest amount of time in the lab with Nick. You have been an invaluable resource, and an

even better friend. You gave me the idea of LTC mass fraction at a very crucial moment of my PhD, and magically my flame ignited the next day, and now finally I am defending my thesis. A huge thank you to Dr. Michael Kuhn for including me in the fantasy league, which elevated the experience of watching Premier League on a different level. Austin, Joseph, Lam and Fabien: it has been a great pleasure sharing the same office with you on both personal and professional level.

A major part of what made this PhD bearable was the social relationships I had with other graduate and post graduate students from different departments at Cornell. Anything I mention about Kritanjan, Soham, Anirudra, Souvik, Sutanuka, Sujata, Aleesha, Sadhan, Gourab, Udit, Ankush, Kushal, Anirban (two of them!), Sudipta Chatterjee, Arnab, Surajit, Arpita, Shirsendu, Shikha, Joy, Upasana, Arjun, Mousumi and little Amu would be a tremendous understatement. They were part and parcel of my life for 5 long years. From playing cricket to having a picnic, from losing passport, laptop and everything else on the other side of the country to driving together for 8 hours on a day just to have some delicious biriyani, from having those little group politics, gossips and keeping secrets to sharing the pure joy of India winning the test series against Australia in Australia, it has been the most beautiful 5 years of my life. I am grateful to you for sharing all the joys and sorrows of the graduate school with me. Fiona, Sudipta Biswas (Buddy) and Rupai, you three are the most special people in my PhD life. Fiona, you are a role model to me. I enjoyed your paintings and writings, every subtle layers you find in a perfume or in the beautiful sunset of Ithaca. Listening to Vilayat Khan in your LP or appreciating Seinfeld are only a few of the blissful

experiences I had with you. You convinced me to watch some of the best movies in the world, as opposed to Buddy and Rupai, who I watched the worst movies of all time with. I will never forget our late night walks at the Slope, and of course the Barcelona football matches we watched together. Speaking of Football, my life would take a different shape altogether if it was not for Pep and Messi, the best manager and the best player of all time in my opinion. They have motivated me everyday, provided me the perfect medicine during extreme frustration. Lionel Messi, you are truly like the friend next door, who performs like God.

At last but not the least, I would extend my wholehearted sense of gratitude to my family, for all the love they showered on me; with their desire to see me in a good position has been the prime inspiration for me. In this limited space, it is simply impossible to give the names of large number of friends and colleagues whose direct or indirect help has assisted me to work on this project. I am personally obliged and highly grateful to all those who have guided me.

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	ix
List of Tables	xii
List of Figures	xiv
1 Introduction	1
1.1 Motivation and Background	1
1.2 High-fidelity turbulent combustion simulation	5
1.3 Traditional approaches for accelerating chemistry source term integration	9
1.4 Analytical Jacobian	12
1.5 Adaptive Chemistry	16
1.6 Contribution of this work	21
1.7 Outline	25
2 A unifying analytical Jacobian framework for chemical kinetics dynamical systems	27
2.1 Abstract	27
2.2 Introduction	28
2.3 Definitions and notations	33
2.4 Impact of a state vector on the derivative of a scalar field	34
2.4.1 Case (a): If $d\mathbf{x}$ is in the tangent subspace, τ	37
2.4.2 Case (b): If $d\mathbf{x}$ is not on the tangent subspace, τ	38
2.5 Projection matrices, \mathbb{P}	39
2.5.1 Projection on the N_s^{th} coordinate	39
2.5.2 Orthogonal Projection	41
2.5.3 Scaling	42
2.6 Governing equations of chemically dynamic systems	43
2.7 Implementation of Projection Matrix	44
2.8 Results and Discussion	47
2.8.1 0D auto-ignition	48
2.8.2 0D auto-ignition with mixing	53
2.8.3 Partially Stirred Reactor (PaSR)	56
2.8.4 Eigen-decomposition of Jacobian matrix	59
2.8.5 Linear implicit solver: Rosenbrock method	62

2.9	Conclusions	64
2.10	Appendix	66
2.10.1	Comparison of error propagation between idealized ignition and idealized ignition-mixing case	66
2.10.2	Auto-ignition with quadruple precision	68
2.10.3	Difference in DVODE and RODAS	70
3	An Analytical Jacobian Generator for Reduced Chemical Kinetic Models Involving Quasi-Steady-State Assumptions	74
3.1	Abstract	74
3.2	Introduction	75
3.3	Theory	79
3.3.1	Thermodynamic Relations	79
3.3.2	Description of chemical kinetics	81
3.4	Analytical Jacobian Formulation	85
3.4.1	Derivatives of mass fraction rate change w.r.t mass fractions	87
3.4.2	Derivatives of temperature rate change w.r.t mass fractions	90
3.4.3	Derivatives of mass fraction rate change w.r.t temperature	91
3.4.4	Derivatives of temperature rate change w.r.t temperature	95
3.5	Quasi-Steady-State (QSS) species implementation	95
3.6	Results and Discussion	99
3.7	Conclusions	107
4	Efficient treatment of secondary kinetic processes for pre-partitioned adaptive chemistry approaches	108
4.1	Abstract	108
4.2	Introduction	109
4.3	Numerical framework	112
4.3.1	Partially stirred reactor (PaSR)	112
4.3.2	LES/PDF flow solver	113
4.4	Methodology	114
4.4.1	Pre-partitioned adaptive chemistry (PPAC)	114
4.4.2	Building algorithm	116
4.4.3	Additive treatment of secondary pathways in PPAC	118
4.5	Results and Discussion	121
4.5.1	PaSR configuration	121
4.5.2	LES/PDF of Sandia Flame D	127
4.6	Conclusions	132

5	A dynamically partitioned adaptive chemistry methodology for efficient implementation of combustion chemistry	134
5.1	Introduction	135
5.2	Numerical Framework	138
5.2.1	Partially stirred reactor (PaSR)	139
5.2.2	LES/PDF flow solver	140
5.3	Overview of DPAC	141
5.3.1	Pre-processing stage	141
5.3.2	Adaptive simulation	143
5.4	Building algorithm	146
5.5	Partitioning algorithm	149
5.6	Results and discussion	152
5.6.1	PaSR configuration	153
5.6.2	LES/PDF configuration	165
5.7	Conclusions	178
6	Conclusions	180
6.1	Summary	180
6.2	Future work	184

LIST OF TABLES

2.1	Comparison of integration statistics, conservation properties and error norms between \mathbb{R}^ϕ and different choice of projections for two 0D constant pressure auto-ignition cases: heptane-air and heptane-oxygen. NST, NFE and NJE refer to number of steps, number of source term evaluations and number of Jacobian evaluations during the full auto-ignition simulation. For \mathbb{R}^ϕ , the N_s^{th} species is considered to be N_2 . The time stepping for these two auto-ignition cases are considered to be 10^{-6} and 10^{-7} s respectively. Two cases were run for 5,000 and 10,000 time steps respectively. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15}	49
2.2	Comparison of integration statistics, conservation properties and error norms between \mathbb{R}^ϕ and different choice of projections for two 0D constant pressure auto-ignition with mixing cases: heptane-air and heptane-oxygen. NST, NFE and NJE refer to number of steps, number of source term evaluations and number of Jacobian evaluations during the full simulation. For \mathbb{R}^ϕ , the N_s^{th} species is considered to be N_2 . The time stepping for these two cases are considered to be 10^{-7} and 10^{-8} s. These two cases were run for 7,000 and 6,000 time steps respectively. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15}	54
2.3	Specification of the parameters for PaSR case. Species are given as mass fractions and the initial condition is same as Stream 1.	57
2.4	Comparison of error norms between different choice of projections and \mathbb{R}^ϕ in PaSR. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15}	58
2.5	Comparison of relative errors of the maximum positive eigenvalue and crossover points between different choice of projections and \mathbb{R}^ϕ in a 0D auto-ignition simulation of heptane-oxygen mixture.	60
2.6	Comparison of relative errors of the maximum positive eigenvalue and crossover points between different choice of projections and \mathbb{R}^ϕ in a 0D auto-ignition with mixing simulation of heptane-oxygen mixture.	61

2.7	Comparison of integration statistics and error norms between different choice of projections and \mathbb{R}^ϕ in a 0D auto-ignition simulation of heptane-oxygen mixture using RODAS. NST and NJE refer to number of steps and number of Jacobian evaluations, respectively. NST and NJE are exactly same as RODAS evaluates the exact Jacobian at each sub-step.	63
2.8	Comparison of error norms between different choice of projections and \mathbb{R}^ϕ in a constant pressure 0D auto-ignition case of heptane-oxygen mixture with two different tolerances and with both double and quadruple precision.	69
3.1	Specification of the parameters for PaSR case. Species are given as mass fractions and the initial condition is same as Stream 1.	101
5.1	Inlet species mass fractions.	166
5.2	Normalized RMSD (%) of resolved mean quantities between LES/PDF using detailed mechanism and DPAC or PPAC. The relative number of species required to achieve those RMSD errors are provided in the last row. All the quantities are evaluated over five flow-through times in their statistically steady period.	170

LIST OF FIGURES

1.1	Energy consumption from different kind of fuels. The energy consumption is in quadrillion British thermal units [1].	2
2.1	Illustration of the realizable simplex, \mathcal{R} , and unity vector, \mathbf{u} for $N_s = 2$. The difference between two realizable vectors, $d\mathbf{y} = \mathbf{y}_2 - \mathbf{y}_1$, lies in τ , the tangent subspace of \mathcal{R} and orthogonal complement of \mathbf{u}	34
2.2	A 2-D interpretation of different projections: (a) parallel to N_s^{th} coordinate (\mathbb{P}^n), (b) orthogonal projection (\mathbb{P}^\perp), (c) scaling (\mathbb{P}^S). In each figure, \mathbf{x} corresponds to the state vector off the realizable simplex, \mathcal{R} and three \mathbf{y} vectors with different superscripts correspond to the state vector after different projections.	39
2.3	The norm of errors: between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for a 0D auto-ignition case of heptane-oxygen mixture (solid blue line), and between the actual and perturbed \mathbb{R}^ϕ (dashed blue line). The plot also depicts the evolution of temperature during ignition (dash and dot green line). Both absolute and relative tolerances for ODE integration are considered to be 10^{-15}	51
2.4	The norm of errors between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for an ignition-mixing model of heptane-oxygen mixture. The plot also depicts the evolution of temperature during ignition. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15}	55
2.5	The norm of errors between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for the PaSR model. The plot also depicts the evolution of temperature during the simulation. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15}	58
2.6	The relative error of λ_{max} between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for an auto-ignition case of heptane-oxygen mixture. The plot also depicts the evolution of temperature during ignition. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15}	61
2.7	The norm of errors between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for a 0D auto-ignition case of heptane-oxygen mixture with quadruple precision and two different ODE tolerances (both absolute and relative): 10^{-15} and 10^{-20} . The plot also depicts the evolution of temperature during ignition.	69

2.8	The order of accuracy of the solution at the end of each user specified time step in DVODE for an auto-ignition case of heptane-oxygen mixture.	73
3.1	The Jacobian matrix structure	87
3.2	Evolution of mean temperature, mass fractions of OH, CO ₂ and n-dodecane, and two QSS species: C ₂ H ₃ and A-C ₃ H ₅ in the PaSR using AJ, NJ and AAJ.	103
3.3	CPU time, number of Jacobian evaluations (NJE) and number of source term evaluation (NFE) as a function of number of QSS in the system for three different Jacobian formulations.	104
3.4	Variation of CPU time and number of source term evaluations with varying tolerances of the solver.	106
4.1	Relative number of species as a function of incurred error in temperature (circles) and NO mass fraction (triangles) for PPAC (black solid line) and PPAC-Additive (red dashed line) compared to the non-adaptive case. Points marked as '1' and '2' are used for further analysis in the text.	122
4.2	Relative number of oxidation ("OX" in legend) and NO _x species as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.	124
4.3	Region-specific difference in the number of oxidation species retained by the PPAC and PPAC-Additive reduction procedures, corresponding to an equivalent incurred error in the adaptive simulation. The circles indicate the probability of a particle to be in region J, P_J	125
4.4	Average clock time per time step relative to the detailed simulation as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.	126
4.5	Relative number of species as a function of incurred error in temperature (circles) and NO mass fraction (triangles) for PPAC (black solid line) and PPAC-Additive (red dashed line) compared to the non-adaptive case.	128
4.6	Relative number of oxidation ('OX' in legend) and NO _x species as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.	129

4.7	Instantaneous particle distribution colored by temperature (left), NO mass fraction (middle) and index of the model used for reaction source term integration (right).	130
4.8	Relative number of oxidation and NO _x species in each model with respect to the detailed mechanism for (a) PPAC, and (b) PPAC-Additive. The color pattern matches that of the right panel of Fig. 4.7.	131
4.9	Comparison of radial profile of the experimental (Exp) and the computed (Detailed, PPAC and PPAC-Additive) resolved mean and RMS of temperature and NO mass fraction at two different axial locations: 15D and 30D.	133
5.1	Flowchart depicting actions and decision criteria of the adaptive simulation of DPAC at time <i>t</i>	144
5.2	Relative number of species as a function of the quality of partition for three different incurred error in temperature.	155
5.3	Relative number of species as a function of incurred error in temperature for the initial database \mathcal{D}_1^0 (detailed PaSR) with re-partitioning (black squares) and without re-partitioning (red triangles).	158
5.4	Relative number of species as a function of incurred error in temperature for the initial database (a) \mathcal{D}_2^0 (0D runs with high and low temperature chemistry) and (b) \mathcal{D}_3^0 (0D runs with only low temperature chemistry) with re-partitioning (black squares) and without re-partitioning (red triangles). The transparent lines correspond to the same two curves from Fig. 5.3	159
5.5	Relative number of species as a function of incurred error in temperature during DPAC runs with all three initial databases. Clearly, the accuracy of DPAC is independent of the initial database.	161
5.6	Relative number of species as a function of incurred error in temperature for DPAC (black squares) and PPAC (red triangles), both with initial database, \mathcal{D}_1^0 (detailed PaSR).	162
5.7	Relative number of species as a function of incurred error in temperature for DPAC (black squares) and PPAC (red triangles), both with initial database, (a) \mathcal{D}_2^0 (0D runs with high and low temperature chemistry) and (b) \mathcal{D}_3^0 (0D runs with only low temperature chemistry).	163

5.8	(a) The evolution of mean temperature in a PaSR with detailed mechanism (black solid line), DPAC with $\epsilon_{th} = 10^{-2}$ (red dashed line) and $\epsilon_{th} = 10^{-4}$ (blue dash dot line), (b) The absolute difference in mean temperature between detailed run and two DPAC runs, (c) Relative number of species used at each time step in case of both DPAC runs.	164
5.9	Comparison of resolved mean of temperature and species mass fractions of CO ₂ , OH and LTC between LES/PDF with detailed mechanism and adaptive runs corresponding to DPAC and PPAC with three different databases.	169
5.10	Relative computational cost of one time-step of individual components of DPAC: chemistry source term integration, building models, partition quality check and re-partitioning with respect to the cost of chemistry source term integration with detailed mechanism.	171
5.11	Relative total computational cost of each of the components of DPAC and overall DPAC, and PPAC with three different databases with respect to the total cost of detailed simulation during the building mode.	172
5.12	Relative total computational cost of each of the components of DPAC and overall DPAC, and PPAC with three different databases with respect to the total cost of detailed simulation during the surveillance mode.	173
5.13	Instantaneous particle distribution colored by temperature and index of the model used at four different time instants of the building mode. (a) Towards the beginning, only two models are being used for jet and co-flow. (b) As the jet and co-flow compositions mix, the accessed region is re-partitioned into 3 regions, the third one being allocated for the mixed compositions, which captures the LTC. (c) A few ignition kernels start to appear. There is enough variance in the composition space for DPAC to use ten models, (d) The partition keeps using all models as the flame evolves, and finally finds the optimal partition.	175
5.14	Instantaneous particle distribution colored by temperature, relative number of species being used by each particle, index of the model used for reaction source term integration and mass fraction of LTC from left to right, respectively, during surveillance mode.	176

CHAPTER 1

INTRODUCTION

1.1 Motivation and Background

In 2022, even in the midst of a pandemic and a war, climate change is still arguably the largest threat to human well-being and health of this planet in the long run. Today, the atmospheric concentration of carbon dioxide (CO₂), which is the most important green house gas, is around 418 ppm. In the past 800,000 years this level never crossed 300 ppm, which means that we have observed almost a 40% increase in the CO₂ level in the last few decades. As a result, the global temperature has increased by 1.1⁰ C since the late 19th century, the arctic sea ice extent is reduced by 13% since 1979, Earth's polar ice sheets are melting by 427 billion metric tons every year since 2002, the amount of heat added to the ocean since 1955 is 337 zettajoules, and the sea level has risen by 4 inches since January 1993 [2]. The largest source (over 65% as in 2021) of the greenhouse gases has been the combustion of fossil fuels in transportation, industry and residential uses. The other class of pollutants commonly created at high temperature combustion of fossil fuels is nitrogen oxides (NO_x), such as NO and NO₂. These NO_x pollutants result in photo-chemical smog, acid rain and so on. Human exposure to NO_x is linked to severe diseases, as well as premature deaths [3].

Figure 1.1 shows the consumption of energy in the last three decades from

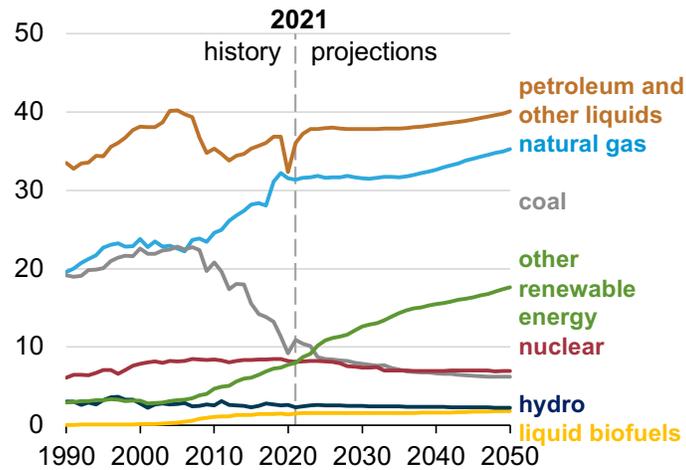


Figure 1.1: Energy consumption from different kind of fuels. The energy consumption is in quadrillion British thermal units [1].

different sources, and the forecast as well for the next three decades. Clearly, a significant shift towards renewable energy is observed in the last decade such as solar, wind, hydroelectric, the use of bio-fuels for a better future. This would definitely contribute in reducing the impact of global climate change in coming decades. However, it is projected that fossil fuels will still constitute more the 65% of the energy consumption in 2050 due to several reasons [1]. First of all, the high energy density of the fossil fuels, which is even one order of magnitude higher than the state-of-the-art batteries [4]. Secondly, although the light transportation will be electric very soon, there currently exists no alternative to combustion of fossil fuels in aviation industry. Third, the technologies relevant to all renewable energy sources are still mostly in the developing stage, and a lot of different issues need to be resolved before they can replace fossil fuels. For instance, Jones [5] pointed out the questions that still need to be answered about electric cars. Thus,

while renewable energy technologies continue to develop, it remains important to improve the combustion devices as well.

Developing more efficient and cleaner design of combustion devices is the most important avenue of either improving fuel economy and thermal efficiency, or reducing emission of pollutants, such as carbon monoxide, soot or NO_x . For instance, the former can be achieved by variable intake and exhaust valve actuation in internal combustion engines, whereas HCCI engines running in fuel-lean conditions decrease the soot productions significantly compared to typical diesel engines. The other dimension of improving combustion devices involve the use of low-to-zero carbon fuels. Hydrogen combustion has been a major topic of research in the last few years. On the other hand, carbon capture technologies store the released carbon dioxide underground. The use of bio-fuels as an alternative renewable fuel is another important step in the right direction.

The design of advanced combustion devices also requires advancement in novel computational tools. Simulations can provide quick design iterations, and novel concepts along with a comprehensive set of data, therefore allowing scope of further optimization. However, there exist certain challenges in capturing a multi-physics combustion system in a computational simulation. Combustion chemistry is described by multi-component fuels, where each fuel contains hundreds to thousands of chemical species in their chemical mechanism [6]. Apart from solving governing equations for each of those species, an additional challenge appears from the presence of wide range of time scales of the species and re-

actions. The complexity increases even further while capturing secondary chemistry, such as soot or NO_x, which has a different order of time scales compared to fuel oxidation. A simulation of fuel-lean combustion in an engine required to reduce such pollutants, as mentioned previously, requires the addition of low temperature chemistry (LTC), the pathways of which are completely different from the high temperature combustion pathways. Chemistry aside, the flow inside a combustor is typically turbulent, which itself contains an extremely wide range of length and time scales. Although the governing equations of each of these processes are known, resolving all the length and time scales in a turbulent combustion simulation is still not an affordable task with the current available computational power. Thus, the idea is to model the governing equations to describe each of the processes accurately while remaining affordable. Large eddy simulations (LES) is a widely used approach to simulate turbulent flows, where the large scales are resolved exactly and the small scales are modeled. In case of reacting flow simulation, capturing the turbulence chemistry interaction both accurately and efficiently is still a major topic of academic research. Transported probability density function (PDF), especially the Lagrangian particle PDF [7] method resolves combustion chemistry accurately without any *a-priori* assumption. However such particle PDF methods are usually computationally expensive requiring the solution of each chemical species.

The basis of this dissertation is to develop two types of algorithms to reduce the cost of capturing the chemistry in LES-PDF type simulations. First, an algorithm is developed to provide an analytical Jacobian in the presence of quasi-

steady state species that speeds up the chemistry solver, and reduces the overall stiffness of the system. Secondly, the concept of adaptive chemistry is explored that reduces the number of species in the system in a very efficient way, thus reducing the number of equations that needs to solved. The first part of the adaptive chemistry extends a previously developed Pre-partitioned adaptive chemistry (PPAC) algorithm in the context of capturing NO_x chemistry. In the second part, a novel and dynamically adaptive algorithm is developed and validated in the context of capturing low-temperature chemistry (LTC).

1.2 High-fidelity turbulent combustion simulation

In an incompressible turbulent non-reacting flow, we need to worry about several factors, such as geometry, resolving large scales of turbulent motions, the scale separation of turbulence depending on the Reynolds number and so on. Turbulent reacting flow simulations add an extra concern about the chemistry source term, for which detailed chemical kinetics is important, especially when it involves limiting phenomena such as ignition or extinction, and pollutant emissions. The set of governing ODEs that we typically solve for all species in a detailed mechanism are of the following form:

$$\frac{D\Phi}{Dt} = \frac{1}{\rho} \nabla \cdot (\rho D \nabla \Phi) + S(\Phi), \quad (1.1)$$

where $\frac{D}{Dt}$ is the material derivative, Φ is the composition vector, typically consisting of all species mass fractions (or concentrations) and temperature, the first

term in the right hand side is the molecular diffusion term, and finally S is chemical source term or the net production rate of each species. During modeling of this set of equations, one of the most demanding task is to model the chemical source term $S(\Phi)$ involving turbulent-chemistry interaction (TCI) in either time-averaged RANS type simulation ($\overline{S(\Phi)}$) or spatially-filtered LES ($\widetilde{S(\Phi)}$). The challenge arises from the fact that neither $\overline{S(\Phi)}$ nor $\widetilde{S(\Phi)}$ can be modelled as $S(\overline{\Phi})$ or $S(\widetilde{\Phi})$ due to the strong non-linearity present in S . The time-averaged or spatially filtered source term or the modeling of TCI has been the focus of the numerous previous studies. Most of these modeling approaches can be divided broadly into two categories: flamelet-like and PDF-like [8].

Flamelet models assume a low-dimensional manifold, typically one, two or three dimensional, in the high dimensional composition space, such that the solution is governed by a few variables, mixture fraction, scalar dissipation rate or different progress variables. In case of premixed flames, the flamelet models [9–14] use 1D laminar strained or freely propagating flames, and a user-defined progress variable. For instance, in flame surface density (FSD) models [9] or G-equation model [12], laminar flame speed is used as a model input, whereas in the flamelet generated manifold (FGM) method [13], the full 1D solution is tabulated. On the other hand, in case of non-premixed flamelet models, such as in steady laminar flamelet (SFM) [15] or in the flamelet progress variable model (FPV) [16, 17], 1D counterflow flame solutions are used. The other categories of flamelet models are flame prolongation of ILDM [14] and reaction diffusion manifolds (REDIM) [18, 19]. The unsteady flamelet models [20–22] have all the

features of flamelet-like models, except they need to add extra dimensions in the low-dimensional manifolds, such as unsteady FPV [23] and the tabulated flamelet model [24]. The advantage of using flamelet type models is the low computational cost as TCI is modeled in a low dimensional manifold, and the finite rate chemistry evaluations are performed typically in 1D cases. Recently Kundu *et al.* [25] implemented the unsteady tabulated flamelet model in an engine simulation using a detailed mechanism with 2755 species in it. However, the assumption of low-dimensional manifold fails when there exists strong TCI so that turbulence penetrates into the flame structure, thus flamelet modes are difficult in problems involving multiple turbulent combustion regimes.

On the other hand, the PDF-like models do not make any assumptions on the flame topology, i.e., these methods evaluate the chemical source term in its exact form. The TCI closure is resolved based on some statistical representation of the thermochemical states using a large number of notional particles. The fundamental idea of the PDF type approaches is to solve a modeled conservation equation for the joint PDF of different fluid properties, e.g., species mass fractions and enthalpy. Although the reaction term is solved exactly, the reactive mixing is treated by a mixing model to account for the effects of molecular diffusion. A comprehensive review is provided by Haworth [26] for all the PDF type models available, including the treatment of the mixing model. The most common mixing models are interaction by exchange with the mean (IEM) [27, 28], the modified Curl's model (MC) [29], and Euclidean minimum spanning tree (EMST) [30]. Kuron *et al.* [31] found that the EMST model performs better than IEM and MC, as EMST,

unlike the latter two, accounts for the effects of chemical reaction on molecular mixing. In this work, we have extensively used the IEM model. Several studies have implemented the PDF calculation efficiently in RANS and LES [7, 32–42]. Apart from modeling the reactive mixing, the biggest disadvantage of this particle PDF type methods is its computational cost. Hiremath *et al.* [38] reported that the transported particle PDF [7] is 8.6 and 17.3 times more expensive than the steady flamelet model while using a 16 and 38 species methane mechanism, respectively, along with an optimal load balancing strategy for ISAT. The memory requirement is also significantly larger for these type of approaches as each cell contains about 20-40 particles per cell, each of them carrying their own full composition vector along with their location in the physical space. As Pope [8] explained, for a 38 species mechanism and 40 particles per cell, the thermo-chemical information content is 760 times greater than just carrying the mean and variance of mixture fraction in a flamelet type model.

Apart from the modeling of TCI, a turbulent combustion simulation of a realistic device needs to capture the multi-component fuel chemistry. The detailed kinetic models of each of the fuel component may include thousands of species [6]. With a higher number of species, the stiffness of the system increases as well. The complexity multiplies by another factor when there is low temperature chemistry (LTC) involved as well in the simulation, whose pathways are significantly different from high temperature chemistry. Naturally, it is obvious that we need to develop methods to accelerate the flame simulation with efficient usage of large detailed kinetic models and minimal loss in accuracy. The next section summa-

rizes different class of methods developed over the decades to reduce the computational cost. This dissertation is also an extension of such methods that accelerate the chemistry source term integration in a turbulent combustion simulation, especially in the context of particle PDF type approaches.

1.3 Traditional approaches for accelerating chemistry source term integration

Numerous techniques have been developed in the past to reduce the computational cost of implementing combustion chemistry. They can be classified as skeletal reduction, dimension reduction and storage-retrieval methodologies.

The objective of the skeletal reduction is to eliminate unimportant species and reactions from a detailed kinetic model such that the governing equations can be solved with a reduced number of species while retaining reasonable accuracy. The most common skeletal reduction methods include principal component analysis (PCA) [43–45], sensitivity analysis [46, 47], direct relation graph (DRG) [48–50], Direct relation graph with error propagation (DRGEP) [51], path flux analysis (PFA) [52], the global pathway selection algorithms (GPS) [53] and chemical lumping [54, 55]. Out of these techniques, the graph based methods, such as DRG and DRGEP, are the most widely adopted ones due to their linear time reduction and effective error control. There exists a few methods which have combined DRG

and DRGEP with sensitivity analysis (DRGSA [56, 57] and DRGEP-SA [58], respectively) to generate even smaller but accurate chemical models. In the graph based methods, based on some user-specified targets, the species and reactions are ranked according to their importance. The least important species and its corresponding reactions are removed first, followed by an error measurement (typically some global parameters in a 0D or 1D simulation) compared to the detailed model. The removal of species and reactions continue until the error goes above a threshold. The smallest model that incurs an error below the threshold is used directly in a turbulent combustion simulation. Heberle *et al.* [59] developed a DRGEP based bottom-up building algorithm, which selectively adds important species and reactions from either scratch or on top of an already reduced model. The advantages of this building algorithm over any traditional top-down reduction method is that it can selectively add species and reactions directly at the reduced level. If a sub-mechanism needs to be added on top of a reduced model, unlike traditional approaches, the reduction process does not start from the beginning. Both DRGEP based reduction and building algorithm are used in this study quite extensively.

Skeletal reductions have been used in several studies to generate a global skeletal model and use it in a combustion simulation. In contrast to using a single skeletal mechanism, the adaptive chemistry method is another approach where multiple skeletal models are used depending on the local compositions, thus reducing the computational cost even further. This work uses the concept of adaptive chemistry extensively in Chapter 4 and 5 . A comprehensive review of the

adaptive chemistry approaches is provided in Sec. 1.5.

The dimension reduction techniques identify a low-dimensional manifold in the composition space, which typically results from the short time scales. The dimension reduction approaches include quasi-steady state approximation (QSS) [60, 61], computational singular perturbation (CSP) [62, 63], invariant constrained equilibrium edge preimage curve method (ICE-PIC) [64], rate controlled constrained equilibrium (RCCE) [65], intrinsic low dimensional manifolds (ILDm) [66], and one-dimensional slow invariant manifold (1D-SIM) [67].

In a storage-retrieval method, the reaction integration information is stored in a table, and then is used to build inexpensive approximate solutions at a later stage of computation, thus accelerating the overall reaction integration step. The most famous storage retrieval methods are *in-situ* adaptive tabulation (ISAT) [68, 69], artificial neural network (ANN) [70] and piecewise reusable implementation of solution mapping (PRISM) [71].

Apart from all these methods described above, *algorithmic speed-up* is achieved by the use of GPUs [72, 73], or the use of analytical Jacobian [74, 75] in any stiff chemistry solver, or even combining both of them [76, 77]. The use of analytical Jacobian and combining it with one of the most popular dimension reduction techniques, QSS, are integral part of this dissertation (Chapter 2 and 3). An extensive review of the use of analytical Jacobian in previous literature is provided in the following section.

1.4 Analytical Jacobian

Out of the several approaches to reduce the overall CPU cost of chemistry source term integration, and consequently to be able to use more detailed kinetic models, one category of techniques involve algorithmic speed-up, which means to develop more efficient solvers to integrate governing ODEs. The analytical Jacobian emerges while solving the governing ODE

$$\frac{d\Phi}{dt} = \mathbf{S}(\Phi) \quad (1.2)$$

in an implicit scheme, such as backward Euler, where the source term, \mathbf{S} , needs to be evaluated in the updated time step:

$$\frac{\Phi(t + \Delta t) - \Phi(t)}{\Delta t} = \mathbf{S}(\Phi(t + \Delta t)) = \mathbf{S}(\Phi) + \frac{\partial \mathbf{S}}{\partial \Phi} \Delta \Phi \quad (1.3)$$

where $\frac{\partial \mathbf{S}}{\partial \Phi}$ is the Jacobian matrix. The system of ODEs to solve for the species mass fraction and temperature source terms are inherently stiff. They require backward differentiation methods that need repeated Jacobian evaluations [78–81]. The CPU cost of using numerical Jacobian scales quadratically with the number of species, whereas the cost of using an exact analytical Jacobian is linearly dependent of number of reactions [82], as an exact analytical Jacobian provided to the solver reduces the number of iterations required to converge to the solution of the ODE within a given tolerance. Additionally, analytical Jacobian matrices are also an important component of computational diagnostic methods, such as computational singular perturbation (CSP) and chemical explosive mode analysis (CEMA), which exploit the spectral properties of the Jacobian matrix.

Motivated by the potential reduction in computational cost, a few analytical Jacobian generator packages have been developed in the last few years. Safta *et al.* [74] developed the TChem toolkit to calculate analytical Jacobian using density, the N_s species mass fraction vector and temperature T as the state vector. This package uses a functional interface to provide the analytical Jacobian rather than providing customized source code for different kinetic models. Youssefi [83] pointed out that the numerical Jacobian is not accurate enough to be used for eigen-value decomposition, and finding the explosive modes. The analytical Jacobian in [83] used the molar concentration of all species and temperature as the state vector. Bisetti [75] also developed analytical Jacobian matrix source code to reduce both computational cost and numerical error. Bisetti used adaptive symbolic analysis tools which provide mechanism specific routines that do not typically use sparse matrix algebra during Jacobian formulation. Different strategies were adopted to reduce the number of calls to ODE system function by providing a sparsity pattern. For instance, the DAEPACK automatic differentiation library [84–86] provides analytical Jacobian calculations along with sparsity pattern information, which reduces computational cost significantly when coupled to sparse ODE solvers [80]. Perini *et al.* [87] used analytical Jacobian for constant volume reactor which helps the matrix to be more sparse as the off-diagonal elements would only be non-zero because of direct species interactions and fall off or pressure dependent reactions. They observed a speed up by 2 to 2 orders of magnitude in case 0D constant volume simulations for smaller to larger mechanisms. They also reported a 77% faster result when the analytical Jacobian was used combined with tabulation

of temperature dependent properties in case of an engine simulation. Later, the sparse analytical Jacobian solver is also used for another engine simulation with multi-component fuels, where Perini *et al.* [88] reported a speed up of factor 7 with a 128 species detailed mechanism. Unfortunately, most of the combustion CFD tools are built on a constant pressure assumption, thus reducing sparsity of the Jacobian matrix. Recently GPU or hybrid CPU/GPU [89] have been used rather than pure CPU to speedup the computational time, which is another category of computational technique that maximizes the efficiency of algorithm. Dijkmans *et al.* [76] took a GPU based analytical Jacobian approach to accelerate the integration algorithm even further. They reported a speed up of factor 25 using only analytical Jacobian in case of the largest mechanism they used with approximately 3000 species. Using GPU with analytical Jacobian reduces the computational cost even further (almost a factor of 120). Niemeyer *et al.* [77] developed pyJac, an open source analytical Jacobian tool optimized for SIMD processors. This code can be easily parallelized with OpenMP. They showed a significant speed up in Jacobian evaluation compared to using finite difference based numerical Jacobian (upto a factor 7.5) and Tchem [74] (1.1 to 2 times).

Recently Hansen and Sutherland [90] argued about the choice of state vectors in an analytical Jacobian formulation in past literature. Jacobian matrices are inherently not uniquely defined due to the dependence of the governing variables: an energy variable (e.g. temperature), pressure, and chemical composition, as the governing equations of these variables are coupled with one another. The most common choice of state vector in an analytical Jacobian formulation involves tem-

perature (T) and either N_s species mass fractions (Y_i) [87, 91, 92] or concentrations (C_i) [75, 76] for both constant pressure and constant volume reactors, where N_s is the number of species in the kinetic mechanism. However, various other formulations have been developed to reduce the cost of Jacobian evaluation with different approximations or choices of state vectors, each of them leading to a different Jacobian expression. For instance, for a reactor with specified constant pressure, Xu *et al.* [93] chose the state vector to be N_s species mass fractions (Y_i), T , density (ρ), and total molar concentration (C). For the same configuration, Safta *et al.* [74] used only ρ , T , and the N_s species mass fractions (Y_i). The concentration-based formulation noticeably increases the sparsity of the resulting matrix compared to mass fraction-based one, consequently decreasing the cost of the simulation. In contrast, Niemeyer *et al.* [77] opted to use only $N_s - 1$ species mass fractions and T in pyJac. The variety of distinct options for state vector specification raises the question what (if any) the correct choice of thermo-chemical state vectors should be, especially because of the fact that the governing equations for any reactive flow problem are constrained by certain relations between state variables, e.g., the sum of the mass fractions must be unity, while using the mass fractions in the state vector. Similar constraints also exist between the concentrations. Hansen and Sutherland [90] argued that using a full mass fraction vector does not always satisfy the constraint, as the constraint is never explicitly used in the formulation. They mentioned the use of such over-specified state vectors as *inconsistent*. The use of these inconsistent state vectors leads to ambiguous derivative evaluations. They also speculated that such analytical Jacobian may lead to erroneous eigen-

value decomposition, or may even provide incorrect solution of the governing equations if a linear implicit solver is used rather than the most commonly used Newton-Krylov based solvers. Hansen and Sutherland [90] suggested the choice of state vector in pyJac [77] to be the correct one, where the first $N_s - 1$ species mass fraction ODEs are solved using the analytical Jacobian formulation, whereas the last (N_s -th) mass fraction to be explicitly evaluated using the mass conservation constraint. However, it is also true that a significant number of recent works have used analytical Jacobian extensively with full mass fraction space not only with Newton-Krylov solvers, but also with Rosenbrock method [94] and in CEMA [95–97]. In this work, we address this question of consistency while using all N_s mass fractions and temperature as state vector quite extensively, and attempt to provide a framework to deal with such inconsistencies.

1.5 Adaptive Chemistry

One way forward to mitigate the cost of using such large detailed mechanisms is to generate skeletal mechanisms by eliminating unimportant species and reactions [49, 51, 52, 58, 59, 61, 98, 99], as mentioned before. However, using only one reduced model may not be adequate in terms of reaching the level of reduction necessary for capturing complex fuel chemistry in a high fidelity turbulent combustion simulation of a realistic combustion device. Another step in reducing the cost of chemistry source term integration even further is to exploit the fact that

out of the variety of compositions encountered in a given turbulent reacting flow, only a few are chemically reactive. Thus, a large fraction of the composition space can be described properly with significantly smaller mechanisms. Consequently, the concept of *adaptive chemistry* tailors the fidelity and size of the kinetic models used for reaction integration to the local flame conditions. The reduced models can be generated on-the-fly or in an offline pre-processing stage.

The idea behind on-the-fly model reduction approach is to analyze the local composition at each time step, and then determine the smallest subset of the species required to accurately capture the chemistry for those compositions. Liang *et al.* [100] proposed a dynamically adaptive chemistry (DAC) approach, where the models generated using DRGEP [51] showed a factor of 30 speed-up in case of an HCCI simulation. The same approach has been applied in a PaSR [101] (with DRG [48] reduction), 1D unsteady methane/air freely propagating flame [102], 0D ignition [103], even in a large scale non-premixed ethylene flame [104]. It is important to note that on-the-fly approach is most time-efficient when the cost of the reduction algorithm is smaller compared to the efficiency gained by their use. To ensure such a requirement, several modifications of the DAC approach have been developed over the years. Sun *et al.* [105] developed a correlated DAC method, where the same reduced mechanism is utilized to integrate *similar* compositions. The similarity of different compositions are measured by a metric based on the normalized difference of mass fractions of fuel, O₂, OH, CH₂O, HO₂ and temperature. The method is validated for 0D ignition and 1D premixed flames in [105], and extended to DNS of a canonical turbulent premixed flame in a similar ap-

proach of finding similar compositions based on either consecutive time steps or immediate neighbors on a 7-point stencil [106]. The idea of avoiding redundant reduction of mechanisms by utilizing the same model for similar compositions are also used by Zhou *et al.* [107], where they proposed a clustered DAC method. This method creates clusters of encountered compositions at runtime based on the mass fractions of a subset of species using k-means algorithm. They used their algorithm for HCCI and DICI engine simulations, and observed a speed-up of factor 3 compared to the standard DAC, and a factor of 5 compared to using the detailed chemistry. On the other hand, an empirical criterion based on a progress variable and equivalence ratio is used by Shi *et al.* [108] to adaptively choose targets during the DRGEP reduction based on the local compositions.

An alternative to on-the-fly DAC approaches is to use an offline pre-processing stage to generate a set of reduced models, and use them adaptively during the actual simulation. In general, this class of approaches are found to be computationally less expensive as it avoids the computational cost of generating a significantly large number of reduced models at each time step without a significant loss of accuracy. Schwer *et al.* [109] developed one of the earliest adaptive chemistry approaches, where a library of reduced models were manually assembled based on expected compositions that may be encountered at runtime. The choice of a reduced model for a certain composition during the adaptive simulation was made based on temperature. The method was successfully applied with a speed up factor of 2 to 3 in a turbulent hydrogen shear layer flame and an axisymmetric laminar methane flame. On the other hand, for 1D laminar flames,

a speed-up of 3.5 was observed with adaptive proper orthogonal decomposition (POD) framework, proposed by Singer and Green [110]. Another adaptive chemistry approach was developed by Xu *et al.* [111] based on an approximate chemical explosive mode analysis (CEMA). In this method, the flame was segmented into different zones, and a set of reduced models generated previously based on low dimensional simulations were used adaptively for different zones. However, at the absence of an ideal load-balancing scheme, the gain in CPU cost was not significant. Oluwole *et al.* [112] proposed a rigorous method to identify the hyper-rectangles to find the ranges of validity of the reduced models in the full composition space. The hyper-rectangles were formed based on the compositions from a detailed simulation. However, the hyper-rectangles provide a conservative estimate of the validity ranges, as the actual regions of validity are highly non-convex and sometimes even disjoint. Banerjee and Ierapetritou [113] identified this non-convex regions in the temperature, fuel and oxidizer mass fraction space. Oluwole *et al.* [112] and Banerjee and Ierapetritou [113] validated their methods in the context of 1D freely propagating flames and PaSR simulations, respectively. D'Alessio *et al.* [114] divided an initial sample database into a user-specified number of regions based on an iterative local principle component analysis (LPCA) approach by minimizing the reconstruction error between the original and reconstructed dataset via PCA [115]. Then reduced models are generated for each cluster using DRGEP, which are then used adaptively during the actual simulation. A speed-up of factor 4 was observed for a laminar co-flow methane flame. D'Alessio *et al.* [116] compared the previous algorithm with a clustering approach based on Self-

Organizing Maps with K-Means (SKM), however found superior performances in case of using LPCA. However, another extension of the same approach [117] using artificial neural network (ANN) and feature extraction (FE) exhibited better performance compared to LPCA for a laminar co-flow n-heptane flame.

Most of the strategies mentioned above focus on adaptively using reduced models with a subset of species from the detailed mechanism, which decreases the CPU cost significantly, but they still carry the full composition vector at each point in time and space. Although keeping the full composition vector avoids the conceptual problem of mechanism augmentation, this approach becomes quite expensive in case of large mechanisms for complex fuels. Tosatto *et al.* [118] proposed a method to enable the removal of species and their reintroduction through a graph analysis modified to account for the transport process. However, the implementation of this method in an actual flame simulation is not straightforward. The issue of memory cost becomes even more acute for particle PDF type approaches, i.e., it can be 2 to 3 times more expensive compared to flamelet-type approaches. Liang *et al.* [119] developed a pre-partitioned adaptive chemistry (PPAC) approach tailored for particle PDF methods that addresses both CPU and memory cost. In this approach, based on a sample database, the composition space is partitioned into a user-specified number of regions based on short-time kinetics, i.e., the compositions belonging to the same region are expected to exhibit similar chemical kinetic characteristics. Then reduced models are generated using DRGEP [51] for each region. It is important to note that the sample database used in the pre-processing stage must be representative of the compositions encoun-

tered at runtime. During the actual simulation of interest, each composition at every time step is assigned to a region using a low-dimensional binary tree search algorithm, and then advanced in time using the reduced model corresponding to its region. Additionally, during the reaction integration step, each particle only retains the species mass fraction vector corresponding to its reduced model. The algorithm was examined for a non-premixed piloted propane-air case with a detailed mechanism (115 species), and reduced the CPU and memory cost by 85% and 70%, respectively compared to the detailed simulation. Newale *et al.* [120] combined the PPAC methodology with in-situ adaptive simulation (ISAT) and rate-constrained chemical equilibrium (RCCE) to reduce the cost even further. For the same PaSR run as in [119], they observed an additional 67% reduction in CPU cost and 38% reduction in memory requirement. The combined PPAC-ISAT approach was applied for an LES-PDF simulation of full scale Sandia Flame D in cylindrical configuration by Newale *et al.* [42], and observed a speed up of 45-55% for different set of reduced models compared to using ISAT with the detailed mechanism. The concept of PPAC is extensively utilized, and advanced even further in the present work.

1.6 Contribution of this work

The primary contributions of this dissertation are developing multiple new algorithms to reduce the computational cost of chemistry source term integration

in particle PDF simulation of turbulent combustion to enable an computationally efficient use of large chemical kinetic mechanisms.

As previously mentioned, the use of all N_s species mass fractions and temperature in the state vector is questionable [90] as it does not explicitly satisfy the mass conservation constraint. Such a choice of state vector may affect the eigen-decomposition of the Jacobian matrix, as well as the solution of the governing equations with certain type of solvers. An efficient way to resolve the issue is to use $N_s - 1$ mass fractions in the Jacobian formulation [77], and explicitly evaluate the last mass fraction using the mass conservation constraint. The last mass fraction is typically chosen to be the most abundant species in the system, e.g., N_2 . However, a more general analytical Jacobian framework is proposed here that uses N_s species mass fractions and temperature as the state vector, and still satisfies the mass conservation constraint. The framework uses a projection matrix to ensure that the mass fraction vector never goes off its $N_s - 1$ dimensional convex polytope or simplex, defined by the mass conservation constraint. Three different types of projection matrices along with their physical interpretations, advantages and disadvantages are discussed in great detail for different types of simulations. The framework also shows that with the projection matrix, the eigenvalues are found to be within round off errors of its counterpart obtained from pyJac formulation. Finally, this work shows that the framework is capable of providing accurate solutions using both Newton-Krylov and liner implicit solvers.

One of the most popular approaches to reduce the dimension of the system

of ODEs to solve, and also to reduce the stiffness of the chemical system is to consider a subset of species to be quasi-steady (QSS), whose production rates are very slow compared to their consumption rates. With such an assumption, the concentrations or mass fractions of those species can be obtained using algebraic expressions, which are computationally cheaper to solve than ODEs. The introduction of the QSS species is very simple and has been widely used in the last few decades. In this work, a new analytical Jacobian generator is developed that invokes the QSS species assumption in the chemical system. The algorithm is developed and fully implemented in Python, and right now is being integrated with the modular reduction software ARCANÉ [121]. Given a chemical mechanism and a set of QSS species, this algorithm provides the necessary Fortran routines related to QSS concentrations and their derivatives that can be readily used in any high-fidelity simulation.

Liang *et al.* [119] developed the PPAC algorithm, and validated it in PaSR for a propane oxidation case. Later, PPAC is combined with another dimension reduction method (RCCE) and a tabulation method (ISAT) [120]. This PPAC-RCCE-ISAT approach is also validated for the same simulation as [119]. Newale *et al.* [42] implemented the PPAC-ISAT method in an LES/PDF simulation of Sandia Flame D with only methane oxidation. None of them have used the PPAC approach (or combining it with other methods) in the context of predicting the secondary chemistry, e.g, NO_x formation. In this work, the PPAC algorithm is observed to be providing very conservative set of reduced models while capturing both oxidation and NO_x chemistry. To address that issue, a PPAC Additive framework is

developed, where in the pre-processing stage, the reduced models are developed in two steps. The first step is similar to traditional PPAC, where reduced models are developed capturing only the oxidation chemistry. Then important NO_x pathways are selectively added on top of the oxidation models in the second step using a building algorithm [59]. This PPAC Additive algorithm is validated for Sandia Flame D including both methane oxidation and NO_x production.

An important component of the PPAC pre-processing stage is its dependence on the initial database. The compositions in the database must represent the compositions encountered at runtime. In each of the previous validation cases of PPAC mentioned in the previous paragraph, including the validation of PPAC Additive, the database is obtained from the same simulation with the detailed mechanism. To avoid this issue, a novel dynamically partitioned adaptive chemistry (DPAC) algorithm is proposed in this work. In the pre-processing stage of DPAC, an initial partition is generated with a database that does not need to be representative of the actual simulation of interest. The initial database is typically generated using any low dimensional simulation, such as 0D homogeneous reactors. The reduced models in each region is initialized with a few species and zero reactions. Then during the adaptive simulation, the models are built at runtime based on the compositions encountered using the building algorithm [59], and the accessed region of the composition space during the simulation can also be re-partitioned using the same algorithm as in PPAC. The DPAC algorithm is validated for the propane oxidation case in PaSR, and an LES/PDF simulation of a lifted n-dodecane flame.

1.7 Outline

The thesis is divided into 6 chapters. The first chapter presents the introduction, which covers the background on different aspects of turbulent combustion simulation, and reviews the methods developed over the last few decades to reduce the cost of the turbulent reactive flow simulations. The summary of the work and potential future steps are discussed in Chapter 6. Chapter 2 to 5 contain the following material:

- Chapter 2 contains the details of the analytical Jacobian framework based on a full mass fraction vector and temperature. The framework provides accurate solution of the governing equations using different types of solvers, predicts the spectral properties accurately, and most importantly satisfies the mass conservation constraint at every time step.
- The algorithm to provide an analytical Jacobian with QSS species is outlined in Chapter 3. The algorithm is validated for 47 species n-dodecane mechanism including 18 QSS species, showing significant reduction in CPU cost.
- The PPAC-Additive algorithm is described in Chapter 4, which efficiently captures both primary and secondary combustion pathways in an adaptive chemistry framework. PPAC-Additive is validated for a PaSR and an LES/PDF simulation, both capturing the NO_x emission.
- A detailed description of the dynamically partitioned adaptive chemistry (DPAC) framework is provided in Chapter 5. The flexibility, efficiency and

accuracy of this algorithm is examined for a PaSR simulation involving propane/air non-premixed combustion, and a 2D LES/PDF simulation of lifted n-dodecane flame.

CHAPTER 2

A UNIFYING ANALYTICAL JACOBIAN FRAMEWORK FOR CHEMICAL KINETICS DYNAMICAL SYSTEMS

2.1 Abstract

A major challenge in simulating complex combustion systems with detailed chemical kinetic models is the cost of integrating the chemical source terms, often done using stiff ODE solvers that require frequent Jacobian evaluations. Using analytically derived Jacobians matrices instead of divided-difference-based numerical Jacobian approximations can significantly reduce the associated computational cost. However, ambiguities arise in the formulation of analytical Jacobians because the chemical state of the system, or state vector, can be expressed in multiple ways, involving variables that are typically not independent from one another. In this work, the consequences of those ambiguities on practical calculations are characterized in detail, and a generalized, projection-based framework is proposed as a mitigation strategy. Performances are assessed in a series of test cases involving a variety of configurations and numerical solution approaches. Results show that with proper treatment, commonly used analytical Jacobians formulations can be considered as equivalent for practical purposes, thereby alleviating concerns that the state vector chosen to express the governing equations and corresponding analytical Jacobian may significantly impact the accuracy of the simulations.

2.2 Introduction

The recent developments in combustion research increasingly highlight the importance of using detailed chemical kinetic models to accurately simulate and analyze reactive flow problems. The increase in size and complexity of these kinetic models in last few decades demands prohibitively large computational power as the memory requirements and CPU cost increase significantly with increasing number of species and reactions. To minimize such a large computational demand, a wide range of techniques has been developed in the literature focusing on *mechanism reduction*, e.g., skeletal reduction [49, 51, 59], dimension and time-scale reduction [62, 65, 66, 119, 120, 122] and chemical lumping [54, 123], and on *algorithmic speed-up*, i.e., more efficient solvers to integrate the governing ordinary differential equations (ODE). Due to the inherent stiffness of these ODEs, most kinetic models require robust backward differentiation methods that need repeated Jacobian evaluations. The computational cost of evaluating a numerical Jacobian matrix scales quadratically with the number of species, whereas matrix factorization scales cubically [82]. As opposed to that, the cost of using an exact analytical Jacobian is linearly dependent on the number of reactions [82]. Motivated by the potential improvements in both accuracy and cost in stiff ODE integration, analytical Jacobians are now used extensively in reactive flow solution techniques, leading to a significant increase in efficiency [74–77, 83, 87]. Additionally, analytical Jacobian matrices are also an important component of computational diagnostic methods, such as computational singular perturbation (CSP) and chemical explo-

sive mode analysis (CEMA), which exploit the spectral properties of the Jacobian matrix.

Jacobian matrices are inherently not uniquely defined due to the dependence of the governing variables: an energy variable (e.g. temperature), pressure, and chemical composition, as the governing equations of these variables are coupled with one another. The most common choice includes temperature (T) and either N_s species mass fractions (Y_i) [87, 91, 92] or concentrations (C_i) [75, 76] for both constant pressure and constant volume reactors, where N_s is the number of species involved in the kinetic mechanism. However, various other formulations have been developed to reduce the cost of Jacobian evaluation with different approximations or choices of state vectors, each of them leading to a different Jacobian expression. For instance, for a reactor with specified constant pressure, Xu *et al.* [93] chose the state vector to be N_s species mass fractions (Y_i), T , density (ρ), and total molar concentration (C). For the same configuration, Safta *et al.* [74] used only ρ , T , and the N_s species mass fractions (Y_i). The concentration-based formulation noticeably increases the sparsity of the resulting matrix compared to mass fraction-based one. In contrast, Niemeyer *et al.* [77] opted to use only $N_s - 1$ species mass fractions and T in `pyJac`, an analytical Jacobian generator for constant pressure reactors. The variety of possible combination of variables to express the state of a reactive system, along with the fact that those variables are typically not independent of one another, raises an important question: are some formulations more *correct* than others?

Conservation of matter imposes a number of constraints on the variables describing the state of the system. Choosing mass fractions as state variables, the most important constraint is expressed as

$$\sum_{i=1}^{N_s} Y_i = 1, \quad (2.1)$$

which can be geometrically interpreted as imposing that the mass fraction vector always lie in an $N_s - 1$ dimensional convex polytope, or simplex. Hansen and Sutherland [90] argued that using a dependent set of state variables, for example when using the full species mass fraction vector, does not always satisfy their constraints if these constraints are not explicitly used in the formulation of Jacobian. In such cases, the state vector may go off the simplex. In other words, such a choice of state vector is *inconsistent* [90] for Jacobian evaluation.

They also argued that these inconsistent state vectors may also result in ambiguous evaluations of partial derivatives. To quantify the effects of the choice of mass fraction vector, they compared a formulation using all N_s species as state vector to the formulation of `pyJac` [77], which explicitly enforces Eq. 2.1 by calculating the N_s^{th} species (typically, the diluent N_2) as the difference between 1 and the sum of the other $N_s - 1$ species mass fractions, and therefore, only uses $N_s - 1$ species in the Jacobian formulation. Although selecting N_s or $N_s - 1$ species state variables did not affect the final solution while using Newton-Krylov solvers, Hansen and Sutherland [90] speculated that such over-specification of state vectors (N_s species mass fractions) may result in some degree of inaccuracy while using a linear implicit solver for ODE integration, for example in the Rosenbrock method, or us-

ing the analytical Jacobian itself directly for eigen-decomposition, for example in CSP [124] or CEMA [95]. However, it is also true that a significant number of recent works have used analytical Jacobian extensively with full mass fraction space not only with Newton-Krylov solvers, but also with Rosenbrock method [94] and in CEMA [95–97].

Motivated by this review of the existing literature, it is important to look into a few fundamental questions regarding the Jacobian, its use and the choice of state vectors in its formulation. In particular, we explicitly account for the normalization constraint (Eq. 2.1) in this work, hence focusing only on the choice of keeping N_s or $N_s - 1$ species mass fractions in the state vector. The specific questions that we attempt to answer here are:

1. How does the choice of N_s or $N_s - 1$ mass fractions in the state vector impact the Jacobian evaluation, if the state vector lies in its simplex, or in other words the mass fraction vector satisfies the normalization constraint (Eq. 2.1)?
2. Most importantly, what will happen if the chosen state vector is not lying in the simplex, and thus *inconsistent*? For instance, can we incorporate the full mass fraction vector and temperature in the state vector during analytical Jacobian formulation, and still satisfy Eq. 2.1?
3. Does an analytical formulation with the full state vector provide accurate solutions, while using linear implicit solvers for ODE integration? How different will the solutions be from the solution using a Newton-Krylov solver?

4. Do the spectral properties of the Jacobian, which are important in the context of CSP and CEMA, depend on the choice of the state vector? How do the spectral properties get affected if the state vector is in or not in its simplex?

We first describe necessary definitions and notations required in this work in Sec. 2.3. Then we answer the first and second questions analytically for a simplified version of a chemically dynamic system in Sec. 2.4. We show in Sec. 2.5 that it is indeed possible to use the full mass fraction vector as a state vector for computing an analytical Jacobian that is consistent with the normalization constraint (Eq. 2.1) using three different choices. Next the governing equations of an actual chemical system are described in Sec. 2.6 followed by extending the idea of the previously mentioned three choices for such system in Sec. 2.7. The advantages and disadvantages of each of these choices are examined in Sec. 2.8 in the context of three different configurations: 0D auto-ignition; auto-ignition with mixing; and, a partially-stirred reactor (PaSR). Finally, the third and fourth questions are answered by verifying each of these choices using eigen-decomposition of the Jacobian matrix (Sec. 2.8.4), and with an open-source linear implicit solver based on the Rosenbrock method (Sec. 2.8.5), respectively.

2.3 Definitions and notations

Let \mathbf{x} be a vector of dimension N_s . \mathbf{x} is realizable, that is, can be interpreted as a mass fraction vector, if and only if each element of \mathbf{x} is positive

$$x_i \geq 0 \text{ for } i = 1..N_s \quad (2.2)$$

and the sum of all elements in \mathbf{x} is unity

$$\sum_{i=1}^{N_s} x_i = 1. \quad (2.3)$$

The ensemble of realizable vectors \mathbf{x} defines the realizable simplex \mathcal{R}

$$\mathcal{R} = \{ \mathbf{x} : x_i \geq 0, \mathbf{u}^T \mathbf{x} = 1 \}, \quad (2.4)$$

where \mathbf{u} is the unity vector of size N_s

$$\mathbf{u} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (2.5)$$

We further define the vector $\hat{\mathbf{x}}$ as the first $N_s - 1$ components of \mathbf{x} , such that

$$\mathbf{x} = \begin{bmatrix} \hat{\mathbf{x}} \\ x_{N_s} \end{bmatrix}. \quad (2.6)$$

Finally, we denote by \mathbf{y} a vector that always lies in \mathcal{R} . Following the notation in Eq. 2.6, and introducing \mathbf{v} as the unity vector of size $N_s - 1$, \mathbf{y} can be written

$$\mathbf{y} = \begin{bmatrix} \hat{\mathbf{y}} \\ y_{N_s} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}} \\ 1 - \mathbf{v}^T \hat{\mathbf{y}} \end{bmatrix} \quad (2.7)$$

Since \mathcal{R} is a simplex, the difference dy between two vectors in \mathcal{R} will lie in the tangent subspace of \mathcal{R} . This property is illustrated in two dimensions in Fig. 2.1. The tangent subspace of \mathcal{R} , which is also the orthogonal complement of \mathbf{u} , will be denoted by τ in the following.

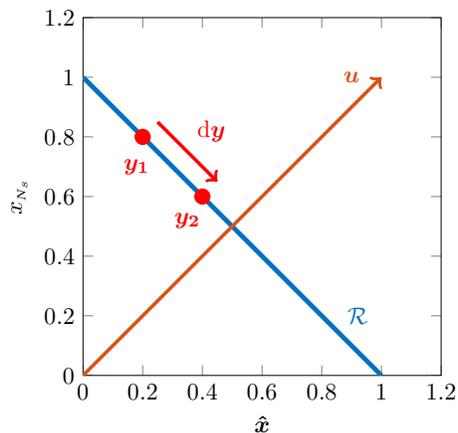


Figure 2.1: Illustration of the realizable simplex, \mathcal{R} , and unity vector, \mathbf{u} for $N_s = 2$. The difference between two realizable vectors, $d\mathbf{y} = \mathbf{y}_2 - \mathbf{y}_1$, lies in τ , the tangent subspace of \mathcal{R} and orthogonal complement of \mathbf{u} .

2.4 Impact of a state vector on the derivative of a scalar field

Any vector in the N_s dimensional space, \mathbf{x} can be written as

$$\mathbf{x} = \begin{bmatrix} \hat{\mathbf{x}} \\ x_{N_s} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}} \\ 1 - \mathbf{v}^T \hat{\mathbf{y}} \end{bmatrix} + \mathbf{u}w, \quad (2.8)$$

where w is a scalar. If $w = 0$, then \mathbf{x} is in \mathcal{R} , since we have in this case:

$$\mathbf{u}^T \mathbf{x} = \begin{bmatrix} \mathbf{v}^T & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} \\ 1 - \mathbf{v}^T \hat{\mathbf{y}} \end{bmatrix} = 1 \text{ for } \mathbf{x} \text{ in } \mathcal{R}. \quad (2.9)$$

Thus, \mathbf{y} can be seen as a parametrization of \mathcal{R} , with w being proportional to the normal distance of \mathbf{x} to \mathcal{R} . From Eq. 2.8,

$$d\mathbf{x} = \begin{bmatrix} d\hat{\mathbf{x}} \\ dx_{N_s} \end{bmatrix} = \begin{bmatrix} d\hat{\mathbf{y}} \\ -\mathbf{v}^T d\hat{\mathbf{y}} \end{bmatrix} + \mathbf{u}dw = \begin{bmatrix} d\hat{\mathbf{y}} + \mathbf{v}dw \\ -\mathbf{v}^T d\hat{\mathbf{y}} + dw \end{bmatrix}. \quad (2.10)$$

Considering the following initial value problem describing the evolution of \mathbf{x} in time

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (2.11)$$

where $f(\mathbf{x})$ is a scalar field defined as:

$$f(\mathbf{x}) = g(\hat{\mathbf{y}}) + h(\hat{\mathbf{y}}, w), \quad (2.12)$$

with $h(\hat{\mathbf{y}}, 0) = 0$ ($\forall \mathbf{x} \in \mathcal{R}$), which also implies that

$$\left. \frac{\partial h}{\partial \hat{\mathbf{y}}} \right|_{w=0} = 0. \quad (2.13)$$

In Eq. 2.12, g gives the value of f in \mathcal{R} and h gives the increment of f in the normal direction, i.e., parallel to \mathbf{u} . From Eq. 2.12, we obtain

$$\begin{aligned} df &= \frac{\partial f}{\partial \mathbf{x}} d\mathbf{x} = \begin{bmatrix} \frac{\partial f}{\partial \hat{\mathbf{x}}} & \frac{\partial f}{\partial x_{N_s}} \end{bmatrix} \begin{bmatrix} d\hat{\mathbf{x}} & dx_{N_s} \end{bmatrix}^T \\ &= \frac{\partial f}{\partial \hat{\mathbf{x}}} d\hat{\mathbf{x}} + \frac{\partial f}{\partial x_{N_s}} dx_{N_s} \end{aligned} \quad (2.14a)$$

From RHS of Eq. 2.12,

$$df = \frac{\partial g}{\partial \hat{\mathbf{y}}} d\hat{\mathbf{y}} + \frac{\partial h}{\partial \hat{\mathbf{y}}} d\hat{\mathbf{y}} + \frac{\partial h}{\partial w} dw \quad (2.14b)$$

Substituting Eq.2.10 in Eq.2.14a,

$$df = \frac{\partial f}{\partial \hat{\mathbf{x}}} (d\hat{\mathbf{y}} + \mathbf{v}dw) + \frac{\partial f}{\partial x_{N_s}} (-\mathbf{v}^T d\hat{\mathbf{y}} + dw) \quad (2.14c)$$

$$= \left(\frac{\partial f}{\partial \hat{\mathbf{x}}} - \frac{\partial f}{\partial x_{N_s}} \mathbf{v}^T \right) d\hat{\mathbf{y}} + \left(\frac{\partial f}{\partial \hat{\mathbf{x}}} \mathbf{v} + \frac{\partial f}{\partial x_{N_s}} \right) dw. \quad (2.14d)$$

By comparing Eq. 2.14b and Eq. 2.14d, we obtain the following transformation rule:

$$\begin{aligned} \frac{\partial}{\partial \hat{\mathbf{y}}} (g + h) &= \frac{\partial f}{\partial \hat{\mathbf{x}}} - \mathbf{v}^T \frac{\partial f}{\partial x_{N_s}}, \\ \frac{\partial h}{\partial w} &= \frac{\partial f}{\partial \hat{\mathbf{x}}} \mathbf{v} + \frac{\partial f}{\partial x_{N_s}}. \end{aligned} \quad (2.15)$$

We now investigate how df is impacted by two different cases. Considering \mathbf{x}_0 to be a realizable mass fraction vector at $t = 0$ (initial condition) and $\mathbf{x} = \mathbf{x}_0 + d\mathbf{x}$ to be the vector (may not necessarily lie in \mathcal{R}) after time dt , the two cases considered here are:

(a) $\mathbf{x}_0 \in \mathcal{R}, \mathbf{x}_0 + d\mathbf{x} \in \mathcal{R}, d\mathbf{x} \in \tau$

(b) $\mathbf{x}_0 \in \mathcal{R}, \mathbf{x}_0 + d\mathbf{x} \notin \mathcal{R}, d\mathbf{x} \notin \tau$

2.4.1 Case (a): If dx is in the tangent subspace, τ

We first consider the case where $dx \in \tau$, or in other words, for $x_0 \in \mathcal{R}$, $x_0 + dx \in \mathcal{R}$. Hence, $h = 0$ in Eq. 2.12 and consequently, $\frac{\partial h}{\partial \hat{y}} = 0$ (Eq. 2.13) and $dw = 0$. Substituting this in Eq. 2.10, we obtain

$$dx = \begin{bmatrix} d\hat{y} + vdw^0 \\ -v^T d\hat{y} + dw^0 \end{bmatrix} = \begin{bmatrix} d\hat{y} \\ -v^T d\hat{y} \end{bmatrix}. \quad (2.16)$$

Additionally, as $x = x_0 + dx \in \mathcal{R}$, we obtain

$$x = \begin{bmatrix} \hat{x} \\ 1 - v^T \hat{x} \end{bmatrix}, \quad (2.17)$$

and

$$dx = \begin{bmatrix} d\hat{x} \\ -v^T d\hat{x} \end{bmatrix}. \quad (2.18)$$

Substituting Eq. 2.18 into Eq. 2.14a, we get the following:

$$df = \frac{\partial f}{\partial x} dx = \left(\frac{\partial f}{\partial \hat{x}} - \frac{\partial f}{\partial x_{N_s}} v^T \right) d\hat{x}. \quad (2.19)$$

On the other hand, Eq 2.14b and 2.15 together can be rewritten for this case as:

$$df = \frac{\partial g}{\partial \hat{y}} d\hat{y} = \left(\frac{\partial f}{\partial \hat{x}} - \frac{\partial f}{\partial x_{N_s}} v^T \right) d\hat{x}. \quad (2.20)$$

Clearly Eq. 2.19 and 2.20 are identical. Hence, for dx in τ or alternatively $x_0 + dx$ in \mathcal{R} , df evaluated based on N_s species (Eq. 2.19) is identical to that based on $N_s - 1$

species (Eq. 2.20). It should be noted here that in Eq. 2.19, the $d\mathbf{x}$ is projected on \mathcal{R} parallel to N_s^{th} coordinate.

2.4.2 Case (b): If $d\mathbf{x}$ is not on the tangent subspace, τ

Next we consider the case where $\mathbf{x}_0 + d\mathbf{x}$ is not in \mathcal{R} , or in other words, $d\mathbf{x}$ is not in τ . While integrating Eq. 2.30 or 2.33, $d\mathbf{x}$ vector not being in the tangent subspace can possibly move the vector, $\mathbf{x}_0 + d\mathbf{x}$, away from \mathcal{R} at each time step. Consequently, in order for $\mathbf{x}_0 + d\mathbf{x}$ to be realizable, the $d\mathbf{x}$ vector must be projected back onto τ .

The projection of $d\mathbf{x}$ onto τ , denoted by $d\mathbf{y}$ (by definition, \mathbf{y} and $d\mathbf{y}$ always lie in \mathcal{R} and τ , respectively), can be written as $d\mathbf{y} = \mathbb{P}d\mathbf{x}$, where \mathbb{P} is the chosen projection matrix. At each time step of ODE integration, by construction the projected $d\mathbf{x}$ vector always stays on the tangent subspace. Consequently \mathbf{x} does not drift further off \mathcal{R} . Including the projection matrix \mathbb{P} , the derivative of the scalar field f becomes:

$$df = \frac{\partial f}{\partial \mathbf{x}} \mathbb{P} d\mathbf{x}, \quad (2.21)$$

where $\frac{\partial f}{\partial \mathbf{x}} \mathbb{P}$ is the modified gradient vector. This projection of $d\mathbf{x}$ onto τ , $d\mathbf{y}$, depends on the choice of the projection matrix, \mathbb{P} . Each choice leads to a specific projected solution of \mathbf{x} on \mathcal{R} , denoted by \mathbf{y} . Figure 2.2 shows three choices for projection matrices and their corresponding \mathbf{y} vectors for a given \mathbf{x} . These choices will be discussed in detail in the next section.

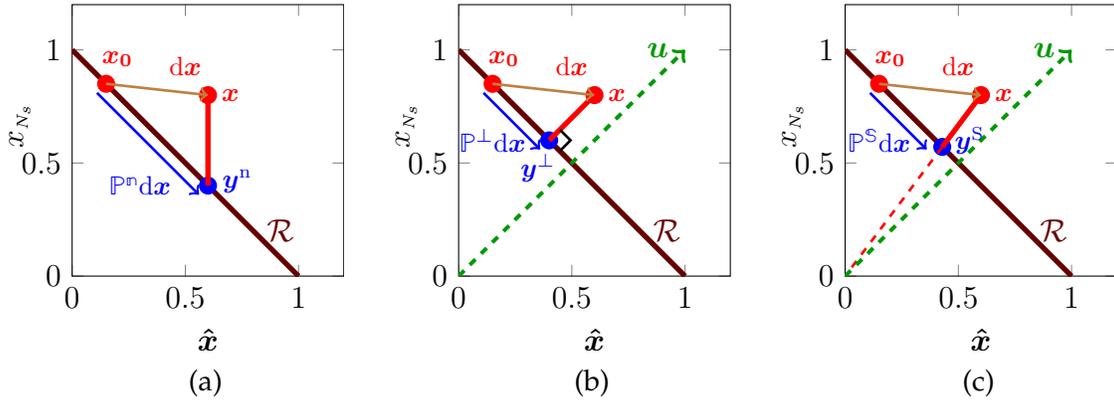


Figure 2.2: A 2-D interpretation of different projections: (a) parallel to N_s^{th} coordinate (P^n), (b) orthogonal projection (P^\perp), (c) scaling (P^s). In each figure, x corresponds to the state vector off the realizable simplex, \mathcal{R} and three y vectors with different superscripts correspond to the state vector after different projections.

2.5 Projection matrices, \mathbb{P}

We recall that the y vector always lies in \mathcal{R} . In this section, y vectors will be used with different superscripts, distinguished based on the choice of the projection options shown in Fig. 2.2.

2.5.1 Projection on the N_s^{th} coordinate

We observed in Section 2.4.1 that Eq. 2.19 and 2.20 provided identical results based on a projection in the N_s^{th} coordinate direction (Eq. 2.18), as only dx_{N_s} was altered. This observation guides the definition of the first projection investigated here. For

a given $d\mathbf{x}$:

$$d\mathbf{x} = \begin{bmatrix} d\hat{\mathbf{x}} \\ dx_{N_s} \end{bmatrix},$$

that is not on τ , the projected $d\mathbf{x}$ becomes

$$d\mathbf{y}^n = \mathbb{P}^n d\mathbf{x} = \begin{bmatrix} d\hat{\mathbf{x}} \\ -\mathbf{v}^T d\hat{\mathbf{x}} \end{bmatrix}, \quad (2.22)$$

where

$$\mathbb{P}^n = \begin{bmatrix} \mathcal{I}_{N_s-1} & 0 \\ \hline -\mathbf{v}^T & 0 \end{bmatrix} \quad (2.23)$$

and \mathcal{I}_{N_s-1} is the identity matrix of size N_s-1 . This projection matrix corresponds to the following relation between \mathbf{x} , which is not in \mathcal{R} and \mathbf{y}^n , the projected solution of \mathbf{x} in \mathcal{R} .

$$\mathbf{y}^n = \begin{bmatrix} \hat{\mathbf{y}}^n \\ 1 - \mathbf{v}^T \hat{\mathbf{y}}^n \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}} \\ 1 - \mathbf{v}^T \hat{\mathbf{x}} \end{bmatrix}. \quad (2.24)$$

The \mathbb{P}^n matrix only alters dx_{N_s} and projects $d\mathbf{x}$ onto τ parallel to the N_s^{th} species coordinate. Consequently, the \mathbf{x} vector projects itself back to \mathcal{R} parallel to N_s^{th} coordinate following Eq. 2.24, which is also graphically shown in Fig. 2.2a. This formulation will be subsequently referred as \mathbb{P}^n .

2.5.2 Orthogonal Projection

An orthogonal projection of $d\mathbf{x}$ onto τ is in the direction of \mathbf{u} (Fig. 2.2b) and the corresponding state vector \mathbf{x} projects on the realizable simplex, \mathcal{R} as Eq. 2.25, which is similar to Eq. 2.8.

$$\mathbf{y}^\perp = \begin{bmatrix} \hat{\mathbf{y}}^\perp \\ 1 - \mathbf{v}^T \hat{\mathbf{y}}^\perp \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}} \\ x_{N_s} \end{bmatrix} - \mathbf{u}w. \quad (2.25)$$

Summing over all the mass fractions on both side of Eq. 2.25 and after rearranging, we get:

$$w = \frac{\mathbf{u}^T \mathbf{x} - 1}{N_s}. \quad (2.26)$$

For such a conversion from \mathbf{x} to \mathbf{y}^\perp , the projected $d\mathbf{x}$ on τ becomes $d\mathbf{y}^\perp = \mathbb{P}^\perp d\mathbf{x}$, where

$$\mathbb{P}^\perp = \mathcal{I}_{N_s} - \frac{\mathbf{u}\mathbf{u}^T}{N_s}. \quad (2.27)$$

In Eq. 2.27, \mathcal{I}_{N_s} is an identity matrix of size N_s . The orthogonal projection considers the normal deviation of \mathbf{x} from \mathcal{R} in the direction parallel to \mathbf{u} and distributes the error equally into all species rather than altering only the N_s^{th} species as in \mathbb{P}^n . This formulation will be referred to as \mathbb{P}^\perp in this article.

2.5.3 Scaling

An alternative way of projecting \mathbf{x} onto \mathcal{R} is to scale or normalize all the species mass fractions, yielding the projected solution \mathbf{y}^S :

$$\mathbf{y}^S = \frac{\mathbf{x}}{S}, \quad (2.28)$$

where $S = \mathbf{u}^T \mathbf{x}$. For such a transformation, the projection of $d\mathbf{x}$ onto τ will be $d\mathbf{y}^S = \mathbb{P}^S d\mathbf{x}$, where the (i, j) th element of \mathbb{P}^S matrix is

$$\mathbb{P}_{i,j}^S = \frac{\delta_{ij}}{S} - \frac{x_i}{S^2}. \quad (2.29)$$

δ_{ij} is the Kronecker Delta function. This formulation projects the state vector \mathbf{x} onto \mathcal{R} in a direction towards the origin as shown in Fig. 2.2c. Unlike \mathbb{P}^n and \mathbb{P}^\perp , this projection is a non-linear function of \mathbf{x} . However, this choice of projection, referred to as \mathbb{P}^S subsequently, has the advantage of satisfying the individual boundedness of each mass fraction, provided that \mathbf{x} satisfies the individual boundedness. In contrast, even if all components of \mathbf{x} are positive, the other two projections can result in \mathbf{y}^n and \mathbf{y}^\perp having negative mass fractions.

The analysis until now involves a scalar field, $f(\mathbf{x})$, and its derivative df . The focus of this study is to extend the idea of projection on \mathcal{R} from $f(\mathbf{x})$ to an actual chemically dynamic system involving the source term vector, $\mathbf{S}(\mathbf{x})$. Before going into the details of the implementation, the next section revisits the governing equations concerned with such dynamic systems.

2.6 Governing equations of chemically dynamic systems

This study focuses on the time evolution of a homogeneous gaseous mixture in a constant-pressure or constant-volume system through the integration of a system of stiff ODEs using an analytical Jacobian. This initial value problem is defined by:

$$\frac{d\boldsymbol{\phi}(t)}{dt} = \mathbf{S}(\boldsymbol{\phi}(t)), \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \quad (2.30)$$

where $\boldsymbol{\phi}$ is the state vector, and \mathbf{S} is the rate of change due to chemical reactions (or chemical source term) with an initial state, $\boldsymbol{\phi}_0$. The state vector, $\boldsymbol{\phi}$, is a set of variables that uniquely defines the thermo-chemical state of the system at any given time t . In the current work, unless otherwise mentioned, $\boldsymbol{\phi}$ consists of all N_s species mass fractions and temperature, T , as the energy variable ($\boldsymbol{\phi} = \{\mathbf{Y}, T\}$). While integrating Eq. 2.30 using an implicit scheme, such as backward Euler method, the source term, \mathbf{S} , needs to be evaluated in the updated time step using the fundamental property

$$\mathbf{S}(\boldsymbol{\phi} + d\boldsymbol{\phi}) = \mathbf{S}(\boldsymbol{\phi}) + d\mathbf{S} = \mathbf{S}(\boldsymbol{\phi}) + \mathcal{J}d\boldsymbol{\phi}, \quad (2.31)$$

where the Jacobian, \mathcal{J} , is defined as,

$$\mathcal{J}_{i,j} = \left. \frac{\partial S_i}{\partial \phi_j} \right|_{\phi_{k,k \neq j}}. \quad (2.32)$$

To account for a more realistic situation where the mixture may not be considered homogeneous, the following modified governing equation will be used:

$$\frac{d\boldsymbol{\phi}(t)}{dt} = \mathbf{S}(\boldsymbol{\phi}(t)) + \mathbf{M}(t), \quad \boldsymbol{\phi}(0) = \boldsymbol{\phi}_0, \quad (2.33)$$

where the additional term \mathbf{M} is the rate of change due to transport, which will be referred as the *mixing term*, and evaluated by:

$$\mathbf{M}(t) = \frac{\phi'_e - \phi'}{\tau_{\text{mix}}}, \quad (2.34)$$

where ϕ' is a state vector consisting of mass fraction vector and enthalpy (h) as the energy variable ($\phi' = \{Y, h\}$), ϕ'_e is the equilibrium composition, and τ_{mix} is the specified mixing time scale, which is typically no smaller than 10^{-3} or 10^{-4} seconds [68]. Equation 2.33 is solved in this paper using a simple splitting method [68] in which the solution is advanced in two separate stages: a reaction fractional stage in which the chemical source term is integrated (Eq. 2.30), and a mixing fractional step (Eq. 2.34). While the former requires the use of the Jacobian matrix if an implicit or semi-implicit scheme is used, the latter can be integrated explicitly, thus does not require a Jacobian. After mixing fractional step, the state vector is converted from ϕ' to ϕ before chemical source term integration.

2.7 Implementation of Projection Matrix

The concept of the modified gradient of a scalar field, df , is extended in this section to the case of the source term vector, \mathbf{S} , and its derivative $d\mathbf{S}$, corresponding to a state vector, ϕ . Using the previous convention, the mass fraction vector, which is a subset of ϕ , is denoted by \mathbf{x} or \mathbf{y} (according to their definitions) in this section. If the mass fraction vector is in \mathcal{R} , the df vector is shown to be identical for choosing either N_s or $N_s - 1$ mass fractions to be independent. For both of these

choices, a similar conclusion is achieved as well in the supplementary section for the actual Jacobian matrices (Eq. 2.32). The eigenvalues and eigenvectors are also shown to be identical for both Jacobian matrices based on N_s and $N_s - 1$ species mass fractions, if the mass fraction vector is already in \mathcal{R} . However, projection is required if the mass fractions vector goes off \mathcal{R} . The implementation of any of the projections requires multiple modifications in the evaluation of both the chemical source term (RHS of Eq. 2.30) and the Jacobian (Eq. 2.32). These modifications are discussed below:

1. Extending the idea of the modified gradient of a scalar field (Eq. 2.21) to the case of \mathcal{S} , the modified Jacobian matrix, \mathcal{J}_{mod} , becomes,

$$\mathcal{J}_{\text{mod}} = \mathcal{J}\mathbb{P} = \frac{\partial \mathcal{S}}{\partial \boldsymbol{\phi}} \mathbb{P}, \quad (2.35)$$

where \mathbb{P} is either \mathbb{P}^n (Eq. 2.23) or \mathbb{P}^\perp (Eq. 2.27) or $\mathbb{P}^{\mathcal{S}}$ (Eq. 2.29) depending on the choice of projection.

2. An important question to answer is how to evaluate \mathcal{S} and \mathcal{J} with a mass fraction vector that is not realizable, especially because they require evaluation of the thermodynamic properties, e.g., specific heat and specific enthalpy of each species. Evaluation of these properties with a mass fraction vector \boldsymbol{x} that is off \mathcal{R} , is not well-defined. The issue is solved here is by projecting the state vector, \boldsymbol{x} , onto \boldsymbol{y} by using any of the aforementioned projections (Eq. 2.24 or 2.25 or 2.28), thus the source term and the Jacobian are evaluated in terms of \boldsymbol{y} rather than \boldsymbol{x} .¹ It should be noted that the projection

¹An alternative approach is to define the thermodynamic properties during source term and

matrix used to evaluate modified Jacobian matrix (Eq. 2.35) must be consistent with how \mathbf{x} is projected to \mathbf{y} to evaluate the thermodynamic properties.

At this junction, we make a few important points which are relevant to both the implementation and the consequences of using a projection matrix.

1. The analytical discussion in previous sections involve infinitesimal changes in time (dt) or mass fractions ($d\mathbf{x}$). However, the implementation of such projection matrices in a CFD simulation involves a finite time step or sub-step (Δt), thus a finite difference in mass fraction (Δx). As a result of that, projection at each time step introduces round-off errors, thus the solution vectors after ODE integration using different projections are not exactly identical, rather they remain within round-off errors. The impact of such round-off errors in different kind of CFD simulations are discussed in the next section.
2. **Difference in solution for different projections:** As mentioned before, the purpose of this projection matrix is to keep the $d\mathbf{x}$ vector in τ of \mathcal{R} so that \mathbf{x} stays in \mathcal{R} at each time step of the simulation. Different projections such as P^n , P^\perp and P^S will project the solution vector onto different points on \mathcal{R} (Fig. 2.2). However, if the initial condition, \mathbf{x}_0 , is in \mathcal{R} and $d\mathbf{x}$ is always in τ by using any of the projections, then \mathbf{x} at each time step will always lie in \mathcal{R} and the difference between the projected solution vectors corresponding to

Jacobian evaluation directly for a point $\mathbf{x} \notin \mathcal{R}$. For instance, the mixture enthalpy corresponding to a non-realizable mass fraction vector can be defined as $h = \frac{\sum_i Y_i h_i}{\sum_i Y_i}$ rather than the usual definition of $h = \sum_i Y_i h_i$. Both expressions are valid, and identical on \mathcal{R} . However, their derivatives $\frac{\partial h}{\partial Y_i}$ are different.

different projections remains of the order of round-off errors. This assertion will be supported by results in the next section.

2.8 Results and Discussion

This section covers a quantitative assessment of the accuracy of different projections, as applied to three different configurations, i.e., 0D auto-ignition, 0D auto-ignition with mixing, and a partially stirred reactor (PaSR). The motivation for choosing such configurations is detailed in the following sections. For each configuration, Eq. 2.33 is integrated using the commonly-used stiff solver DVODE [80].

To compare the performance of different projections, we also investigate one existing choice that includes only $N_s - 1$ species mass fractions and temperature in the state vector [77]. This formulation solves only N_s ODEs ($N_s - 1$ species mass fractions and temperature). Finally the N_s^{th} species mass fraction is evaluated using Eq. 2.1. This is one possible choice that always explicitly satisfies Eq. 2.1 by enforcing the error in the evaluation of $N_s - 1$ mass fractions into the N_s^{th} species. The N_s^{th} species for such formulation is usually chosen to be the most abundant species in the system, e.g., N_2 . This formulation will be referred as \mathbb{R}^ϕ (reduced state vector) in the rest of the article. We recognise that \mathbb{R}^ϕ is a practical solution which works in most of the reactive flow simulation with reasonable accuracy to ensure the mass conservation constraint at every time step, thus we are comparing the accuracy of the three projection cases with respect to \mathbb{R}^ϕ . However, we

do not agree that this is the only possible solution as suggested by Hansen and Sutherland [90], and the concept of projection matrix provides a more general (using all mass fractions in the state vector) and mathematically rigorous solution than dumping all the errors in the most abundant species, as in \mathcal{R}^ϕ .

2.8.1 0D auto-ignition

To assess the accuracy of projections, two constant-pressure auto-ignition cases are examined using an 88 species skeletal mechanism of n-heptane developed by Yoo *et al.* [125]. These two cases correspond to heptane-air and heptane-oxygen mixture, with an initial temperature, pressure and equivalence ratio of 800 K, 10 bar and 1 (stoichiometric), respectively.

Table 2.1 compares the results for these two auto-ignition cases between \mathcal{R}^ϕ and three different choices of projections. The integration performance of DVODE, quantified using number of steps (NST), number of source term (NFE) and Jacobian evaluations (NJE), are almost the same for \mathcal{R}^ϕ , \mathcal{P}^n and \mathcal{P}^s , but slightly higher for the orthogonal projection (\mathcal{P}^\perp).

As expected, the maximum deviation of sum of the mass fractions from \mathcal{R} is always of the order of round-off error for all projections. This observation proves the numerical viability of using the full mass fraction space and temperature as state vector with proper projection without violating the joint boundedness constraint. During the simulation, the individual boundedness of each species can be

Table 2.1: Comparison of integration statistics, conservation properties and error norms between R^ϕ and different choice of projections for two 0D constant pressure auto-ignition cases: heptane-air and heptane-oxygen. NST, NFE and NJE refer to number of steps, number of source term evaluations and number of Jacobian evaluations during the full auto-ignition simulation. For R^ϕ , the N_s^{th} species is considered to be N_2 . The time stepping for these two auto-ignition cases are considered to be 10^{-6} and 10^{-7} s respectively. Two cases were run for 5,000 and 10,000 time steps respectively. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15} .

<i>Heptane - Air</i>				
Parameters	R^ϕ	P^n	P^\perp	P^S
NFE	152×10^3	152×10^3	200×10^3	152×10^3
NST	95×10^3	95×10^3	139×10^3	95×10^3
NJE	5,103	5,108	5,673	5,103
$\max \left 1 - \sum_{i=1}^{N_s} Y_i \right $	–	1.64×10^{-14}	6.99×10^{-15}	1.97×10^{-14}
No of negative Y_i	3,965	4,278	112×10^3	3,664
Maximum negative Y_i	2.20×10^{-16}	3.39×10^{-16}	7.06×10^{-16}	2.66×10^{-16}
ϵ_{mean}	0 (Ref)	1.19×10^{-9}	1.72×10^{-7}	1.19×10^{-9}
ϵ_{max}	0 (Ref)	6.63×10^{-7}	9.50×10^{-5}	6.63×10^{-7}
<i>Heptane - Oxygen</i>				
Parameters	R^ϕ	P^n	P^\perp	P^S
NFE	225×10^3	224×10^3	275×10^3	224×10^3
NST	130×10^3	130×10^3	181×10^3	130×10^3
NJE	10,113	10,110	10,217	10,107
$\max \left 1 - \sum_{i=1}^{N_s} Y_i \right $	–	4.25×10^{-13}	3.51×10^{-14}	4.23×10^{-13}
No of negative Y_i	2,756	1,537	170×10^3	1,480
Maximum negative Y_i	1.46×10^{-14}	1.42×10^{-15}	8.94×10^{-16}	3.29×10^{-16}
ϵ_{mean}	0 (Ref)	6.81×10^{-9}	3.96×10^{-7}	6.81×10^{-9}
ϵ_{max}	0 (Ref)	1.64×10^{-5}	9.31×10^{-4}	1.64×10^{-5}

violated, i.e., it is possible to have negative mass fractions as tabulated in Table 2.1. This may occur either as a result of ODE integration (usually for the species with very small mass fractions), which is dependent on the ODE solver, or due to the projection itself. We observe that the orthogonal projection always suffers from a large number of negative mass fractions, although the maximum negative mass

fractions are always of the order of round-off error. The orthogonal projection alters all the mass fractions (Eq. 2.25), consequently there is a high likelihood that the species with zero or very small mass fractions to achieve a negative value after projection. The similar disadvantage is also applicable for R^ϕ as the N_s^{th} mass fraction is evaluated using Eq. 2.1. Specially for any oxy-fuel combustion case (heptane-oxygen), the N_2 mass fraction, which is always zero, reaches a negative value in R^ϕ formulation more than P^n and P^S . It is also important to note that the fundamental idea behind P^n and R^ϕ is very similar as both of them alter the N_s^{th} species with the only exception of solving ODEs for N_s mass fractions in case of P^n as opposed to $N_s - 1$ in case of R^ϕ . This makes the projection method a more general approach as it does not enforce the errors of calculating all species mass fraction in one single species. As a result, for heptane-oxygen case, the number of negative mass fractions are significantly lower in P^n than R^ϕ . This is the reason why R^ϕ is not necessarily the best choice, especially when it is oxy-combustion, where there is no obvious abundant species in the system. In the case of P^S , the projection is always toward the origin (Fig. 2.2c), which ensures that there are no negative mass fractions due to the projection itself. Consequently, the number of negative mass fractions is the lowest for the scaling in case of both auto-ignition runs.

The error measures in Table 2.1 take R^ϕ as a reference case and compare the species mass fraction trajectories of all three projections. It should be noted that the exact solutions do not exist for these problems, thus both R^ϕ and projection-based formulations are inaccurate and obtained using the same ODE tolerances.

However, for the sake of comparison, the errors evaluated in this article simply quantify the difference between the widely-used reference case, R^ϕ , and different projections, suggested in this article. The error, ϵ , at each time step is the norm of absolute errors of all species mass fractions for each projection compared to R^ϕ , normalized by the number of species. The error norm at the m^{th} time step, ϵ_m , is defined as follows:

$$\epsilon_m = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} (Y_{i,m}^{R^\phi} - Y_{i,m}^P)^2}, \quad (2.36)$$

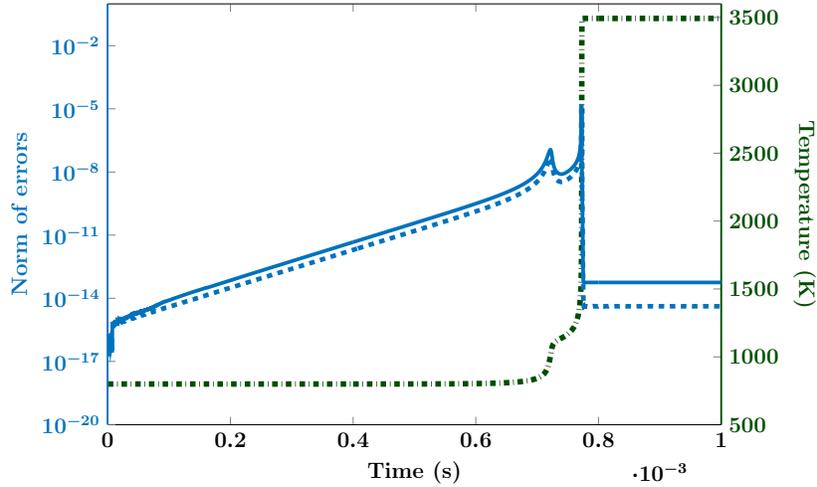


Figure 2.3: The norm of errors: between R^ϕ and P^n at each time instant for a 0D auto-ignition case of heptane-oxygen mixture (solid blue line), and between the actual and perturbed R^ϕ (dashed blue line). The plot also depicts the evolution of temperature during ignition (dash and dot green line). Both absolute and relative tolerances for ODE integration are considered to be 10^{-15} .

where $Y_{i,m}$ is the mass fraction of the i^{th} species at the m^{th} time step, and superscript P corresponds to any of the three projections. The mean and maximum of the norms over all time steps are denoted by ϵ_{mean} and ϵ_{max} , respectively. Clearly, P^n and P^S exhibit the least errors, whereas P^L contains a higher magnitude of er-

rors, which can be attributed to the significantly higher number of negative mass fractions. However, a mean error of $\mathcal{O}(10^{-9})$ (\mathbb{P}^n and \mathbb{P}^S) with an absolute and relative tolerance of 10^{-15} during ODE integration is significantly high, which needs more justification. Figure 2.3 shows the growth of the norm of errors between \mathbb{R}^ϕ and \mathbb{P}^n (solid blue line) over time during the auto-ignition case of heptane-oxygen mixture along with the evolution of temperature (dash and dot green line). We attribute this significantly high error to the inherent sensitivity of an auto-ignition simulation, as there exists at least one explosive mode (positive eigenvalue of the Jacobian matrix). Hence, an initial round-off error between \mathbb{R}^ϕ and \mathbb{P}^n increases exponentially over time during ignition. At the ignition point, both the positive eigenvalue and the error reach their corresponding maximum values simultaneously. After ignition, the error drops again while an equilibrium composition and temperature are reached, as there does not exist any explosive mode.

To confirm the issue with auto-ignition simulation, we perform another simulation of \mathbb{R}^ϕ with a slightly perturbed initial condition. The initial condition for this perturbed \mathbb{R}^ϕ case is chosen to be a solution vector that corresponds to an error norm of $\mathcal{O}(10^{-15})$ compared to the unperturbed \mathbb{R}^ϕ at the beginning. Then the error norm is evaluated between the perturbed and the unperturbed \mathbb{R}^ϕ case over time, and plotted in Fig. 2.3 (dashed blue line). It is evident that the initial perturbation increases exponentially even with the same formulation (\mathbb{R}^ϕ) and reaches a maximum value at the ignition point. Consequently, an auto-ignition case is not the ideal configuration to assess the performance of these projections. This issue with an auto-ignition simulation is explored further with a simple analytical

model in 2.10.1.

To emphasize the point even further that such a large error is not a manifestation of any of the projections, rather due to the inherent sensitivity of an auto-ignition model, we analyzed the heptane-oxygen mixture case with quadruple precision during ODE integration. The results are tabulated and explained in 2.10.2.

For a more suitable comparison, we investigate the accuracy of these projections with more realistic models: auto-ignition with mixing and PaSR. Unlike 0D auto-ignition, both these models involve mixing, which helps ignition to occur much faster, thus the error cannot increase as rapidly as in the auto-ignition case.

2.8.2 0D auto-ignition with mixing

The governing equation of an auto-ignition with mixing problem contains both the chemical source term and a mixing term (Eq. 2.33). Similar to the auto-ignition cases, two constant-pressure auto-ignition with mixing cases are examined, i.e., heptane-oxygen and heptane-air mixtures with the same initial conditions as before. The equilibrium composition and temperature (ϕ_e) are evaluated using Cantera [126] for the mixing term. The mixing time scale for both mixtures is specified as 10^{-3} seconds.

Table 2.2 compares the same parameters as for the auto-ignition cases between

Table 2.2: Comparison of integration statistics, conservation properties and error norms between \mathcal{R}^ϕ and different choice of projections for two 0D constant pressure auto-ignition with mixing cases: heptane-air and heptane-oxygen. NST, NFE and NJE refer to number of steps, number of source term evaluations and number of Jacobian evaluations during the full simulation. For \mathcal{R}^ϕ , the N_s^{th} species is considered to be N_2 . The time stepping for these two cases are considered to be 10^{-7} and 10^{-8} s. These two cases were run for 7,000 and 6,000 time steps respectively. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15} .

<i>Heptane - Air</i>				
Parameters	\mathcal{R}^ϕ	\mathcal{P}^n	\mathcal{P}^\perp	\mathcal{P}^S
NFE	894×10^3	893×10^3	940×10^3	893×10^3
NST	727×10^3	726×10^3	802×10^3	726×10^3
NJE	15,480	15,441	18,956	15,541
$\max \left \mathbf{1} - \sum_{i=1}^{N_s} Y_i \right $	-	1.17×10^{-13}	2.51×10^{-14}	1.31×10^{-13}
No of negative Y_i	92×10^3	101×10^3	288×10^3	99×10^3
Maximum negative Y_i	1.79×10^{-30}	2.04×10^{-30}	5.42×10^{-15}	2.51×10^{-30}
ϵ_{mean}	0 (Ref)	8.49×10^{-15}	1.55×10^{-14}	9.87×10^{-15}
ϵ_{max}	0 (Ref)	4.94×10^{-14}	5.43×10^{-13}	2.05×10^{-13}
<i>Heptane - Oxygen</i>				
Parameters	\mathcal{R}^ϕ	\mathcal{P}^n	\mathcal{P}^\perp	\mathcal{P}^S
NFE	987×10^3	987×10^3	991×10^3	986×10^3
NST	811×10^3	812×10^3	826×10^3	812×10^3
NJE	17,280	17,023	17,307	17,014
$\max \left \mathbf{1} - \sum_{i=1}^{N_s} Y_i \right $	-	6.06×10^{-13}	2.35×10^{-13}	5.72×10^{-13}
No of negative Y_i	25,882	30,290	96,138	28,724
Maximum negative Y_i	8.66×10^{-15}	3.32×10^{-18}	2.55×10^{-15}	1.07×10^{-17}
ϵ_{mean}	0 (Ref)	2.03×10^{-14}	2.37×10^{-13}	5.08×10^{-14}
ϵ_{max}	0 (Ref)	3.14×10^{-12}	4.74×10^{-11}	1.19×10^{-11}

\mathcal{R}^ϕ and three different projections. The trend of integration statistics, maximum deviation from \mathcal{R} , number of negative mass fractions and maximum negative mass fraction turn out to be similar to the previous auto-ignition model and the explanations are similar as well. However, the motivation behind the ignition-mixing model was the error measures for different projections assuming \mathcal{R}^ϕ as the reference case. The norm of error at each time step is evaluated using Eq. 2.36.

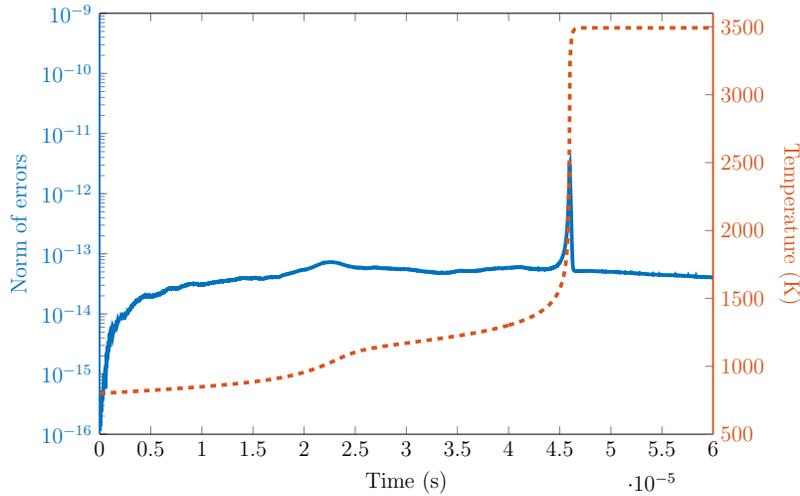


Figure 2.4: The norm of errors between R^ϕ and P^n at each time instant for an ignition-mixing model of heptane-oxygen mixture. The plot also depicts the evolution of temperature during ignition. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15} .

As expected, for an ignition-mixing model, the mean (ϵ_{mean}) and max (ϵ_{max}) norm of errors are orders of magnitude less compared to the 0D auto-ignition case. For instance, the mean errors in case of P^n are $O(10^{-9})$ and $O(10^{-14})$ for auto-ignition and ignition-mixing model respectively.

Figure 2.4 shows the error between R^ϕ and P^n for heptane-oxygen mixture at each time instant of the ignition-mixing model. Although in an ignition-mixing model, there still exists an explosive mode or a positive eigenvalue of Jacobian, the time to reach the equilibrium composition drops significantly due to mixing, thus an initial round-off error does not increase as rapidly as observed in Fig. 2.3. This phenomenon is further explained by comparing a simplified idealized ignition-mixing model with an auto-ignition model in 2.10.1. The norm of errors between

different projections and \mathbb{R}^ϕ shown in Table 2.2 proves the accuracy of all the projections compared to the existing method of handling the state vector. The error in the case of the orthogonal projection is slightly higher than the other two, which, as before, is attributed to the larger number of negative mass fractions. But the overall performance shows the viability of using the full state vector along with proper projections without violating the normalization constraint (see Table 2.2).

2.8.3 Partially Stirred Reactor (PaSR)

PaSR is represented by a number of notional reactive particles containing their own species composition and temperature. The composition of each particle evolves in time due to inflow/outflow/pairing, mixing and reaction. A characteristic residence time corresponds to the inflow/outflow rate inside the reactor. In the inflow/outflow step, an even number of particles are randomly selected with equal probability and their compositions are replaced by the inflow composition. Each particle is assigned to a partner and the pairing of these two particles, specified by a pairing time, provides a local mixing process inside the reactor. A detailed description of the PaSR system and methodology is provided by Pope [68] and Liang *et al.* [119]. The PaSR parameters used for the simulations are tabulated in Table 2.3. There are three streams with normalized mass flow rates of 0.057, 0.893 and 0.05 kg/s, containing pure fuel (300 K), air (300 K), and their equilibrium composition, respectively. The same heptane skeletal mechanism [125] is used for the PaSR simulations. All the simulations are performed for 10 residence

times. There exists a transient period for a few initial residence times, after which the PaSR reaches a statistically-stationary state.

Table 2.3: Specification of the parameters for PaSR case. Species are given as mass fractions and the initial condition is same as Stream 1.

Parameters	Values
<i>Reactor characteristics</i>	
Pressure (p)	10 bar
Number of particles (n_p)	100
Pairing time (τ_{pair})	1 ms
Mixing time (τ_{mix})	1 ms
Residence time (τ_{res})	10 ms
Time step (Δt)	0.1 ms

Similar to the ignition-mixing model, the motivation behind using PaSR as a configuration is to assess the performance of different projections in a more realistic model than 0D auto-ignition, where the inherent explosive mode leads to a rapid increase in error. For the error evaluation, the \mathbb{R}^ϕ case is considered to be the reference. The norm of errors at the m^{th} time step, calculated over all species mass fractions and particles, is given by

$$\epsilon_m = \sqrt{\frac{1}{N_s n_p} \sum_{l=1}^{n_p} \sum_{i=1}^{N_s} (Y_{i,m,l}^{\mathbb{R}^\phi} - Y_{i,m,l}^{\text{P}})^2}, \quad (2.37)$$

where $Y_{i,m,l}$ is the mass fraction of the i^{th} species in the l^{th} particle at the m^{th} time step, and superscript P corresponds to any of the three projections. The mean and maximum of the norms over all time steps are denoted by ϵ_{mean} and ϵ_{max} , respectively and these are tabulated in Table 2.4.

As expected, all the error norms are of the order of round-off error compared to \mathbb{R}^ϕ , which shows the feasibility of using all species mass fractions and temperature

Table 2.4: Comparison of error norms between different choice of projections and R^ϕ in PaSR. The ODE integration tolerances (absolute and relative) are considered to be 10^{-15} .

Parameters	P^n	P^\perp	P^S
ϵ_{mean}	3.85×10^{-14}	2.84×10^{-13}	3.83×10^{-14}
ϵ_{max}	3.29×10^{-12}	1.40×10^{-11}	3.28×10^{-12}

as state vector with proper projections and still not violate the joint boundedness constraint of Eq. 2.1. The norm of errors between R^ϕ and P^n at every time instant is plotted in Fig 2.5. Unlike the auto-ignition and ignition-mixing models, the error does not increase rapidly, but remains statistically steady at $O(10^{-14})$ throughout the simulation, which can be attributed to the inherent nature of PaSR involving mixing and inflow/outflow.

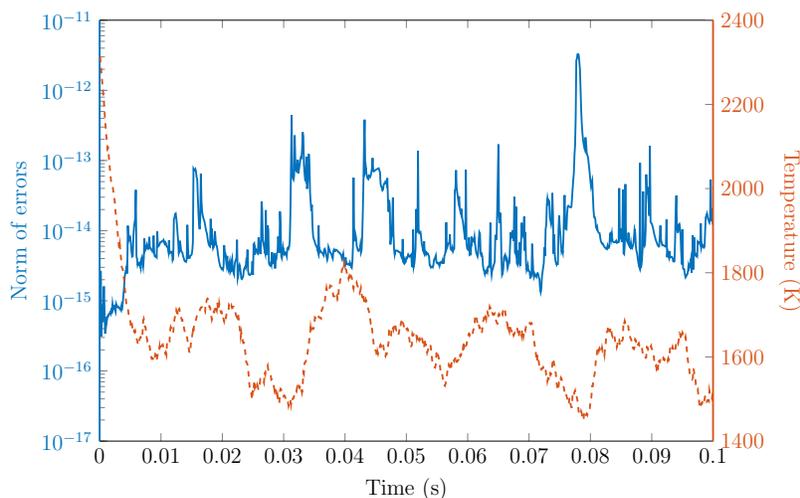


Figure 2.5: The norm of errors between R^ϕ and P^n at each time instant for the PaSR model. The plot also depicts the evolution of temperature during the simulation. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15} .

2.8.4 Eigen-decomposition of Jacobian matrix

Hansen and Sutherland [90] stated that an analytical Jacobian considering a full mass fraction state vector may affect the accuracy of different flame diagnostic tools, which are based on the eigen-decomposition of Jacobian matrix. There exists two methods of time scale analysis, i.e., CSP [124] and CEMA [95], where the concept of explosive modes in a reactive flow is introduced based on the eigenvalues of the Jacobian matrix. Our objective in this article is to examine whether the use of a full mass fraction space as state vector with projection affects the spectral properties of the analytical Jacobian matrix.

We choose a constant pressure 0D auto-ignition case of heptane-oxygen stoichiometric mixture with initial temperature and pressure of 800 K and 10 bar as a test case to investigate the accuracy of the projection approach. For an auto-ignition case, the trajectory of the maximum positive real part of the eigenvalue (λ_{\max}) is followed throughout the simulation and the zero crossing or the crossover point of λ_{\max} is considered to be the ignition point (t_{ig}) [95], which is the most important quantity in the context of an auto-ignition case. Other than auto-ignition simulations, the crossover point is also important to identify the flame front in 1D flame or direct numerical simulation of turbulent flames [95].

In the eigen-decomposition of a Jacobian matrix, there exists one conservative mode (zero eigenvalue) associated with energy conservation along with M other modes with zero eigenvalues associated with element conservation, where M is

the number of elements. To obtain the zero eigenvalue corresponding to energy, it is convenient to use an alternative state vector $\phi' \equiv \{Y, h\}$, where enthalpy, h , is used as the energy variable replacing T . The transformation from $\{Y, T\}$ state vector to $\{Y, h\}$ is straightforward and does not impact the eigenvalues or the eigenvectors of the Jacobian matrix. The eigenvalues of the Jacobian matrix are evaluated using LAPACK [127].

The trajectories of λ_{\max} of the Jacobian matrix with different projections are compared to the case of \mathbb{R}^ϕ and, the mean and maximum relative errors are tabulated in Table 2.5. The same table also compares the crossover points (two stage ignition) of λ_{\max} , t_{ig}^1 and t_{ig}^2 , for each projection to \mathbb{R}^ϕ . The exact crossover points are evaluated by linear interpolation between two nearest points on each side of the crossover point. All the simulations were run for 10,000 time steps with a time step of 10^{-7} s.

Table 2.5: Comparison of relative errors of the maximum positive eigenvalue and crossover points between different choice of projections and \mathbb{R}^ϕ in a 0D auto-ignition simulation of heptane-oxygen mixture.

Parameters	\mathbb{P}^n	\mathbb{P}^\perp	\mathbb{P}^S
ϵ_{mean}	2.00×10^{-6}	1.18×10^{-4}	2.00×10^{-6}
ϵ_{max}	0.003	0.165	0.003
t_{ig}^1	4.22×10^{-11}	2.29×10^{-9}	4.22×10^{-11}
t_{ig}^2	7.84×10^{-12}	4.83×10^{-10}	7.84×10^{-12}

The relative error at each time instant between \mathbb{R}^ϕ and \mathbb{P}^n is plotted in Fig 2.6. It is evident from the figure that the error is maximum at the two ignition points, which is because of the exponential increase of error up to t_{ig} , the same as in Fig. 2.3, due to the inherent sensitivity of an auto-ignition model.

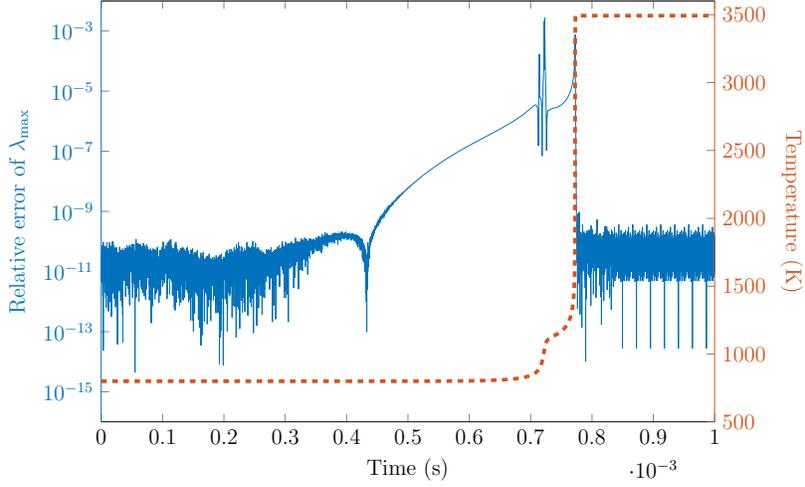


Figure 2.6: The relative error of λ_{\max} between \mathbb{R}^ϕ and \mathbb{P}^n at each time instant for an auto-ignition case of heptane-oxygen mixture. The plot also depicts the evolution of temperature during ignition. Both absolute and relative tolerances for ODE integration are considered to be 10^{-15} .

As a natural extension of an auto-ignition model, as discussed before, we investigate the performance of the projections in a constant pressure auto-ignition with mixing case of heptane-oxygen mixture with the same initial conditions as the auto-ignition case. The relative errors of λ_{\max} and t_{ig} for this ignition-mixing model are tabulated in Table 2.6. All the simulations were run for 7,000 time steps with a time stepping of 10^{-8} s.

Table 2.6: Comparison of relative errors of the maximum positive eigenvalue and crossover points between different choice of projections and \mathbb{R}^ϕ in a 0D auto-ignition with mixing simulation of heptane-oxygen mixture.

Parameters	\mathbb{P}^n	\mathbb{P}^\perp	\mathbb{P}^S
ϵ_{mean}	2.62×10^{-7}	4.54×10^{-6}	2.50×10^{-7}
ϵ_{max}	1.03×10^{-5}	8.88×10^{-5}	1.00×10^{-4}
t_{ig}^1	1.68×10^{-16}	6.19×10^{-16}	1.37×10^{-16}
t_{ig}^2	1.09×10^{-17}	5.31×10^{-16}	1.47×10^{-17}

As expected, both mean and maximum relative errors of λ_{\max} have decreased from the ignition case, and most importantly, the accuracy in prediction of crossover point remains within machine error. Thus, the use of an analytical Jacobian with full state vector and proper projection is completely justified not only for integrating ODEs, but also in CSP or CEMA, where the spectral properties of exact analytical Jacobian is required.

2.8.5 Linear implicit solver: Rosenbrock method

Apart from different diagnostic tools that use eigen-decomposition of Jacobian matrix, Hansen and Sutherland [90] also mentioned a second example of linear implicit solvers for ODE integration, such as Rosenbrock Method, where an inaccurate Jacobian may be critical. In the following discussion, a user specified time-step is referred as ‘time-step’ and the internal solver time levels within a time-step is referred as ‘sub-step’. Unlike DVODE, the linear-implicit solvers do not perform iterations at each sub-step, rather they implicitly evaluate the solution at each sub-step within a time-step [90]. For Rosenbrock methods, the Jacobian is evaluated once at each sub-step to update the solution [128], thus accuracy of Jacobian becomes more important compared to DVODE, where one Jacobian evaluation is reused multiple times during the corrective iteration process to update the solution (similar to Newton’s method) [80].

Our objective in this section is to examine if the use of full state vector

($\phi = \{Y, T\}$) with proper projection affects the accuracy of the final solution with a solver based on Rosenbrock method. For this purpose, we choose RODAS [129], which is an embedded Rosenbrock method of order 4(3) [the order reduces from 4 to 3 for stiff problems] with adaptive time stepping, implemented in Fortran. A constant pressure 0D auto-ignition case of heptane-oxygen mixture with same initial condition as in Sec. 2.8.4 is chosen as the validation case. Similar to previous analyses, we choose R^ϕ as the reference case. The error at each time step in evaluation of mass fractions for different projections compared to R^ϕ is evaluated using Eq. 2.36. The maximum and mean errors over all time steps are tabulated in Table 2.7. Both absolute and relative tolerances for ODE integration are set to 10^{-15} . The time stepping and number of time steps for each simulation are chosen to be 10^{-7} s and 10,000, respectively.

Table 2.7: Comparison of integration statistics and error norms between different choice of projections and R^ϕ in a 0D auto-ignition simulation of heptane-oxygen mixture using RODAS. NST and NJE refer to number of steps and number of Jacobian evaluations, respectively. NST and NJE are exactly same as RODAS evaluates the exact Jacobian at each sub-step.

Parameters	R^ϕ	P^N	P^\perp	P^S
NST	77,044	76,912	77,423	76,911
NJE	77,044	76,912	77,423	76,911
ϵ_{mean}	0 (Ref)	1.87×10^{-14}	1.74×10^{-9}	1.78×10^{-14}
ϵ_{max}	0 (Ref)	4.05×10^{-11}	4.93×10^{-6}	3.56×10^{-11}

There are two important observations from Table 2.7. First, the error norm between R^ϕ and different projections, except P^\perp , are within round-off errors. The maximum error goes up to $O(10^{-11})$ near the ignition because of the explosive mode (the real positive eigenvalue is maximum at the ignition point) involved

with an auto-ignition model as explained before. A relatively higher errors for P^\perp are also observed previously, which can be attributed to higher number of negative mass fractions due to the nature of this projection. The overall errors show that using full mass fraction space as state vector with proper projection does not affect the solution accuracy even with a linear implicit method for ODE integration, such as RODAS.

Secondly, the same errors are tabulated in Table 2.1 while using DVODE. Comparing those errors with Table 2.7, it is evident that the error norms over all time steps are significantly smaller in case of using RODAS than DVODE. Similar observations were also made in atmospheric chemistry problems [30]. This observation is explained in 2.10.3 by identifying the fundamental differences in the method of integration between the two solvers.

2.9 Conclusions

In the present work, a unifying analytical framework to formulate Jacobian matrices has been developed, which can handle a state vector comprising all species mass fractions and temperature without violating the constraint that the sum of mass fractions must be unity. To keep the mass fraction vector evaluation at all times during any reactive flow simulation in the realizable simplex, defined by the constraint, we use a projection matrix directly in the analytical Jacobian evaluation. Three different projections have been investigated for this purpose. The

benefits and drawbacks of each option have been examined and compared in the context of three different configurations: an auto-ignition model, an auto-ignition with mixing model, and PaSR. In terms of the individual boundedness, the projection that normalizes the mass fractions exhibits better accuracy than other projections, whereas the orthogonal projection is the least accurate. The error norms of all the mass fractions between an existing method of dealing with consistent state vectors and each of the projections are evaluated for all three configurations. The error was found to be significant for an auto-ignition case, which is attributed to the inherent explosive nature of such model. However, for the other two configurations, the errors turn out to be of the order of round-off errors. The applicability of the projection approach to handle a full state vector is also verified in the context of CSP or CEMA analysis, where the spectral properties of the exact Jacobian matrix is important, and with a linear implicit solver to integrate the governing equations, which requires exact Jacobian at every sub-step to update the solution. In both cases, the analytical Jacobian with projection provided satisfactory results compared to the existing methods.

2.10 Appendix

2.10.1 Comparison of error propagation between idealized ignition and idealized ignition-mixing case

This section compares how a round-off error in the initial condition propagates over time between an auto-ignition model and an ignition-mixing case by using a very simplified model of an idealized ignition and ignition-mixing case. First, we consider the idealized ignition problem

$$\frac{dy}{dt} = \lambda y \quad \text{for } \lambda > 0 \text{ and } 0 < y(0) \ll 1, \quad (2.38)$$

where y can be thought of as the mass fraction of a species with an initial condition of $y(0)$ which itself is very small and λ corresponds to the positive eigenvalue or the explosive mode of an ignition problem. The solution of Eq. 2.38 at time t_1 will be

$$y(t_1) = y(0)e^{\lambda t_1}. \quad (2.39)$$

Now, we consider the same ODE as Eq. 2.38 where the initial condition is perturbed by ϵ ,

$$\frac{dz}{dt} = \lambda z \quad \text{for } z(0) = y(0) + \epsilon \text{ and } \epsilon > 0, \quad (2.40)$$

the solution of which at time t_1 will be

$$z(t_1) = [y(0) + \epsilon]e^{\lambda t_1}. \quad (2.41)$$

Thus, after time t_1 , an initial round-off error of ϵ will be

$$E_{\text{ign}} = \frac{z(t_1) - y(t_1)}{y(t_1)} = \frac{\epsilon}{y(0)}. \quad (2.42)$$

Next, we consider the idealized mixing-ignition problem with \hat{y} as the mass fraction

$$\frac{d\hat{y}}{dt} = 1 + \lambda\hat{y} \quad \text{for } \lambda > 0 \text{ and } 0 < \hat{y}(0) \ll 1. \quad (2.43)$$

The solution of this equation at time t_1 is

$$\hat{y}(t_1) = \left(\hat{y}(0) + \frac{1}{\lambda} \right) e^{\lambda t_1} - \frac{1}{\lambda}. \quad (2.44)$$

Similar to the previous exercise, we consider \hat{z} to be the solution of 2.43 with an initial condition of $\hat{z}(0) = \hat{y}(0) + \epsilon$. At time t_1 , \hat{z} will be

$$\hat{z}(t_1) = \left(\hat{y}(0) + \epsilon + \frac{1}{\lambda} \right) e^{\lambda t_1} - \frac{1}{\lambda}. \quad (2.45)$$

Thus the error after time t_1 for this idealized mixing problem is

$$E_{\text{mix}} = \frac{\hat{z}(t_1) - \hat{y}(t_1)}{\hat{y}(t_1)} = \frac{\epsilon e^{\lambda t_1}}{\left(\hat{y}(0) + \frac{1}{\lambda} \right) e^{\lambda t_1} - \frac{1}{\lambda}}. \quad (2.46)$$

Note that, starting from a very small mass fraction ($y(0)$ or $\hat{y}(0)$), a 10^{10} increase in unit time (e.g., a radical mass fraction increasing from 10^{-15} to 10^{-5}) corresponds to a $\lambda = \ln 10^{10} \approx 23$ (using Eq. 2.39). With this approximate value of λ , $y(0) \ll \frac{1}{\lambda}$ and $e^\lambda \gg 1$, thus we have (from Eq. 2.46),

$$E_{\text{mix}} \approx \lambda \epsilon \quad (2.47)$$

It can be observed that the error in case of an idealized ignition (Eq. 2.42) tends to infinity as the initial condition $y(0)$ tends to zero. However, for an idealized mixing case, the error (Eq. 2.47) is always of the order of round-off error. Hence, an initial small error, ϵ , propagates more rapidly in auto-ignition case (Fig. 2.3) compared to the ignition-mixing case (Fig. 2.4).

2.10.2 Auto-ignition with quadruple precision

The auto-ignition model is investigated for heptane-oxygen mixture with different tolerances and precision of ODE integration in this section. The objective is to show that a rapid error increase in case of auto-ignition is not a consequence of the projection method itself, rather due to the inherent sensitive nature of the auto-ignition model. Three different simulations were run for all the projections with: (1) 10^{-15} tolerance and double precision (same as Table 2.1), (2) 10^{-15} tolerance and quadruple precision, and (3) 10^{-20} tolerance and quadruple precision. The tolerances mentioned for these three cases correspond to both absolute and relative values. The error between the projection case and R^ϕ is evaluated using Eq. 2.36 and tabulated in Table 2.8.

The result shows that changing the precision from double to quadruple keeping tolerances constant at 10^{-15} does not make any difference in the error propagation and the norm of errors remains significantly high. On the other hand, a change in tolerance from 10^{-15} to 10^{-20} with quadruple precision significantly af-

Table 2.8: Comparison of error norms between different choice of projections and R^ϕ in a constant pressure 0D auto-ignition case of heptane-oxygen mixture with two different tolerances and with both double and quadruple precision.

Tolerance	Precision	Mean error (ϵ_{mean})			Maximum error (ϵ_{max})		
		P^n	P^\perp	P^S	P^n	P^\perp	P^S
10^{-15}	Double	6.81×10^{-9}	3.96×10^{-7}	6.81×10^{-9}	1.64×10^{-5}	9.31×10^{-4}	1.64×10^{-5}
10^{-15}	Quadruple	6.39×10^{-9}	5.74×10^{-9}	4.97×10^{-9}	1.76×10^{-5}	1.58×10^{-5}	1.36×10^{-5}
10^{-20}	Quadruple	2.11×10^{-13}	2.11×10^{-13}	2.11×10^{-13}	6.00×10^{-10}	6.00×10^{-10}	6.00×10^{-10}

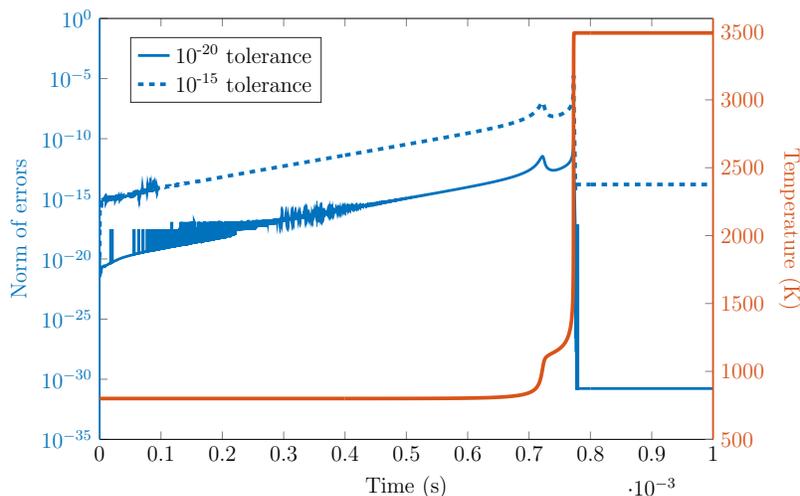


Figure 2.7: The norm of errors between R^ϕ and P^n at each time instant for a 0D auto-ignition case of heptane-oxygen mixture with quadruple precision and two different ODE tolerances (both absolute and relative): 10^{-15} and 10^{-20} . The plot also depicts the evolution of temperature during ignition.

ffects the error norm as the mean drops from 10^{-9} to 10^{-13} . Figure 2.7 shows the error propagation for the case (2) and case (3), both of them having quadruple precision. For case (3), the errors starts to increase from a much smaller value compared to case (2), thus the overall norm also remains smaller. It should be noted that the norm of errors depends largely on the initial error or the perturbation ϵ as shown in 2.10.1 and from Fig. 2.7, it is evident that the initial error is

dictated strictly by the tolerance. Consequently, a stricter tolerance of 10^{-20} compared to 10^{-15} induces a smaller initial perturbation, and as a result, the overall error norms become smaller as well. We can conclude that the higher error observed in Table 2.1 is not an artifact of the projection, rather it is the initial perturbation based on tolerance, and the nature of the auto-ignition problem itself that always increases the error over time, especially while using DVODE as a solver.

2.10.3 Difference in DVODE and RODAS

The use of a full state vector with proper projection does not significantly affect the accuracy of the final solution of the governing ODEs with a solver based on Rosenbrock method, such as RODAS, as shown in Sec. 2.8.5. However, while comparing the errors for a 0D auto-ignition problem between DVODE (Table 2.1) and RODAS (Table 2.7), it is evident that the mean and max error norms over all time steps are significantly smaller in case of using RODAS than DVODE. This observation can be explained by looking into the inherent properties of these two different solvers. Following the convention in Sec. 2.8.5, a user specified time-step is referred as 'time-step' and the internal solver time levels within a time-step is referred as 'sub-step'.

DVODE is a variable order method, with a maximum order of 5, if BDF method is used [80]. In a BDF method, within each time-step, a very small initial sub-step is chosen (typically between 10^{-12} and 10^{-8} s for a stiff chemical mechanism)

based on user-specified tolerances. The solution at the end of the first sub-step is evaluated using a first-order method. As integration progresses, the sub-step size increases gradually along with the order of accuracy (if needed) while minimizing the overall computational time and also ensuring the stability until the end of time-step. The higher order of accuracy is achieved gradually by using the solution of previous sub-steps. For instance, to achieve an order of q , the solver needs at least q previous sub-steps, thus the initial few sub-steps are always lower order accurate and as number of sub-steps increases, the solver can increase its accuracy by using more available sub-steps. As a result of such small sub-step and lower order accurate solution at the beginning of any time-step, the number of total sub-steps required for DVODE (Table 2.1) is also higher compared to RODAS (Table 2.7). In most of the reactive flow problems, the solver needs to reinitialize at the beginning of each chemical time-step, which essentially means that at each time-step the solver will start again from a very small sub-step and first order accuracy, then rebuild to achieve a higher order accurate solution at the end of time-step. This repeated stopping and reinitialization along with gradual increase in accuracy starting from a first order accurate solution results in a large number of floating-point operations, consequently, the error continuously accumulates over time [130]. Figure 2.8 shows the order of accuracy of the solution at the end of each time-step while using DVODE for the auto-ignition case. At times significantly earlier than the ignition point, the solution is only second order accurate. The method accuracy only goes to 4 or 5 in between the beginning of first ignition and the second ignition point. As a result, the error will definitely

be higher toward the beginning of the solution, which will propagate over time due to the sensitive nature of an auto-ignition model, as observed before.

Unlike DVODE, RODAS is a method with 4th order accuracy, while the order reduces to 3 for stiff problems. Although RODAS, similar to DVODE, starts from a very small sub-step within a time-step, the solution at each sub-step does not depend on previous sub-steps. It evaluates the Jacobian and updates the solution using six stages to achieve at least a 3rd order accurate solution at each sub-step. As a result, the solution is significantly more accurate than DVODE even in the first sub-step and does not accumulate over time as much as DVODE. Although RODAS provides a more accurate solution, it is less efficient due to the use of the exact Jacobian at each sub-step, which makes RODAS more expensive compared to multi-step BDF methods, like DVODE, which save and reuse the Jacobian for multiple sub-steps [128]. A significant difference in NJE (Table 2.1 and 2.7) is also observed for the auto-ignition case as well.

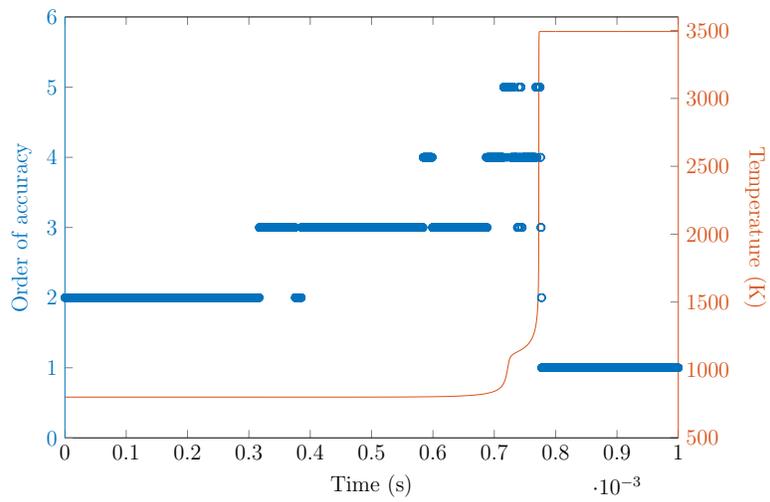


Figure 2.8: The order of accuracy of the solution at the end of each user specified time step in DVODE for an auto-ignition case of heptane-oxygen mixture.

CHAPTER 3

AN ANALYTICAL JACOBIAN GENERATOR FOR REDUCED CHEMICAL KINETIC MODELS INVOLVING QUASI-STEADY-STATE ASSUMPTIONS

3.1 Abstract

Recognizing the role of chemistry in the development of novel, more efficient engine technologies, numerical simulations of practical combustion systems are employing increasingly complex chemical kinetic models. A major challenge in such simulations is the cost of evaluating the chemical source terms, which often dominate the overall CPU cost, especially when using implicit numerical methods. In this case, a significant contributor to the CPU cost of chemical source term is the evaluation of the Jacobian matrix by the stiff ODE solver employed. This contribution can, however, be significantly decreased if an analytical formulation is explicitly provided to the solver. While automatic analytical Jacobian code generators have been developed in the literature, in this work, we extend the concept to handle the case of reduced kinetic models that involve quasi-steady state assumptions. The algorithm includes smart ordering of the quasi-steady-state variables to minimize the number of operations required to invert the QSS algebraic system, and the automatic generation of the routines required to evaluate the analytical Jacobian corresponding to the non-QSS species. The algorithm is validated and its performance assessed using a reduced kinetic model for n-dodecane oxidation in the context of partially stirred reactor simulations.

3.2 Introduction

Detailed numerical simulations of complex combustion systems, such as I.C. engines, require accurate and comprehensive chemical kinetic models. The size and complexity of these models have increased by orders of magnitudes in the last few decades [82]. Very recently developed reaction mechanisms, such as oxidation for surrogates of bio-diesel [131, 132], gasoline [133], alkyl esters [134], hydrocarbons from n-octane to n-hexadecane [135], contain more than 1000 species and 10000 reactions. A detailed mechanism developed for 2-methylalkanes, developed by Sarathy et. al. [136] contains almost 7200 species and 31400 reactions. The computational requirements for such large reaction mechanisms are very high as the number of species and reactions affect the memory requirement and CPU cost. This problem is further exacerbated by the large disparity that exists in the species and reaction timescales, which introduces stiffness into the system of ODEs and that controls the performance of time integration algorithm as well [82]. As a result, detailed modeling using a complex kinetic mechanism is still a daunting task even for 0D simulations such as auto ignition and perfectly stirred reactors and almost unaffordable for 1D simulations [137].

A wide range of techniques aiming at minimizing the computational demand have been developed and successfully applied in past literature, nicely reviewed by Law [138] and Lu and Law [82]. The most important categories of reducing the size and complexity of detailed mechanism are: (1) skeletal reduction which essentially removes the unimportant species and reactions by either Di-

rect Relation Graph (DRG) [49] or DRG with error propagation (DRGEP) [51] or Simulation Error Minimization Connectivity Method (SEM-CM) [99], (2) dimension or time-scale reduction methods which reduces the chemical stiffness of the system, e.g., rate-controlled constrained equilibrium (RCCE) [65], computational singular perturbation (CSP) [62], intrinsic low-dimensional manifolds (ILDm) [66], trajectory-generated low-dimensional manifolds (TGLDM) [122], (3) chemical lumping [54, 123]. In another category of reducing computational cost, Pope [68] developed *in situ* adaptive tabulation (ISAT) method where the expensive terms are tabulated for later use and a variable is evaluated from the tabulated values by interpolation if the variable is in the same mapping space. Later on, Hiremath and Pope [139] combined ISAT with dimension reduction methods and reported significant reduction in simulation time compared to only using ISAT.

The focus of this work is on the introduction of quasi steady state (QSS) assumptions, which is a time-scale reduction technique. The most important advantage of invoking QSS assumption is the conversion of the differential equations corresponding to a carefully selected set of QSS species into computationally more efficient algebraic expressions, thus reducing the number of ODEs that need to be transported, and typically reducing significantly the stiffness of the remaining equations [61, 140]. The introduction of QSS species is very simple and has been used widely in last few decades [141–145]. Very recently, Felden et. al. [146] used the QSS assumption in large scale LES of turbulent spray simulation.

Apart from chemistry reduction and other techniques mentioned above, differ-

ent mathematical techniques were also developed focusing on algorithm speed-up, e.g., more efficient solvers to integrate the governing ordinary differential equations. Due to the inherent stiffness of the ODE system, chemical kinetic models require solvers using backward differentiation methods which are developed based on Newton's method that requires repeated Jacobian matrix evaluations [78, 80, 81]. The computational cost of evaluating an approximate Jacobian scales quadratically with number of species (n_s^2) as the reaction rates need to be re-evaluated for each perturbed species concentration, whereas the factorization of the matrix scales as n_s^3 , where n_s is the number of species [75, 87]. For a medium to large size mechanism, Jacobian matrix factorization is the most expensive step [75]. On the other hand, the cost of using analytical Jacobian is linearly dependent on the number of species or reactions [82].

Motivated by the potential reduction in computational cost, a few analytical Jacobian generator packages have been developed in last few years. Safta et al. [74] developed TChem toolkit, Youssefi [83] and Bisetti [75] also developed their own analytical Jacobian matrix source code to reduce both computational cost and numerical error. Bisetti [75] used adoptive symbolic analysis tools which provide mechanism specific routines that do not typically use sparse matrix algebra during Jacobian formation. Different strategies were adopted to reduce the number of calls to ODE system function by providing a sparsity pattern. For instance, the DAEPACK automatic differentiation library [84–86] provides analytical Jacobian calculations along with sparsity pattern information, which reduces computational cost significantly when coupled to sparse ODE solvers [80]. Perini et

al [87] used analytical Jacobian for constant volume reactor which helps the matrix to be more sparse as the off-diagonal elements would only be non-zero because of direct species interactions and fall off or pressure dependent reactions. Perini et al [87] reported a 80% faster result when the analytical Jacobian was used combined with tabulation of temperature dependent properties. Unfortunately, most of the combustion CFD tools are built on a constant pressure assumption, thus reducing sparsity of the Jacobian matrix. Recently GPU or hybrid CPU/GPU [89] have been used rather than pure CPU to speedup the computational time which is another category of computational technique that maximizes the efficiency of algorithm. Dijkmans et al. [76] took a GPU based analytical Jacobian approach to accelerate the integration algorithm even further. Niemeyer et al. [77] developed pyJac, an open source analytical Jacobian tool optimized for SIMD processors.

To our knowledge, no analytical Jacobian formulation exists in literature that can also exploit the advantage of invoking QSS species in the system. In this work, the concept of analytical Jacobian formulation is extended further for reduced kinetic mechanisms involving QSS species. This package calculates the QSS concentrations using their corresponding algebraic expressions and automatically develops the analytical Jacobian matrix for non-QSS species. The organization of the paper is as follows. The next section talks about the basic theory and governing equations for chemical kinetics. Section 3.4 discusses the formulation of the general analytical Jacobian without QSS assumption. Then section 3.5 describes the modifications required for the QSS species implementation in analytical Jacobian formulation. The accuracy and performance of the current algorithm are

evaluated (section 3.6) in the context of a partially stirred reactor (PaSR) with non-premixed n-dodecane/air combustion.

3.3 Theory

The present study considers the time evolution of reactive homogeneous gaseous mixtures in a constant pressure system whose solution is achieved through the integration of a system of ODE using an analytical Jacobian matrix with QSS assumptions. This section mostly describes the basic theory and governing equations. This initial value problem (IVP) is described by the independent variables: *species mass fractions* (Y_i) and *temperature* (T).

$$\phi = \{Y_1, Y_2, \dots, Y_{N_s}, T\}^T \quad (3.1)$$

where N_s is the total number of species.

3.3.1 Thermodynamic Relations

Assuming ideal gas law, the mixture density (ρ) can be expressed in terms of pressure (p) and temperature (T)

$$\rho = \frac{pW}{\mathcal{R}T} = \frac{p}{\mathcal{R}T} \left(\sum_{i=1}^{N_s} \frac{Y_i}{W_i} \right)^{-1} \quad (3.2)$$

where W is the mixture molecular weight, W_i is the molecular weight of i -th species and \mathcal{R} is the universal gas constant. The analytical Jacobian can be simplified by formulating in terms of molar concentrations rather than species mass fractions directly, as the reaction rates depend directly on molar concentrations (will be explained later), thus it is important to relate species mass concentration (ρ_i) and species mass fraction (Y_i) with species molar concentrations (C_i) by

$$C_i = \frac{\rho_i}{W_i} = \rho \frac{Y_i}{W_i} \quad (3.3)$$

The thermodynamic properties of species i in molar units are calculated using standard NASA polynomials [147].

$$\frac{C_{p,i}^0}{\mathcal{R}} = a_{1,i} + a_{2,i}T + a_{3,i}T^2 + a_{4,i}T^3 + a_{5,i}T^4 \quad (3.4)$$

$$\frac{H_i^0}{\mathcal{R}T} = a_{1,i} + \frac{a_{2,i}}{2}T + \frac{a_{3,i}}{3}T^2 + \frac{a_{4,i}}{4}T^3 + \frac{a_{5,i}}{5}T^4 + \frac{a_{6,i}}{T} \quad (3.5)$$

$$\frac{S_i^0}{\mathcal{R}} = a_{1,i} \ln T + a_{2,i}T + \frac{a_{3,i}}{2}T^2 + \frac{a_{4,i}}{3}T^3 + \frac{a_{5,i}}{4}T^4 + a_{7,i} \quad (3.6)$$

where $C_{p,i}$, H_i and S_i are the constant pressure specific heat, enthalpy and entropy of i -th species in molar units, respectively. The superscript ⁰ indicates the standard state, i.e., one atmospheric pressure. These thermodynamic property values can be converted in to a mass based specific heat ($c_{p,i}$) and enthalpy (h_i) by using

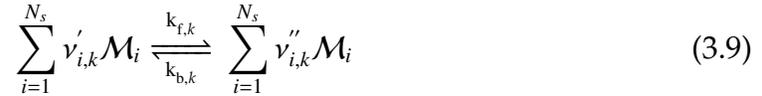
$$c_{p,i} = \frac{C_{p,i}}{W_i} \quad \text{and} \quad h_i = \frac{H_i}{W_i} \quad (3.7)$$

The mixture averaged specific heat can be written as

$$c_p = \sum_{i=1}^{N_s} Y_i c_{p,i} \quad (3.8)$$

3.3.2 Description of chemical kinetics

The reaction mechanism consists of total N_R number of reactions involving N_s species. The k -th reaction can be represented as



where $\nu'_{i,k}$ and $\nu''_{i,k}$ are the reactant and product stoichiometric coefficients of species i in reaction k , \mathcal{M}_i is the name of species i , $k_{f,k}$ and $k_{b,k}$ are the forward and reverse reaction rate coefficients of reaction k . From law of mass action,

$$\dot{C}_i = \frac{\partial C_i}{\partial t} = \sum_{k=1}^{N_R} (\nu''_{i,k} - \nu'_{i,k}) \omega_k = \sum_{k=1}^{N_R} \nu_{i,k} \omega_k \quad (3.10)$$

where $\nu_{i,k} = \nu''_{i,k} - \nu'_{i,k}$ and ω_k is the reaction rate of k -th reaction. ω_k is dependent on temperature and species concentrations directly. The nature of ω_k may differ depending on the type of reactions or the specific interaction of the involved reactants. The details of different formulations of ω_k for different kind of reactions are discussed below.

Simple elementary reactions

For a simple elementary reaction,

$$\omega_k = k_{f,k} \prod_{i=1}^{N_s} C_i^{v'_{i,k}} - k_{b,k} \prod_{i=1}^{N_s} C_i^{v''_{i,k}} \quad (3.11)$$

where the forward and reverse rate coefficients are temperature dependent. The forward rate coefficient ($k_{f,k}$) can be evaluated using Arrhenius expression:

$$k_{f,k} = A_k T^{\beta_k} \exp\left(-\frac{E_{a,k}}{RT}\right) \quad (3.12)$$

where A_k , β_k and $E_{a,k}$ are the pre-exponential/collision frequency factor, temperature coefficient and activation energy of reaction k , respectively. The evaluation of Eq. 3.12 is expensive because of the exponential functions and a large number of reactions in a reaction mechanism. However, Eq. 3.12 can be efficiently modified in order to reduce the computational cost by taking advantage of zero activation energy or zero temperature coefficient for a large number of reactions [77, 82]. For a case when $E_{a,k}$ and β_k are not trivial, Eq. 3.12 can be re-written as

$$k_{f,k} = \exp\left(\ln A_k + \beta_k \ln T - \frac{E_{a,k}}{RT}\right) \quad (3.13)$$

where $\ln T$ and $1/RT$ are pre-evaluated in rate evaluation subroutine to make it even more computationally efficient.

The pre-exponential factor, temperature coefficient and activation energy for backward reactions are calculated by data fitting of forward reaction Arrhenius coefficients and assuming equilibrium condition in a wide temperature

range [148]. Then the backward reaction rate coefficient are also evaluated using Eq. 3.12 or 3.13 and treated as separate forward reactions. Consequently, Eq 3.11 can be simplified as

$$\omega_k = k_{f,k} \prod_{i=1}^{N_s} C_i^{v'_{i,k}} \quad (3.14)$$

where k counts for all forward and backward reactions.

Third body reactions

For a third body reaction (TB), the reaction rate (ω_k) is modified by an effective concentration of the third body.

$$\omega_k^{TB} = C_{M,k} \omega_k \quad (3.15)$$

where

$$C_{M,k} = \sum_{i=1}^{N_s} \alpha_{i,k} C_i \quad (3.16)$$

and ω_k^{TB} is the reaction rate of k -th reaction after considering the effective third body concentration ($C_{M,k}$), $\alpha_{i,k}$ is the third-body efficiency of species i in k -th reaction, C_i is the concentration of species i . ω_k in Eq. 3.15 is evaluated from Eq. 3.14. If all the species contribute equally in third body concentration, then $\alpha_{i,k} = 1$ for all species.

Fall-off reactions

The fall-off reactions (F) are pressure dependent, i.e., unlike simple reactions, fall-off reaction rate shows different behavior at high and low pressure limits. The forward reaction rate coefficient in high and low pressure limits are given by k_{f0} and $k_{f\infty}$ respectively for a fall-off reaction. So, the effective reaction rate coefficient ($k_{f,k}$) will be

$$k_{f,k}^F = k_{f\infty,k} F_k \quad (3.17)$$

and effective reaction rate (ω_k^F) is

$$\omega_k^F = \omega_k F_k \quad (3.18)$$

where F_k is the weighing factor. ω_k in Eq. 3.18 is evaluated using Eq. 3.14 with high pressure limit rate coefficient ($k_{f\infty,k}$). The weighing factor (F_k) depends on the reduced pressure corresponding to the k -th reaction ($P_{r,k}$), which is given by,

$$P_{r,k} = \frac{k_{f0,k}}{k_{f\infty,k}} C_{M,k} \quad (3.19)$$

For a simple pressure-dependent reaction, like Lindemann (Lind) [149], the weighing factor is

$$F_k^{\text{Lind}} = \frac{P_{r,k}}{1 + P_{r,k}} \quad (3.20)$$

For Troe's kinetic approach (Troe) [150], the weighing factor is more complicated.

$$F_k^{\text{Troe}} = \frac{P_{r,k}}{1 + P_{r,k}} F_{\text{cent},k}^{(1+f^2)^{-1}} \quad (3.21)$$

where the centering factor, $F_{\text{cent},k}$ is given by

$$F_{\text{cent},k} = (1 - a_k) \exp\left(-\frac{T}{T_{3,k}}\right) + a_k \exp\left(-\frac{T}{T_{1,k}}\right) + \exp\left(-\frac{T_{2,k}}{T}\right) \quad (3.22)$$

and

$$f = \frac{J_k}{n_k - d_k J_k} \quad (3.23)$$

where

$$J_k = \log_{10} P_{r,k} + c_k \quad (3.24)$$

$$c_k = -0.40 - 0.67 \log_{10} F_{\text{cent},k} \quad (3.25)$$

$$n_k = 0.75 - 1.27 \log_{10} F_{\text{cent},k} \quad (3.26)$$

$$d_k = 0.14 \quad (3.27)$$

and a_k , $T_{1,k}$, $T_{2,k}$ and $T_{3,k}$ are specified parameters for each fall-off reaction. The third term of Eq. 3.22 is optional and can be omitted if $T_{2,k}$ is not specified.

3.4 Analytical Jacobian Formulation

This section presents the details of a general analytical Jacobian formulation (without QSS approximation). The analytical Jacobian, \mathcal{J} is filled with partial deriva-

tives, such as

$$\mathcal{J}_{i,j} = \frac{\partial f_i}{\partial \phi_j} \quad (3.28)$$

where ϕ corresponds to the independent variables, i.e., species mass fractions and temperature (Eq. 3.1) and f denotes the system of ODE governed by the rate of change of mass fractions and temperature.

$$f = \frac{d\phi}{dt} = \left\{ \frac{dY_1}{dt}, \frac{dY_2}{dt}, \dots, \frac{dY_{N_s}}{dt}, \frac{dT}{dt} \right\}^T \quad (3.29)$$

where

$$\dot{Y}_i = \frac{dY_i}{dt} = \frac{W_i}{\rho} \dot{C}_i \quad i = 1 \text{ to } N_s \quad (3.30)$$

and

$$\dot{T} = \frac{dT}{dt} = \frac{-1}{\rho c_p} \sum_{i=1}^{N_s} h_i W_i \dot{C}_i = \frac{-1}{c_p} \sum_{i=1}^{N_s} h_i \dot{Y}_i \quad (3.31)$$

c_p and h_i are calculated using Eq. 3.7 and 3.8. Equations 3.30 and 3.31 define the initial value problem consisting total $N_s + 1$ unknowns with the initial condition, $\phi(t = 0) = \phi_0$. Thus the dimension of Jacobian matrix for a reaction mechanism with N_s species is $(N_s + 1) \times (N_s + 1)$ as shown in Fig. 3.1.

The Jacobian matrix consists of four different kind of partial differential equations (Fig. 3.1). The derivation of the analytical expression of each of these derivatives are discussed below.

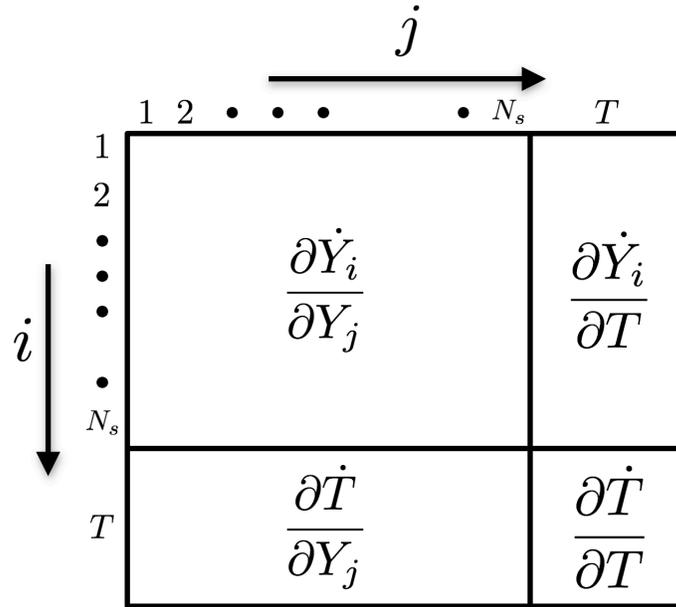


Figure 3.1: The Jacobian matrix structure

3.4.1 Derivatives of mass fraction rate change w.r.t mass fractions

The mass fraction derivatives, $\frac{\partial \dot{Y}_i}{\partial Y_j}$, where both i and j vary from 1 to N_s , can be simplified by rewriting the expressions in terms of concentration derivatives as the rate of change of concentration (\dot{C}_i) has direct relation with reaction rates (Eq. 3.10).

$$\begin{aligned}
\mathcal{J}_{i,j} &= \frac{\partial \dot{Y}_i}{\partial Y_j} = \frac{\partial}{\partial Y_j} \left(\frac{W_i}{\rho} \dot{C}_i \right) \quad [\text{using Eq. 3.30}] \\
&= \frac{W_i}{\rho} \frac{\partial \dot{C}_i}{\partial Y_j} - \frac{W_i \dot{C}_i}{\rho^2} \frac{\partial \rho}{\partial Y_j} \\
&= \frac{W_i}{\rho} \frac{\partial \dot{C}_i}{\partial Y_j} - \frac{\dot{Y}_i}{\rho} \frac{\partial \rho}{\partial Y_j}
\end{aligned} \tag{3.32}$$

where

$$\frac{\partial \dot{C}_i}{\partial Y_j} = \sum_{l=1}^{N_s} \frac{\partial \dot{C}_i}{\partial C_l} \frac{\partial C_l}{\partial Y_j} \tag{3.33}$$

where l varies from 1 to N_s . The partial derivative of density w.r.t mass fraction can be derived using Eq. 3.2.

$$\frac{\partial \rho}{\partial Y_j} = \frac{p}{\mathcal{R}T} \frac{\partial}{\partial Y_j} \left[\left(\sum_{j=1}^{N_s} \frac{Y_j}{W_j} \right)^{-1} \right] = -\rho \frac{W}{W_j} \tag{3.34}$$

The second term in the right hand side of Eq. 3.33 can be expressed as

$$\frac{\partial C_l}{\partial Y_j} = \frac{\partial}{\partial Y_j} \left(\rho \frac{Y_l}{W_l} \right) = \frac{\rho}{W_j} \delta_{lj} + \frac{Y_l}{W_l} \frac{\partial \rho}{\partial Y_j} \tag{3.35}$$

After plugging Eq. 3.34 in Eq. 3.35,

$$\frac{\partial C_l}{\partial Y_j} = -C_l \frac{W}{W_j} + \delta_{lj} \frac{\rho}{W_j} \tag{3.36}$$

The partial derivative of concentration rate change w.r.t. concentration (first term of Eq. 3.33) is given by (using Eq. 3.10),

$$\frac{\partial \dot{C}_i}{\partial C_l} = \frac{\partial}{\partial C_l} \left(\sum_{k=1}^{N_R} \nu_{i,k} \omega_k \right) = \sum_{k=1}^{N_R} \nu_{i,k} \frac{\partial \omega_k}{\partial C_l} \tag{3.37}$$

The reaction rate, ω_k in Eq. 3.37 denotes the effective reaction rates of different types of reactions. For third body enhanced and fall-off reactions, ω_k are ω_k^{TB} (Eq. 3.15) and ω_k^F (Eq. 3.18) respectively. For a **simple elementary reaction** (using Eq. 3.14),

$$\frac{\partial \omega_k}{\partial C_l} = k_{f,k} \nu'_{l,k} C_l^{\nu'_{l,k}-1} \prod_{i \neq l} C_i^{\nu'_{i,k}} \quad (3.38)$$

For a **third body enhanced reaction** (using Eq. 3.15),

$$\begin{aligned} \frac{\partial \omega_k^{TB}}{\partial C_l} &= C_{M,k} \frac{\partial \omega_k}{\partial C_l} + \omega_k \frac{\partial C_{M,k}}{\partial C_l} \\ &= C_{M,k} \frac{\partial \omega_k}{\partial C_l} + \omega_k \alpha_{l,k} \end{aligned} \quad (3.39)$$

where ω_k and $\frac{\partial \omega_k}{\partial C_l}$ are calculated using Eq. 3.14 and 3.38 respectively.

For a pressure-dependent **Fall-off reaction** (using Eq. 3.18),

$$\frac{\partial \omega_k^F}{\partial C_l} = F_k \frac{\partial \omega_k}{\partial C_l} + \omega_k \frac{\partial F_k}{\partial C_l} \quad (3.40)$$

For a Lindemann type reaction,

$$\begin{aligned} \frac{\partial \ln F_k^{\text{Lind}}}{\partial C_l} &= \frac{1 + P_{r,k}}{P_{r,k}} \frac{\partial}{\partial C_l} \left(\frac{P_{r,k}}{1 + P_{r,k}} \right) \\ &= \frac{1}{1 + P_{r,k}} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial C_l} \right) \end{aligned} \quad (3.41)$$

For a Troe type reaction,

$$\begin{aligned} \frac{\partial \ln F_k^{\text{Troe}}}{\partial C_l} &= \frac{\partial}{\partial C_l} \left[\ln \frac{P_{r,k}}{1 + P_{r,k}} + \frac{1}{1 + f^2} \ln F_{\text{cent},k} \right] \\ &= \frac{1}{1 + P_{r,k}} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial C_l} \right) - \frac{\ln F_{\text{cent},k}}{(1 + f^2)^2} \frac{\partial f}{\partial C_l} \end{aligned} \quad (3.42)$$

where

$$\frac{\partial f}{\partial C_l} = \frac{\partial}{\partial C_l} \left[\frac{J_k}{n_k - d_k J_k} \right] = \frac{n_k}{(n_k - d_k J_k)^2} \frac{\partial J_k}{\partial C_l} = \frac{n_k}{(n_k - d_k J_k)^2} \frac{1}{\ln 10} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial C_l} \right) \quad (3.43)$$

Finally, the reduced pressure derivative is calculated by using Eq. 3.19

$$\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial C_l} = \frac{\alpha_{l,k}}{C_{M,k}} \quad (3.44)$$

3.4.2 Derivatives of temperature rate change w.r.t mass fractions

Starting from Eq. 3.31, the temperature rate change derivative w.r.t. mass fractions ($\frac{\partial \dot{T}}{\partial Y_j}$) where j varies from 1 to N_s , is given by

$$\begin{aligned} \mathcal{J}_{N_s+1,j} &= \frac{\partial \dot{T}}{\partial Y_j} = \frac{\partial}{\partial Y_j} \left(-\frac{1}{\rho c_p} \sum_{i=1}^{N_s} h_i W_i \dot{C}_i \right) \\ &= -\sum_{i=1}^{N_s} \left[\frac{h_i}{c_p} \frac{\partial}{\partial Y_j} \left(\frac{W_i \dot{C}_i}{\rho} \right) - \frac{\dot{T}}{c_p} \frac{\partial c_p}{\partial Y_j} + \frac{W_i \dot{C}_i}{\rho c_p} \frac{\partial h_i}{\partial Y_j} \right] \end{aligned} \quad (3.45)$$

where the derivative of enthalpy w.r.t. mass fraction is zero.

$$\frac{\partial h_i}{\partial Y_j} = 0 \quad (3.46)$$

So, finally Eq. 3.45 can be rewritten as,

$$\frac{\partial \dot{T}}{\partial Y_j} = \frac{-1}{c_p} \left(c_{p,j} \dot{T} + \sum_{i=1}^{N_s} h_i \frac{\partial \dot{Y}_i}{\partial Y_j} \right) \quad (3.47)$$

where the evaluation of $\frac{\partial \dot{Y}_i}{\partial Y_j}$ is discussed in Section 3.4.1.

3.4.3 Derivatives of mass fraction rate change w.r.t temperature

The mass fraction derivative w.r.t temperature ($\frac{\partial \dot{Y}_i}{\partial T}$), where i varies from 1 to N_s , can be evaluated using Eq. 3.30.

$$\begin{aligned} \mathcal{J}_{i,N_s+1} &= \frac{\partial \dot{Y}_i}{\partial T} = \frac{\partial}{\partial T} \left(\frac{W_i \dot{C}_i}{\rho} \right) \\ &= \frac{W_i}{\rho} \frac{\partial \dot{C}_i}{\partial T} - \frac{W_i \dot{C}_i}{\rho^2} \frac{\partial \rho}{\partial T} \\ &= \frac{W_i}{\rho} \frac{\partial \dot{C}_i}{\partial T} - \frac{\dot{Y}_i}{\rho} \frac{\partial \rho}{\partial T} \end{aligned} \quad (3.48)$$

where

$$\frac{\partial \rho}{\partial T} = \frac{\partial}{\partial T} \left(\frac{pW}{\mathcal{R}T} \right) = -\frac{\rho}{T} \quad (3.49)$$

So, finally Eq. 3.48 can be simplified as

$$\mathcal{J}_{i,N_s+1} = \frac{\partial \dot{Y}_i}{\partial T} = \frac{W_i}{\rho} \frac{\partial \dot{C}_i}{\partial T} + \frac{\dot{Y}_i}{T} \quad (3.50)$$

The partial derivative of concentration w.r.t temperature is evaluated by (using Eq. 3.10)

$$\frac{\partial \dot{C}_i}{\partial T} = \frac{\partial}{\partial T} \left(\sum_{k=1}^{N_R} \nu_{i,k} \omega_k \right) = \sum_{k=1}^{N_R} \nu_{i,k} \frac{\partial \omega_k}{\partial T} \quad (3.51)$$

The reaction rate, ω_k in Eq. 3.51 denotes the effective reaction rates of different types of reactions. For third body enhanced and fall-off reactions, ω_k are ω_k^{TB} (Eq. 3.15) and ω_k^F (Eq. 3.18) respectively. For a **simple elementary reaction** (us-

ing Eq. 3.14),

$$\frac{\partial \omega_k}{\partial T} = \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} + k_{f,k} \sum_{j=1}^{N_s} \nu'_{j,k} C_j^{\nu'_{j,k}-1} \frac{\partial C_j}{\partial T} \prod_{\substack{i=1 \\ i \neq j}}^{N_s} C_i^{\nu'_{i,k}} \quad (3.52)$$

where

$$\frac{\partial C_j}{\partial T} = \frac{Y_j}{W_j} \frac{\partial \rho}{\partial T} = -\frac{C_j}{T} \quad (3.53)$$

Thus Eq. 3.52 can be simplified as,

$$\begin{aligned} \frac{\partial \omega_k}{\partial T} &= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} + k_{f,k} \sum_{j=1}^{N_s} \nu'_{j,k} C_j^{\nu'_{j,k}-1} \left(-\frac{C_j}{T}\right) \prod_{\substack{i=1 \\ i \neq j}}^{N_s} C_i^{\nu'_{i,k}} \\ &= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} - \left(k_{f,k} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}}\right) \frac{1}{T} \sum_{j=1}^{N_s} \nu'_{j,k} \\ &= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} - \frac{\omega_k}{T} \sum_{j=1}^{N_s} \nu'_{j,k} \end{aligned} \quad (3.54)$$

For a simple elementary reaction, partial derivative of rate coefficient can be calculated from the Arrhenius equation (Eq. 3.12).

$$\frac{\partial k_{f,k}}{\partial T} = \frac{k_{f,k}}{T} \left(\beta_k + \frac{E_{a,k}}{\mathcal{R}T} \right) \quad (3.55)$$

For a **third body enhanced reaction**, the derivatives are similar to simple elemen-

tary reactions with additional terms for third body concentrations.

$$\begin{aligned}
\frac{\partial \omega_k^{TB}}{\partial T} &= C_{M,k} \frac{\partial \omega_k}{\partial T} + \omega_k \frac{\partial C_{M,k}}{\partial T} \\
&= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} C_{M,k} - \frac{\omega_k}{T} C_{M,k} \sum_{j=1}^{N_s} \nu'_{j,k} + \omega_k \left(-\frac{C_{M,k}}{T} \right) \\
&= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_s} C_i^{\nu'_{i,k}} C_{M,k} - \frac{\omega_k^{TB}}{T} \left(1 + \sum_{j=1}^{N_s} \nu'_{j,k} \right)
\end{aligned} \tag{3.56}$$

where partial derivative of rate coefficient is calculated using Eq. 3.55.

For a pressure dependent **fall-off reaction**,

$$\frac{\partial \omega_k^F}{\partial T} = F_k \frac{\partial \omega_k}{\partial T} + \omega_k \frac{\partial F_k}{\partial T} \tag{3.57}$$

where ω_k and $\frac{\partial \omega_k}{\partial T}$ is calculated using Eq. 3.14 and 3.54 respectively.

For a Lindemann type reaction,

$$\begin{aligned}
\frac{\partial \ln F_k^{\text{Lind}}}{\partial T} &= \frac{1 + P_{r,k}}{P_{r,k}} \frac{\partial}{\partial T} \left(\frac{P_{r,k}}{1 + P_{r,k}} \right) \\
&= \frac{1}{1 + P_{r,k}} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial T} \right)
\end{aligned} \tag{3.58}$$

For a Troe type reaction,

$$\begin{aligned}
\frac{\partial \ln F_k^{\text{Troe}}}{\partial T} &= \frac{\partial}{\partial T} \left[\ln \frac{P_{r,k}}{1 + P_{r,k}} + \frac{1}{1 + f^2} \ln F_{\text{cent},k} \right] \\
&= \frac{1}{1 + P_{r,k}} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial T} \right) - \frac{\ln F_{\text{cent},k}}{(1 + f^2)^2} \frac{\partial f}{\partial T} + \frac{1}{1 + f^2} \frac{\partial \ln F_{\text{cent},k}}{\partial T}
\end{aligned} \tag{3.59}$$

The partial derivative of f in the second term of Eq. 3.59 is evaluated by

$$\frac{\partial f}{\partial T} = \frac{n_k}{(n_k - d_k J_k)^2} \frac{\partial J_k}{\partial T} - \frac{J_k}{(n_k - d_k J_k)^2} \frac{\partial n_k}{\partial T} \tag{3.60}$$

where

$$\frac{\partial J_k}{\partial T} = \frac{1}{\ln 10} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial T} \right) + \frac{\partial c_k}{\partial T} = \frac{1}{\ln 10} \left(\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial T} \right) - \frac{0.67}{\ln 10} \frac{\partial \ln F_{\text{cent},k}}{\partial T} \quad (3.61)$$

and

$$\frac{\partial n_k}{\partial T} = -\frac{1.27}{\ln 10} \frac{\partial \ln F_{\text{cent},k}}{\partial T} \quad (3.62)$$

Finally, the partial derivative of $\ln F_{\text{cent},k}$, required in Eq. 3.59, 3.61 and 3.62 is given by

$$\begin{aligned} \frac{\partial \ln F_{\text{cent},k}}{\partial T} &= \frac{1}{F_{\text{cent},k}} \frac{\partial F_{\text{cent},k}}{\partial T} \\ &= \frac{1}{F_{\text{cent},k}} \left[-\frac{1-a_k}{T_{3,k}} \exp\left(-\frac{T}{T_{3,k}}\right) - \frac{a_k}{T_{1,k}} \exp\left(-\frac{T}{T_{1,k}}\right) + \frac{T_{2,k}}{T^2} \exp\left(-\frac{T_{2,k}}{T}\right) \right] \end{aligned} \quad (3.63)$$

and partial derivative of reduced pressure is given by,

$$\frac{1}{P_{r,k}} \frac{\partial P_{r,k}}{\partial T} = \frac{1}{k_{f0,k}} \frac{\partial k_{f0,k}}{\partial T} - \frac{1}{k_{f\infty,k}} \frac{\partial k_{f\infty,k}}{\partial T} - \frac{1}{T} \quad (3.64)$$

where partial derivatives of $k_{f0,k}$ and $k_{f\infty,k}$ are given by Eq 3.55.

3.4.4 Derivatives of temperature rate change w.r.t temperature

The partial derivative of temperature w.r.t temperature can be derived from Eq. 3.31.

$$\begin{aligned}
 \frac{\partial \dot{T}}{\partial T} &= \frac{\partial}{\partial T} \left(-\frac{1}{c_p} \sum_{i=1}^{N_s} h_i \dot{Y}_i \right) \\
 &= -\sum_{i=1}^{N_s} \left[\frac{\dot{Y}_i}{c_p} \frac{\partial h_i}{\partial T} - \frac{h_i \dot{Y}_i}{c_p^2} \frac{\partial c_p}{\partial T} + \frac{h_i}{c_p} \frac{\partial \dot{Y}_i}{\partial T} \right] \\
 &= -\frac{1}{c_p} \left[\sum_{i=1}^{N_s} \dot{Y}_i c_{p,i} + \dot{T} \sum_{i=1}^{N_s} Y_i \frac{\partial c_{p,i}}{\partial T} + \sum_{i=1}^{N_s} h_i \frac{\partial \dot{Y}_i}{\partial T} \right] \tag{3.65}
 \end{aligned}$$

where partial derivative of c_p can be calculated using Eq. 3.4.

$$\frac{\partial c_{p,i}}{\partial T} = \frac{\mathcal{R}}{W_i} (a_{2,i} + 2a_{3,i}T + 3a_{4,i}T^2 + 4a_{5,i}T^3) \tag{3.66}$$

and $\frac{\partial \dot{Y}_i}{\partial T}$ evaluation is discussed in the previous section.

3.5 Quasi-Steady-State (QSS) species implementation

A species is considered to be quasi steady if it features a fast consumption time scale compared to its production time such that they balance each other resulting in a steady and low concentration after a transient period [61]. These QSS species are identified by the level of importance (LOI) criteria [151], which essentially estimates an error in concentration of a species if it is assumed to be quasi

steady. Introduction of such quasi steady species assumption converts their corresponding ODEs into computationally more effective algebraic expressions thus reducing the number of ODEs that have to be solved and typically reducing the stiffness of the remaining differential equations. However, the QSS species concentrations not only depend on non-QSS species, but also usually coupled with other QSS species (only linear coupling is considered). Numerically optimized, decoupled and explicit expressions for each QSS species have been achieved by reordering them properly before applying Gauss Pivoting [152].

The analytical Jacobian with QSS assumption deviates significantly from the formulation discussed in previous section. As the QSS species concentrations are evaluated using their corresponding algebraic expressions, the analytical Jacobian is required only for non-QSS species. So, the dimension of Jacobian matrix reduces to $(N_{nq} + 1) \times (N_{nq} + 1)$, where N_{nq} is the number of non-QSS species. For the same reason, a lot of equations developed in the previous section will also be modified accordingly. First of all, although the number of independent variables (Eq. 3.1) still remains $(N_s + 1)$, the rate of change of variables, that is required for analytical Jacobian, denoted by f in Eq. 3.29 reduces to

$$f = \frac{\partial \phi}{\partial t} = \left\{ \frac{\partial Y_1}{\partial t}, \frac{\partial Y_2}{\partial t}, \dots, \frac{\partial Y_{N_{nq}}}{\partial t}, \frac{\partial T}{\partial t} \right\}^T \quad (3.67)$$

where \dot{Y}_i (Eq. 3.30) and \dot{T} (Eq. 3.31) are modified as

$$\dot{Y}_i = \frac{\partial Y_i}{\partial t} = \frac{W_i}{\rho} \dot{C}_i \quad i = 1 \text{ to } N_{nq} \quad (3.68)$$

and

$$\dot{T} = \frac{\partial T}{\partial t} = \frac{-1}{\rho c_p} \sum_{i=1}^{N_{nq}} h_i W_i \dot{C}_i = \frac{-1}{c_p} \sum_{i=1}^{N_{nq}} h_i \dot{Y}_i \quad (3.69)$$

The contribution of QSS species in \dot{T} is zero, as for all QSS species, $\dot{Y} = 0$. Similar change, i.e., summing over N_{nq} number of species rather than N_s , is applicable in Eq. 3.33:

$$\frac{\partial \dot{C}_i}{\partial Y_j} = \sum_{l=1}^{N_{nq}} \frac{\partial \dot{C}_i}{\partial C_l} \frac{\partial C_l}{\partial Y_j} \quad (3.70)$$

and in Eq. 3.47:

$$\frac{\partial \dot{T}}{\partial Y_j} = \frac{-1}{c_p} \left(c_{p,j} \dot{T} + \sum_{i=1}^{N_{nq}} h_i \frac{\partial \dot{Y}_i}{\partial Y_j} \right) \quad (3.71)$$

and also in Eq. 3.65:

$$\frac{\partial \dot{T}}{\partial T} = -\frac{1}{c_p} \left[\sum_{i=1}^{N_{nq}} \dot{Y}_i c_{p,i} + \dot{T} \sum_{i=1}^{N_{nq}} Y_i \frac{\partial c_{p,i}}{\partial T} + \sum_{i=1}^{N_{nq}} h_i \frac{\partial \dot{Y}_i}{\partial T} \right] \quad (3.72)$$

Secondly, the QSS species are assumed not to act as third bodies, i.e., the third body efficiency, α_i for all QSS species is zero. This assumption is quite reasonable as the third body concentration has to be large enough to affect the reaction rates significantly, which is in direct contradiction with the fact that QSS concentrations are usually very small. This fact changes Eq. 3.16 and Eq. 3.39 in following two equations respectively.

$$C_{M,k} = \sum_{i=1}^{N_{nq}} \alpha_{i,k} C_i \quad (3.73)$$

$$\frac{\partial \omega_k^{TB}}{\partial C_l} = C_{M,k} \frac{\partial \omega_k}{\partial C_l} + \omega_k \sum_{p=1}^{N_{nq}} \alpha_{p,k} \quad (3.74)$$

All these modifications discussed till now are involved with the reduction of the matrix dimension corresponding to only non-QSS species. However, the major change due to the introduction of QSS species is the additional terms involved in the partial derivative of reaction rate, ω_k . Let us assume that Q_k is the QSS species involved in reactant side of k th reaction (only one QSS must exist in one reaction because of the linear coupling assumption). The algebraic expression of the concentration of Q_k involves non-QSS, other QSS species concentrations and rate constants $k_{f,k}$, thus the partial derivative of reaction rates should have extra terms corresponding to the derivatives of the QSS concentration.

Then, Eq. 3.38 should be reformulated as

$$\frac{\partial \omega_k}{\partial C_l} = k_{f,k} \left(v'_{l,k} C_l^{v'_{l,k}-1} \prod_{i \neq l} C_i^{v'_{i,k}} + \prod_{i \neq Q_k} C_i^{v'_{i,k}} \frac{\partial Q_k}{\partial C_l} \right) \quad (3.75)$$

This derivative is true for a simple elementary reaction with QSS. For third body enhanced or pressure dependent reactions, the partial derivatives of reaction rate follow the same equations discussed before except implementing Eq. 3.75 rather than Eq. 3.38 if QSS is involved. It is important to notice that in Eq. 3.38 (true for a reaction without QSS), for a particular reaction k , $\frac{\partial \omega_k}{\partial C_l}$ is nonzero only when both i and l are the reactants. However, in Eq. 3.75, $\frac{\partial \omega_k}{\partial C_l}$ is nonzero when i corresponds to the reactants, but l corresponds to all non-QSS species that the concentration of Q_k depends on, which includes, but is not limited to the reactants only.

Similarly, Eq. 3.54 should also be modified as

$$\begin{aligned}
\frac{\partial \omega_k}{\partial T} &= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_{nq}} C_i^{v'_{i,k}} Q_k + k_{f,k} \sum_{j=1}^{N_{nq}} v'_{j,k} C_j^{v'_{j,k}-1} \frac{\partial C_j}{\partial T} \prod_{\substack{i=1 \\ i \neq j}}^{N_{nq}} C_i^{v'_{i,k}} Q_k + k_{f,k} \sum_{j=1}^{N_{nq}} C_j^{v'_{j,k}} \frac{\partial Q_k}{\partial T} \\
&= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_{nq}} C_i^{v'_{i,k}} Q_k + k_{f,k} \sum_{j=1}^{N_{nq}} v'_{j,k} C_j^{v'_{j,k}-1} \left(-\frac{C_j}{T} \right) \prod_{\substack{i=1 \\ i \neq j}}^{N_{nq}} C_i^{v'_{i,k}} Q_k + k_{f,k} \sum_{j=1}^{N_{nq}} C_j^{v'_{j,k}} \frac{\partial Q_k}{\partial T} \\
&= \frac{\partial k_{f,k}}{\partial T} \prod_{i=1}^{N_{nq}} C_i^{v'_{i,k}} Q_k - \frac{\omega_k}{T} \sum_{j=1}^{N_{nq}} v'_{j,k} + k_{f,k} \sum_{j=1}^{N_{nq}} C_j^{v'_{j,k}} \frac{\partial Q_k}{\partial T} \tag{3.76}
\end{aligned}$$

The expressions for $\frac{\partial Q_k}{\partial C_l}$ and $\frac{\partial Q_k}{\partial T}$ are evaluated from the algebraic expressions of QSS concentrations, which is dependent on the reaction mechanism itself. The effect of these two final major modifications due to QSS assumption will be tested in the following section, while analyzing the performance of the analytical Jacobian. All other equations involved in analytical Jacobian formulation as discussed in section 3.4 remain unchanged.

3.6 Results and Discussion

The solver used in this study, DVODE [80], is able to provide accurate solution of the governing equations, even if the Jacobian provided to the solver is approximate. In fact, in the absence of an analytical Jacobian, the numerical Jacobian used by the solver is only an approximation of the exact analytical Jacobian. Thus, an accurate solution of ODE does not necessarily ensure the correctness of the analytical Jacobian matrix. However, an exact Jacobian speeds up the solver signifi-

cantly, as the solver converges faster than using an approximate Jacobian. Hence the accuracy and performance of the analytical Jacobian with QSS species are verified in two different ways: how accurate each term of the analytical Jacobian is, and how much speed up it provides compared to the use of numerical Jacobian. Both of them are tested in the context of a n-dodecane/air combustion in a Partially stirred reactor (PaSR).

In PaSR, the flow field is represented by a number of reactive particles carrying their own species composition and temperature. This is an adiabatic and statistically spatially homogeneous reactor, although not perfectly mixed at the molecular level. The composition of each particle evolves in time due to inflow/outflow/pairing, mixing and reaction. A characteristic residence time corresponds to the inflow/outflow rate inside the reactor. Each particle is assigned to a partner and the pairing of these two particles, specified by a pairing time, provides a local mixing process inside the reactor. Although PaSR is not a mathematical representation of the real system, the simplicity of this model permits to explore the effect of reduced chemistry models without a significant computational cost. A detailed description of the PaSR system and methodology is provided in Liang et. al [119]. The ODE solver used for all these simulations is the variable-coefficient VODE [80]. We consider a skeletal n-dodecane mechanism with 54 species and 269 reactions, developed by Yao et al. [153]. This skeletal mechanism was further reduced to a mechanism with 47 species and 198 reactions using YARC, an automatic reduction software described in Pepiot [152]. The reduced mechanism is validated against experimental results and the skeletal mechanism

Table 3.1: Specification of the parameters for PaSR case. Species are given as mass fractions and the initial condition is same as Stream 1.

Parameters	Values
<i>Reactor characteristics</i>	
Pressure (p)	1 bar
Number of particles (n_p)	100
Pairing time (τ_{pair})	1 ms
Mixing time (τ_{mix})	1 ms
Residence time (τ_{res})	10 ms
Time step (Δt)	0.1 ms

itself for laminar flames and auto-ignition [154]. For simplicity, it is made sure that the algebraic expressions corresponding to each QSS species in the reduced mechanism are linear. With this constraint, a maximum of total 18 species are identified to be QSS using level of importance criteria [144]. So, the final reduced n-dodecane oxidation model consists of 29 non-QSS and 18 QSS species. However, the number of QSS species is varied to analyze performance parameters of analytical Jacobian. The PaSR parameters used for the simulations are tabulated in Table 3.1. There are three inflow streams in PaSR: pure fuel, air and a burnt gas pilot. All the simulations are performed for 20 residence times. There exists a transient period for a few initial residence time, after which PaSR reaches a statistically stationary state.

To compare each term of the analytical Jacobian with QSS in our algorithm, exact Jacobian matrices are evaluated via the symbolic differentiation tool in MATLAB [155] and compared to the analytical Jacobian formulation presented in this paper. All the PaSR compositions are collected, and then used for the analytical

Jacobian evaluation in MATLAB and our code. The discrepancy between the analytical and symbolically differentiated Jacobian turns out to be of the order of 10^{-14} in terms of Euclidean norm of the relative errors, which indicates the accuracy of this analytical Jacobian formulation is within the round-off errors. However, the analytical Jacobian formulation is several order of magnitudes faster than the MATLAB symbolic differentiation tool.

The next objective is to verify the performance of the analytical Jacobian with QSS algorithm in terms of speed up in CPU cost. To compare the effect of the major alterations (partial derivatives of reaction rate, ω_k) in analytical Jacobian formulation with QSS species, as discussed in section 3.5, a special case is also simulated for the same PaSR configurations. This special case considers $\frac{\partial Q_k}{\partial C_j}$ and $\frac{\partial Q_k}{\partial T}$ to be zero, thus essentially using Eq. 3.38 and 3.54 for the partial derivatives of reaction rate, ω_k . However, the other minor changes for QSS formulations, e.g., the reduced dimension of analytical Jacobian and so on, are implemented in this special case, which from now onward will be called as approximated analytical Jacobian (AAJ). In this paper, the results from analytical Jacobian (AJ) are compared to both the numerical Jacobian based on finite differences (NJ) and AAJ.

The evolution of mean temperature, mass fractions of OH, CO₂, n-dodecane and two QSS species mass fractions over time are plotted in Fig. 3.2 for all three cases: AJ, NJ and AAJ. The AAJ case provides such an inaccurate Jacobian to the solver that it fails to converge with more than 13 QSS species. As a result, the comparison in Fig. 3.2 is made for PaSR runs with 37 non-QSS and 10 QSS

species. As expected, the profiles for all these cases match exactly with each other. A similar trend is observed while using all 18 QSS species in case of AJ and NJ. Clearly, an approximate Jacobian (NJ and AAJ) does not affect the accuracy of the solution. However, it is definitely more efficient to use the analytical Jacobian (AJ) in terms of CPU performance.

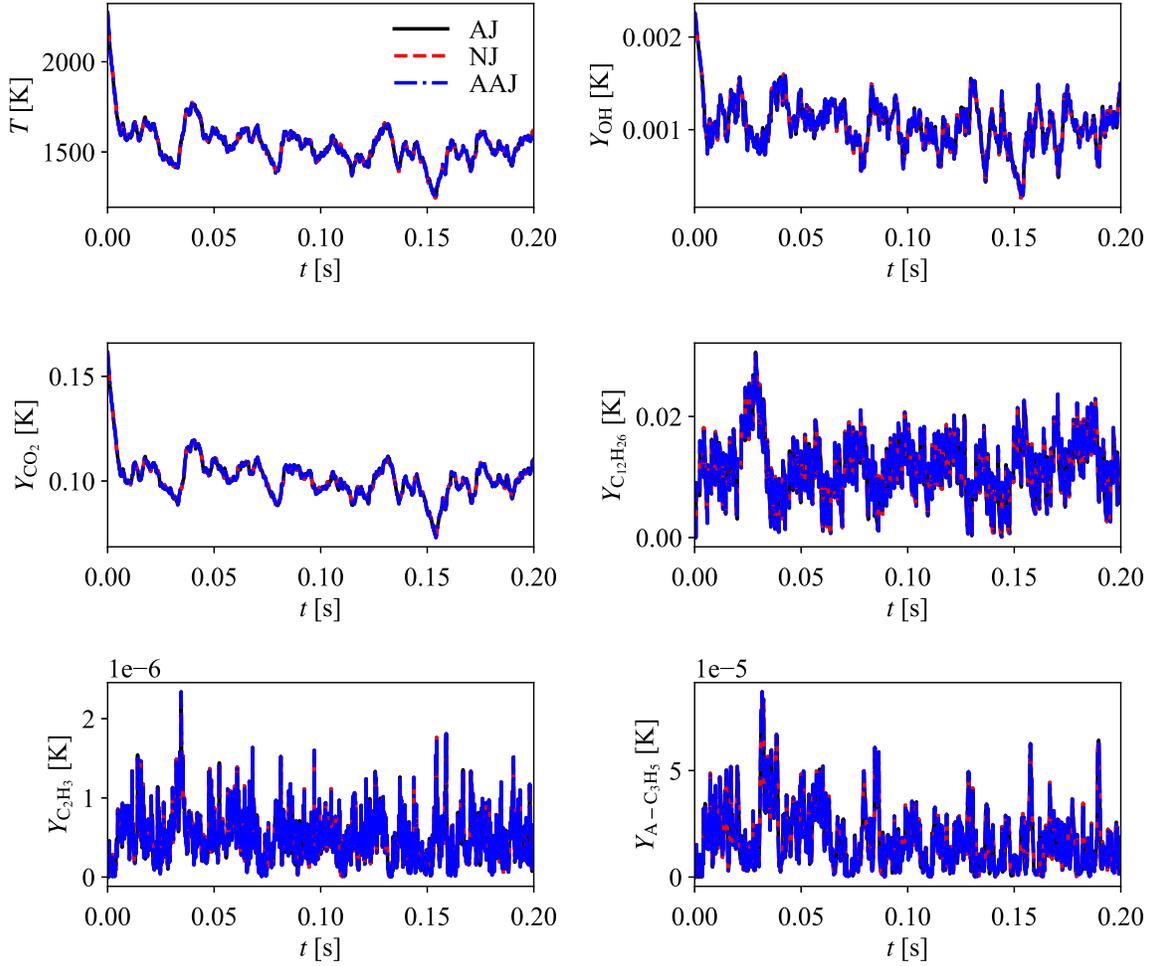


Figure 3.2: Evolution of mean temperature, mass fractions of OH, CO₂ and n-dodecane, and two QSS species: C₂H₃ and A-C₃H₅ in the PaSR using AJ, NJ and AAJ.

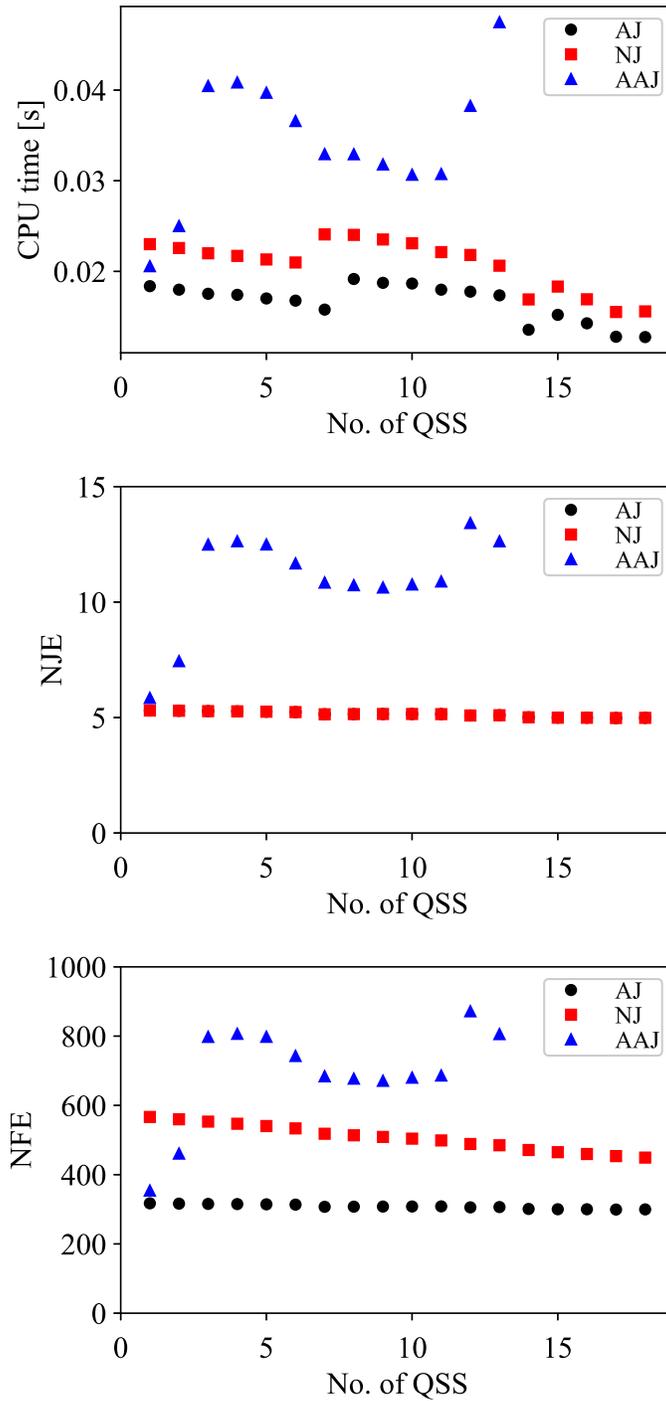


Figure 3.3: CPU time, number of Jacobian evaluations (NJE) and number of source term evaluation (NFE) as a function of number of QSS in the system for three different Jacobian formulations.

Figure 3.3 depicts the variation of different performance parameters while adding QSS species one by one to compare three different cases. The three plots in Fig. 3.3 correspond to average CPU time, number of Jacobian evaluations (NJE), and number of source term evaluations (NFE) for each time step, respectively. All the data points correspond to the statistically steady regime of the PaSR simulation. It is evident from the figure that the CPU time needed for each time step is much lower if an accurate analytical Jacobian (AJ) is provided to the ODE solver rather than the numerical Jacobian (NJ) or the approximate analytical Jacobian (AAJ) to achieve a converged solution. The average CPU time gain for AJ is almost 20% compared to the numerical Jacobian for this set of PaSR simulations. However, the number of Jacobian evaluation (middle plot of Fig. 3.3) in each time step is almost the same for AJ and NJ, but significantly higher for AAJ. The difference in CPU time between NJ and AJ appears from the number of source term evaluations (bottom plot of Fig. 3.3). The number of evaluation of the known mass fractions and temperature source terms increases with decreasing degree of accuracy of the Jacobian matrix in order to get a converged solution within certain tolerance limit in each time step. Consequently AJ provides the most efficient solution method compared to NJ and of course AAJ, which consists clearly an inaccurate Jacobian formulation. In AAJ formulation, the error in analytical Jacobian increases with increasing number of QSS, thus after adding 13 QSS (Fig. 3.3), the error becomes so large that the solution does not even converge.

Figure 3.4 emphasizes on the effect of relative tolerance input values on different performance parameters of the ODE solver. The solver controls the er-

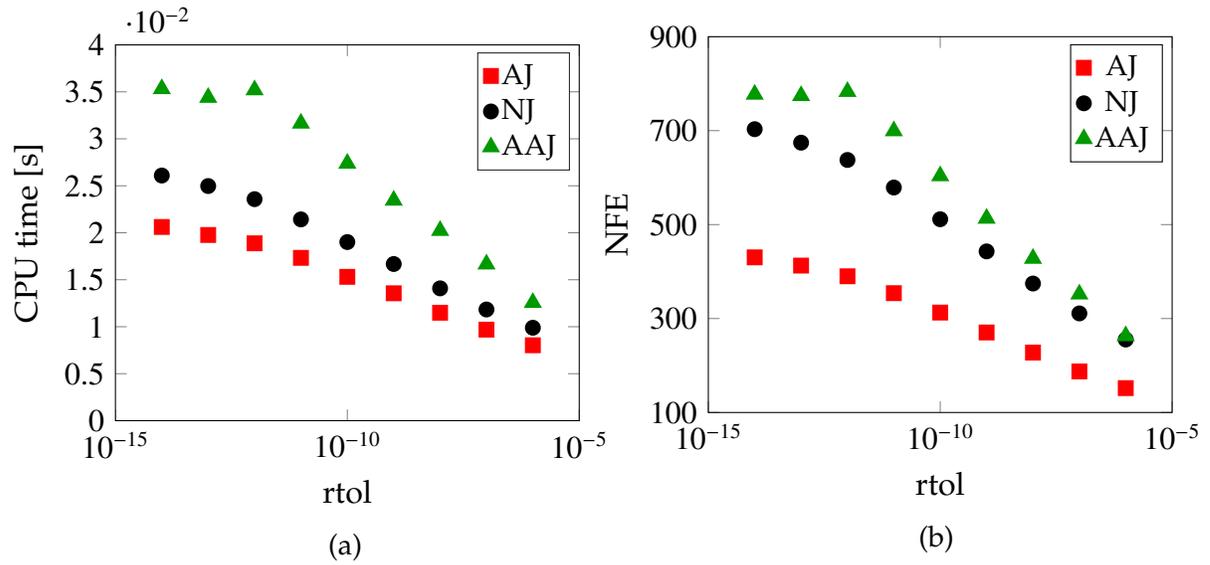


Figure 3.4: Variation of CPU time and number of source term evaluations with varying tolerances of the solver.

ror, e_i in the solution vector Y_i by enforcing an inequality $e_i/EWT_i \leq 1$, where $EWT_i = rtol_i \text{abs}(Y_i) + atol_i$ [80], $rtol$ and $atol$ are the relative and absolute tolerance respectively. The figure clearly indicates that the number of source term evaluation (NFE) increases monotonically with the reduction in $rtol$, consequently increasing both the number of integration sub-steps and CPU time because of a stricter acceptance constraint on local error, e_i . Although not shown, the number of Jacobian evaluation (NJE) also shows a similar trend with reducing tolerance. Similar to Fig. 3.3, the numerical Jacobian (NJ) is more expensive than analytical Jacobian (AJ) because of more number of NFE in case of NJ. As expected, the deviation between performances of NJ and AJ increases as the constraints become more strict with low tolerances. For instance, the analytical Jacobian formulation provided in this paper requires almost 40% less NFE than the numerical Jacobian

when $\text{rtol} = 10^{-14}$. The AAJ, being the least accurate Jacobian among these three cases, requires significantly higher number of source term and Jacobian evaluations to converge to the solution within the given tolerances than the other two, thus turns out to be the most expensive case as observed before.

3.7 Conclusions

In the present work, an analytical Jacobian matrix evaluation approach with QSS species approximation for constant pressure chemical kinetics has been developed. The QSS species are solved using algebraic expressions as a function of all other non-QSS species mass fractions. As a result, the analytical Jacobian needs to take care of the extra derivatives appearing from those algebraic expressions. The package developed here provides all necessary subroutines pertinent to the QSS concentration expressions, and can be easily integrated with any reactive flow simulation. The algorithm is verified in a PaSR simulation with a 47 species dodecane mechanism, out of which 18 species are considered to be QSS. The CPU cost of using an exact analytical Jacobian is observed to 20% less compared to using a numerical Jacobian, as the solver needs to evaluate the source term significantly smaller number of times in case of analytical Jacobian compared to the numerical one.

CHAPTER 4

EFFICIENT TREATMENT OF SECONDARY KINETIC PROCESSES FOR PRE-PARTITIONED ADAPTIVE CHEMISTRY APPROACHES

4.1 Abstract

Probability Density Function (PDF) methods, which allow for the direct integration of chemical kinetics, are well established to accurately simulate turbulent flames with strong turbulence-chemistry interactions. While adaptive chemistry techniques have been proven effective in reducing the high CPU cost and memory requirements associated with the handling of chemistry in such simulations, performance metrics have mostly been focused on the primary oxidation pathways converting fuel to major products. In contrast, this work investigates the ability of adaptive techniques, in this case, the pre-partitioned adaptive chemistry (PPAC) approach, to handle secondary kinetics pathways that are parallel, but tightly coupled to the main oxidation process, taking NO_x formation as a case study. PPAC relies on a partitioning of the composition space into a user-specified number of regions, on which specialized reduced models are generated using the Directed Relation Graph with Error Propagation (DRGEP) reduction technique. The direct application of that methodology to a mix of hydrocarbon oxidation and nitrogen-related targets is shown to yield excessively detailed region-specific reduced mechanisms in order to properly capture both the main oxidation and the secondary NO_x formation processes, thereby decreasing the benefits of the adap-

tive approach. To address this issue, a sequential approach is proposed for the generation of the region-specific reduced mechanisms, in which the primary combustion pathways relevant for each region are identified first, followed by the selective addition, directly at the reduced level, of any secondary pathways relevant for that region using a recently developed build-up technique. This new strategy is assessed in the context of propane combustion in a partially stirred reactor (PaSR) and methane combustion in the Sandia Flame D configuration, demonstrating in both cases the benefits of the sequential approach for reduced model generation.

4.2 Introduction

With recent developments in combustion research, using accurate chemical kinetic models has become essential to simulate multidimensional reactive flow problems. However, accommodating such detailed chemistry in actual computational fluid dynamics (CFD) simulations for realistic fuels and their surrogates is still prohibitively expensive. Especially for turbulent combustion models based on a particle LES/PDF approach, the CPU cost and memory requirements of retaining the full species state vector to evaluate the chemical source term impose a low upper limit of the number of variables that can be used to describe the chemistry [8].

Adaptive chemistry approaches, which tailor the fidelity and size of the kinetic models used for reaction integration to the local flame conditions, have gained

significant attention recently due to their ability to accurately describe the relevant combustion kinetics with significantly fewer variables and equations. The reduced mechanisms can be developed on-the-fly [100, 106, 156] or in an offline pre-processing stage [109, 113, 114, 119]. The pre-partitioned adaptive chemistry (PPAC) [119] follows the latter approach: a set of reduced models is developed in an offline, pre-processing step by partitioning the composition space into several regions, and applying the DRGEP technique [51] in each of those regions using a sample composition database assumed to be representative of the composition space accessed at runtime. Those models are then dynamically selected and used in the actual CFD simulation based on local conditions. This technique has shown good performance in both partially stirred reactors (PaSR) [119, 120] and LES/PDF simulations [42]. However, in all cases, the performance metrics were defined in terms of the main combustion features: prediction of temperature and mass fractions of species directly involved in the hydrocarbon oxidation process. In this work, we further investigate the performance of PPAC, taken here as a representative of adaptive chemistry approaches, in accurately capturing both the main oxidation pathways and some secondary chemical process whose kinetics are driven by, but remain distinct from, the combustion itself. We take NO_x formation as a specific case study.

In DRGEP-based PPAC, the region-specific reduced kinetic mechanisms are obtained by ranking species and reactions by their importance in the prediction of a set of targets (e.g. species or heat release), and removing from the detailed mechanism those species and reactions deemed unimportant. Typical targets in-

cluded to capture the main oxidation process are heat release, fuel, and some major products (CO_2) or radical (OH). Capturing the pathways associated with NO_x formation requires the addition of nitrogen-containing targets, the most obvious choices being NO and NO_2 . We show in the following that when NO_x species are included in the set of targets, the DRGEP reduction procedure becomes overly conservative compared to the oxidation-only case, yielding larger region-specific reduced models for a given level of accuracy in targets prediction.

To address this issue, we propose a sequential approach to the generation of the region-specific kinetic models that leverages a new additive procedure for kinetic model reduction [59]. First the fuel oxidation pathways are identified in each region of the composition space using the original PPAC methodology, only including oxidation species as targets. This initial step is followed by selectively adding, directly at the reduced level, the appropriate NO_x reactions using the “building” algorithm recently developed by Heberle et al. [59]. This paper is organized as follows: the configurations and simulation frameworks used in the study, namely a partially stirred reactor and a CFD solver with LES/PDF capabilities are presented in Section 4.3, followed by a brief description of the two algorithmic components of this work: PPAC [119] and the building algorithm [59] in Section 4.4, and how those two components are integrated into the sequential procedure for region-specific reduced model development. The results of the two specific cases investigated as part of this work, a propane/air partially stirred reactor and a LES/PDF of Sandia Flame D are then presented and analyzed in Section 4.5. Conclusions are provided in Section 4.6.

4.3 Numerical framework

Two numerical configurations are used in this work: a zero-dimensional partially stirred reactor, and a LES/PDF three-dimensional turbulent flame. A brief description of each that includes the most relevant information is provided here, the reader being referred to the cited work for additional details.

4.3.1 Partially stirred reactor (PaSR)

A PaSR represents a statistically spatially homogeneous flow-field with a constant number of particles evolving in time according to three distinct processes, or fractional steps: inflow/outflow, mixing, and reaction. In the inflow/outflow step, particles are selected randomly, and their compositions are replaced by fixed inflow compositions according to a user-specified residence time and stream mass flow rates. A pairwise mixing model is then used to partially mix compositions ($\phi^{(n)}$ and $\phi^{(n+1)}$ for particles n and $n + 1$, respectively) between randomly chosen pairs of particles [68] based on a user-specified mixing time scale:

$$\begin{aligned}\frac{d\phi^{(n)}}{dt} &= -\frac{\phi^{(n)} - \phi^{(n+1)}}{\tau_{\text{mix}}} \\ \frac{d\phi^{(n+1)}}{dt} &= -\frac{\phi^{(n+1)} - \phi^{(n)}}{\tau_{\text{mix}}}\end{aligned}\tag{4.1}$$

where τ_{mix} is the mixing timescale. Finally, all compositions evolve in time due to reaction under adiabatic, isobaric conditions:

$$\frac{d\phi^{(n)}}{dt} = \mathcal{S}(\phi^{(n)})\tag{4.2}$$

where S is the rate of change due to chemical reactions (or chemical source term) of composition $\phi^{(n)}$. A PaSR first goes through a transient stage for a few residence times before reaching a statistically stationary state. The PaSR implementation used in this work is described exhaustively in Liang et al. [119].

4.3.2 LES/PDF flow solver

The LES/PDF of Sandia Flame D is a hybrid mesh-particle method, which is performed here using the variable density low-Mach solver NGA [157]. The LES solver computes the filtered velocity field with second order accurate discretization in space and time. The turbulent viscosity and diffusivity are computed using a Lagrangian dynamic subgrid-scale model [158].

For the reacting flows, a Lagrangian particle PDF method has been implemented in NGA [42] to solve for the one-point one-time joint density weighted filtered PDF of species mass fractions and enthalpy. In this approach, an equivalent particle system is designed to represent the PDF and its evolution. For each particle, a set of stochastic differential equations (SDE) is solved to account for particle transport, mixing, and reaction [37]. These SDEs are integrated using a simple first-order splitting method. The transport fractional step updates the particle position in physical space, the mixing fractional step handles changes occurring in particle compositions due to molecular mixing, and the reaction fractional step advances particle compositions to account for chemical reactions. The

transport fractional step is performed using a simple forward Euler method. The mixing fractional step uses the IEM model [27]. Finally the reaction fractional step is performed efficiently using a dynamic load balancing strategy [159].

The LES/PDF solver is two-way coupled, with the LES solver using a resolved density computed from the PDF solution. Additionally, the resolved composition from the PDF solver is used to compute the kinematic viscosity and molecular diffusivity. The resolved velocity field and turbulent diffusivity from the LES solver are used to advance the particle positions.

4.4 Methodology

Two key components are integrated in the proposed sequential approach for reduced model generation: the adaptive chemistry approach itself, and the additive methodology, referred to as the building algorithm below, used to identify the kinetics associated with the secondary pathways.

4.4.1 Pre-partitioned adaptive chemistry (PPAC)

PPAC [119] is tailored for particle PDF methods to simulate turbulent combustion systems, and consists of two stages: an offline pre-processing stage and the adaptive simulation stage. The pre-processing stage entails the generation of a set

of reduced models, and the adaptive stage selectively utilizes those models during the reaction fractional step. The pre-processing stage consists of the following steps:

1. **Database creation:** A database of compositions is assembled first using the detailed kinetic mechanism. The database is expected to be representative of the compositions likely to be encountered at runtime.
2. **Partitioning:** Using this database and a cutting-plane algorithm, the composition space is partitioned into a user-specified number of regions, so that the compositions in each region are kinetically similar. The partition is stored conveniently in the form of a binary tree.
3. **Generation of Reduced models:** A reduced kinetic model and its corresponding representation is then generated for each region using DRGEP, the compositions in the database belonging to that region, a set of user-specified targets, and an error cut-off ϵ_c . The error measure is defined here by directly comparing reduced and detailed predictions of the targets and temperature:

$$\epsilon = \max_{\mathcal{R}} \left(\frac{\|\mathbf{Y}_{\mathcal{T}}^D(\Delta t) - \mathbf{Y}_{\mathcal{T}}^R(\Delta t)\|_2}{\|\mathbf{Y}_{\mathcal{T}}^D(\Delta t)\|_2}, \left| \frac{T^D(\Delta t) - T^R(\Delta t)}{T^D(\Delta t)} \right| \right) \quad (4.3)$$

where the maximum is taken over all particles in the database belonging to the region \mathcal{R} under consideration, $\mathbf{Y}_{\mathcal{T}}(\Delta t)$ and $T(\Delta t)$ are the targets mass fraction vector and the temperature after Δt , and the superscripts D and R refer to the use of detailed and reduced models, respectively.

During the actual simulation, the key steps through which a particle composition evolves in time are as follows:

1. **Reconstruction:** Before the mixing fractional step, the detailed representation of the particle composition is reconstructed from its reduced skeletal representation.
2. **Mixing:** In their detailed representation, the compositions are mixed following Eq. 4.1.
3. **Classification:** After mixing, each compositions is classified into one of the regions using an efficient low-dimensional binary tree search algorithm.
4. **Reduction:** Each particle composition is then converted to its reduced skeletal representation corresponding to the region it belongs to.
5. **Integration:** The particle compositions in their reduced representations are then advanced in time due to reaction following Eq. 4.2.

A full description of the PPAC algorithm including the reduced representation and reconstruction is provided in [119].

4.4.2 Building algorithm

The building algorithm is a DRGEP-based iterative bottom-up approach to generate reduced kinetic mechanisms. Unlike the conventional graph-based reduction

methods that rely on simulations using the detailed mechanisms to identify and remove unimportant species and reactions in a top-down fashion, the building algorithm follows an add-as-needed approach. Here the reduced mechanisms are progressively ‘built’ with reactions selected from the detailed mechanism using a DRGEP-derived criterion in order to accurately predict the evolution of a set of user-defined targets, \mathcal{T} . Most importantly, this algorithm provides a framework to incrementally expand existing reduced models to accommodate a wider range of conditions, or to include new chemical pathways, such as NO_x chemistry in this work. A summary of the algorithm as used in this paper is provided here, with further implementation details available in Heberle et al. [59].

Let’s consider a set of compositions, an initial reduced mechanism \mathcal{M}_0 , a time interval Δt , a list of additional target species that were not accounted for in the original reduction \mathcal{T}^B , and a desired error cut-off ϵ_c . The objective of the building process is to identify a minimal set of additional reactions required to properly capture the evolution of the target species in each composition over the time interval Δt with an error $\epsilon < \epsilon_c$. Since ϵ is an *a posteriori* measure, that is, it can only be calculated when the reduced mechanism is known, a surrogate criterion corresponding to a DRGEP coefficient cut-off, ϵ_{DRGEP} , is used instead during the building process. ϵ_{DRGEP} is initialized to a large value. The steps are as follows:

1. We define the edge of the mechanism \mathcal{M}_0 , denoted by $\delta\mathcal{M}_0$, as the set of reactions from the detailed mechanism that satisfy one of the following conditions:
 - (a) every reactant in the reaction is already present in \mathcal{M}_0 , or
 - (b) every product

in the reaction is already present in \mathcal{M}_0 . The union of \mathcal{M}_0 and $\delta\mathcal{M}_0$ is called the test mechanism (\mathcal{M}_0^*).

2. DRGEP reaction coefficients (as defined in [51]) are computed for all compositions using \mathcal{M}_0^* over the time interval Δt . The DRGEP reaction coefficients rank the reactions according to their importance in the context of the target species and temperature.
3. Reactions in $\delta\mathcal{M}_0$ with a DRGEP reaction coefficient larger than ϵ_{DRGEP} are added to \mathcal{M}_0 .

Steps 1-3 are repeated with the updated \mathcal{M}_0 until there is no reaction left in $\delta\mathcal{M}_0$ satisfying the criteria in step 3. At this point, the actual error metric ϵ is evaluated. If it is larger than ϵ_c , the process is restarted using a smaller ϵ_{DRGEP} in order to refine the resulting reduced model. The process stops when ϵ becomes smaller than ϵ_c .

4.4.3 Additive treatment of secondary pathways in PPAC

In the regular PPAC process (simply labeled “PPAC” below), the region-specific reduced models for a given ϵ_c are directly obtained using a single set of targets including both oxidation and nitrogen-related species. In contrast, when using the building algorithm to handle the secondary processes, the algorithm, labeled “PPAC-Additive” below, proceeds in two distinct stages, both done during pre-processing:

1. **Reduction:** In the first stage, region-specific reduced models are generated according to the desired error cut-offs, but *using the oxidation targets only*.
2. **Building:** In the second stage, starting from the oxidation-only reduced mechanisms, the building algorithm is applied using a set of targets relevant for the secondary processes only (e.g. NO, NO₂, and heat release). The second stage stops when all regions satisfy their error cut-off criterion, now defined based on the secondary set of targets.

The resulting region-specific models are then used in a similar way during the adaptive simulations. The next section quantifies in two different configurations the benefits of using 2-stage approach. To improve the treatment of PPAC-Additive, two additional modifications are performed compared to the original PPAC and building algorithm, discussed in the previous sections.

1. The first key modification is in the PPAC algorithm itself. The error cut-off ϵ_c in the pre-processing stage of PPAC is made region-specific by using a similar scaling process as that initially developed for DRGEP [51], which allows for greater model reduction in regions where the targets are less chemically active. The region-specific error cut-off ϵ_c^i for region i is defined as:

$$\epsilon_c^i = \frac{\epsilon_c}{\alpha_G^i} \quad \text{where} \quad \alpha_G^i = \frac{\langle \alpha_t^i \rangle_{t \in \mathcal{T}}}{\max_i \langle \alpha_t^i \rangle_{t \in \mathcal{T}}} \quad (4.4)$$

where α_t^i is the scaling coefficient of target t corresponding to the compositions that belong to region i [51, 119], and $\langle \cdot \rangle_{t \in \mathcal{T}}$ denotes the mean over the set of targets, \mathcal{T} . The scaling coefficient α_t^i quantifies the contribution at a time t of each

target to the overall activity of the compositions in region i . This scaling coefficient is unity when the target contributes to its maximum to the exchange of atoms between species, and zero when, for instance, the species mass fraction is constant, at chemical equilibrium, or when the target has been consumed entirely. In essence, α_G^i is unity for the most important region for \mathcal{T} , and a lower than unity value of α_G^i results in an increase in ϵ_c^i , and consequently involves in more reduced models.

2. Second, the building algorithm is modified in PPAC-Additive from its original form [59] as described below:

- (a) In the building stage of PPAC-Additive, both species and reaction DRGEP coefficients are evaluated for \mathcal{M}^* . Then reactions in $\delta\mathcal{M}_0$ with a DRGEP reaction coefficient larger than ϵ_{DRGEP} and involving the species with the highest DRGEP coefficients are added to \mathcal{M}_0 .
- (b) In contrast to the original building algorithm, ϵ_{DRGEP} is not a fixed value in PPAC-Additive. During the iterative process, ϵ_{DRGEP} is adaptively reduced in order to add necessary reactions to achieve a certain error threshold, ϵ_c^i for region i . As a result, the user defined parameter in this algorithm is the error threshold, ϵ_c , as opposed to ϵ_{DRGEP} in [59].

4.5 Results and Discussion

4.5.1 PaSR configuration

The first configuration under consideration is a non-premixed piloted propane-air PaSR. The detailed propane mechanism used is that of Petersen et al. [160], with an additional NO_x sub-mechanism [161], yielding a mechanism with 140 species and 1695 reactions. The PaSR parameters are those used in [119], with 100 particles and three inflow streams: pure fuel, air and a burnt gas pilot. The PaSR is initially run for three residence times after which it reaches a statistically stationary state. 10000 distinct compositions from the detailed run are then randomly sampled over the next 10 residence times to form the composition database. The number of regions is specified to be 10. The oxidation targets are the fuel, CO_2 , CO , OH , and heat release, the nitrogen related targets being NO and NO_2 .

First, we compare the error incurred in the prediction of the NO mass fraction and the temperature obtained during the adaptive simulation when using the region-specific models obtained with PPAC and PPAC-Additive, as a function of the relative number of species used in the kinetic models in each case. The incurred error for a quantity X is defined as:

$$\varepsilon_X = \frac{\sum_{k=1}^{n_t} \sum_{n=1}^{n_p} |X_k^{(n),A} - X_k^{(n),D}|}{\sum_{k=1}^{n_t} \sum_{n=1}^{n_p} |X_k^{(n),D}|}, \quad (4.5)$$

where n_t is the number of time steps and n_p , the number of particles. $X_k^{(n),A}$ and $X_k^{(n),D}$ represent the values of quantity X for particle n at the k -th time step using

the adaptive and detailed chemistry, respectively. The relative number of species used throughout the simulation is defined as:

$$n_{\text{rel}} = \frac{1}{n_t n_p n_s} \sum_{k=1}^{n_t} \sum_{n=1}^{n_p} n_{s,k}^{(n),A}, \quad (4.6)$$

where n_s is the number of species in the detailed model, and $n_{s,k}^{(n),A}$ is the number of species in the reduced model used by particle n at time step k . The comparison is shown in Fig. 4.1.

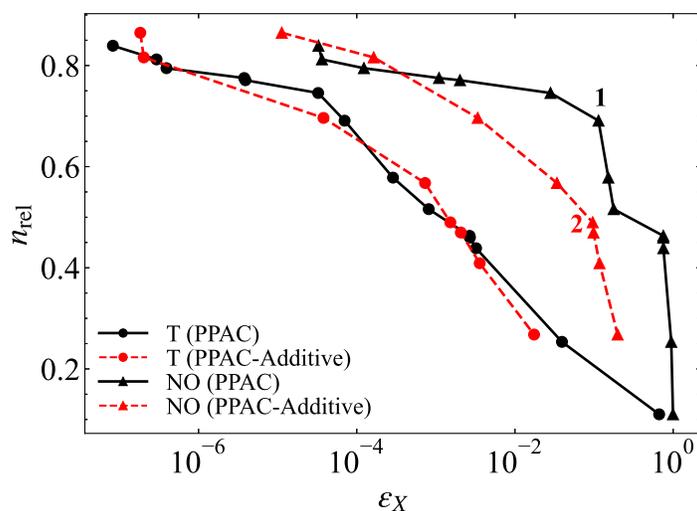


Figure 4.1: Relative number of species as a function of incurred error in temperature (circles) and NO mass fraction (triangles) for PPAC (black solid line) and PPAC-Additive (red dashed line) compared to the non-adaptive case. Points marked as ‘1’ and ‘2’ are used for further analysis in the text.

For both model generation techniques, we observe, as expected, an increase in incurred error for both temperature and NO mass fraction as the relative number of species decreases, that is, as the region-specific models become more reduced. The temperature curves for both approaches are very similar, indicating that a similar level of reduction will lead a similar error in temperature prediction for

both approaches. However, the trend is noticeably different when looking at NO prediction. At very low incurred error levels, corresponding to 80% of the species included on average in the region-specific models, the two methodologies do not show any notable difference. However, as the incurred error increases, the relative number of species required to reach a given error in the prediction of NO is observed to be significantly smaller in the PPAC-Additive case compared to the regular PPAC. For instance, to achieve a 10% error in NO mass fraction, PPAC (point '1' in in Fig. 4.1) requires 70% of the detailed species as opposed to only 48% for PPAC-Additive (point '2'): the model reduction done accounting for all targets at once, oxidation and NO_x formation, appears significantly less efficient at identifying reliably the dominant pathways necessary to predict NO formation.

Exploring this key observation further, n_{rel} is divided into two components: the relative number of oxidation species, $n_{\text{rel, Ox}}$, and nitrogen-containing species, $n_{\text{rel, NO}_x}$, both normalized by n_s . Those two numbers are plotted against ϵ_{NO} in Fig. 4.2. In both approaches, $n_{\text{rel, NO}_x}$ turns out to be very similar, attributing the difference in n_{rel} observed in Fig. 4.1 solely to an excess number of oxidation species selected by the regular, all-targets-at-once, reduction procedure.

During the reduction stage of PPAC, the unimportant species are removed according to their DRGEP species coefficients, which are calculated over all targets of interest. The results above provide evidence that conventional graph-based reduction methods that require users to specify a list of targets of interest do not optimally handle sets of targets associated with chemical pathways very different in

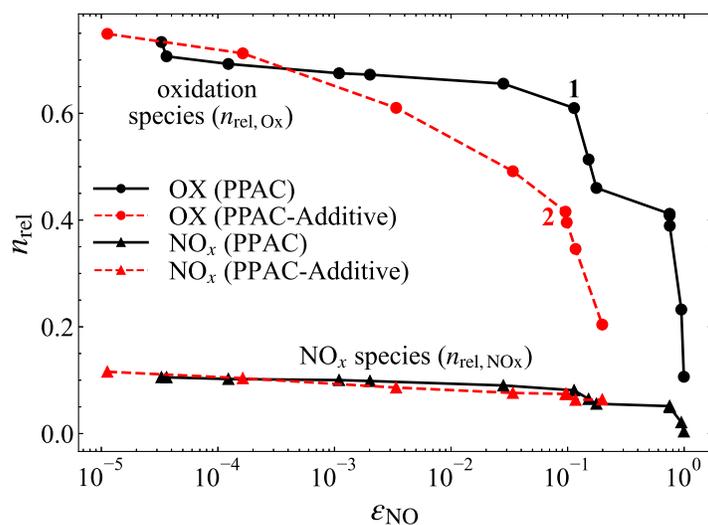


Figure 4.2: Relative number of oxidation (“OX” in legend) and NO_x species as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.

nature, e.g., in this case, the primary pathways (fuel oxidation) and the pollutant formation (NO_x chemistry). When different targets are handled without distinction, unimportant oxidation species are added prior to the addition of important nitrogen-related species, their DRGEP species coefficients being lower and consequently appearing much later in the importance ranking. The NO_x chemistry have different time scales and pathways from the oxidation chemistry. As the error metric takes the maximum error over all targets (Eq. 4.3), a lot of unimportant oxidation species affects the oxidation targets more than the important NO_x species affecting NO and NO_2 within a given time scale. As a result, the DRGEP ranking of those unimportant oxidation species turn out to be higher. The additive procedure highlighted in this work effectively decouples primary and secondary targets, enabling a more selective identification of the species involved in the secondary process and its coupling to the primary combustion pathways. During

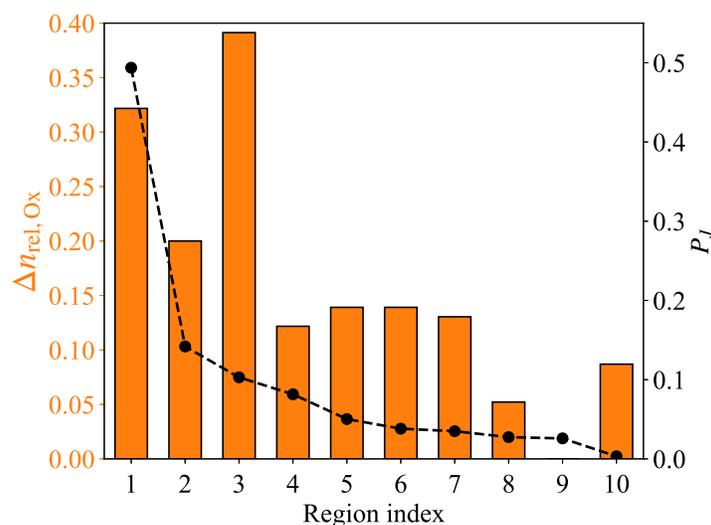


Figure 4.3: Region-specific difference in the number of oxidation species retained by the PPAC and PPAC-Additive reduction procedures, corresponding to an equivalent incurred error in the adaptive simulation. The circles indicate the probability of a particle to be in region J , P_J .

building of NO_x sub-mechanism, the errors are based solely on NO and NO_2 , which helps to keep the important NO_x species higher in DRGEP ranking.

Further region-specific analysis can be done by comparing the models corresponding to point '1' (PPAC) and '2' (PPAC-Additive), since both result in a similar incurred error. In particular, we can investigate the difference in the number of oxidation species retained in each region between those two cases, normalized by the number of oxidation species in the detailed mechanism. The results are plotted in Fig. 4.3. The secondary y -axis shows the probability P_J of a particle to be classified to region J during the adaptive PaSR simulation. The kinetic model associated with Region 1, which is most frequently used ($\sim 50\%$ of the particles) contains 32% more oxidation species when the additive process is not used and

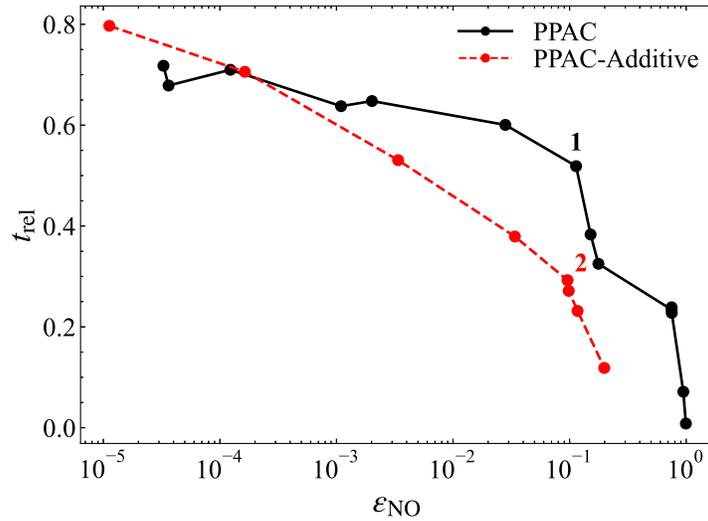


Figure 4.4: Average clock time per time step relative to the detailed simulation as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.

all targets are handled together, thus contributing the most to the difference observed in Fig. 4.1. On the other hand, the kinetic model associated with Region 9, although containing the same number of oxidation species in both cases, is only used for 3% of the particles. Figure 4.4 shows the average wall clock time per time step relative to the detailed simulation. As a direct result of the difference in the number of species retained, the additive procedure for the generation of the reduced models also significantly impacts the CPU cost of the adaptive simulations, with for example, a 43% reduction in wall clock time per time step between PPAC-Additive ('2') and the regular PPAC ('1').

4.5.2 LES/PDF of Sandia Flame D

The additive reduction approach is next assessed in the context of LES/PDF with the simulation of Sandia Flame D [162]. The cylindrical configuration is 60D and 20D in the axial and radial directions, respectively. A non-uniform 192 (axial) \times 140 (radial) \times 32 (azimuthal) mesh is used. The mixing model constant is specified to be 4, and the number of particles per cell is chosen to be 25 [42]. GRI Mech 2.11 [163] with 48 species and 554 reactions is used for these computations. To accelerate the LES/PDF calculations, in-situ adaptive tabulation (ISAT) [68] is used in all LES/PDF runs. The ISAT tolerance is specified to be 10^{-4} , and the maximum table size is set to 500 MB for the detailed mechanism. During the adaptive simulation, the region-specific ISAT tables are allowed to reach a maximum size of 50 MB. The additional error introduced by ISAT is assumed to be negligible compared to the error introduced by the chemical reduction itself.

As the focus of this work is the performance of the additive reduction procedure in handling NO_x formation in an adaptive chemistry LES/PDF simulation, the composition database for the partitioning and reduction is obtained by down-sampling the composition particles obtained from an instantaneous snapshot of the detailed LES/PDF of the Sandia Flame D itself. The targets and number of regions are the same as for the PaSR case, and the partitions and region-specific kinetic models of varying levels of reduction are obtained in a similar fashion as above, using methane oxidation and nitrogen targets together (PPAC) or first using methane oxidation targets only, followed by the addition of reduced NO_x

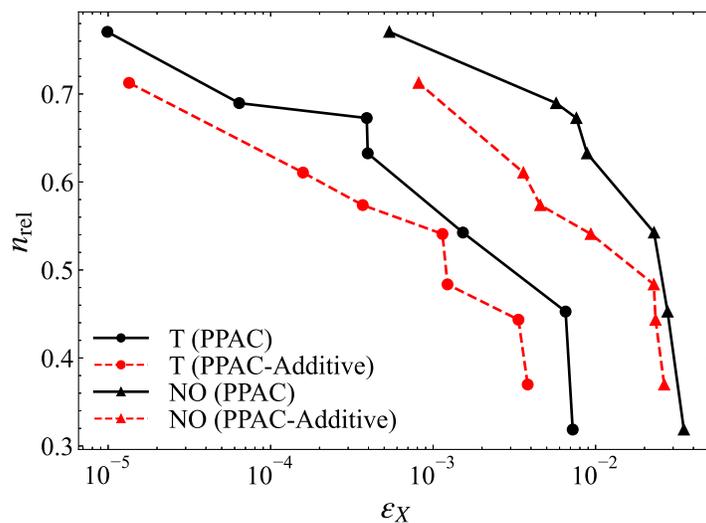


Figure 4.5: Relative number of species as a function of incurred error in temperature (circles) and NO mass fraction (triangles) for PPAC (black solid line) and PPAC-Additive (red dashed line) compared to the non-adaptive case.

pathways using the building algorithm (PPAC-Additive). The incurred errors for the LES/PDF cases are obtained from Eq. 4.5 by integrating the database compositions over 50 time steps using the detailed mechanism as reference. These 50 steps only include reaction source term integration, which essentially means 50 consecutive 0D constant pressure steps, without any micro or macro-mixing. The incurred error in temperature and NO mass fraction as a function of the relative number of species used is plotted in Fig. 4.5

Similarly to Fig. 4.1, the results clearly show that for a given incurred error in NO mass fraction, the additive procedure yields smaller region-specific models. Fig. 4.6, which breaks down the number of species into oxidation and nitrogen-containing species, confirms again that the difference in reduced model sizes

comes primarily from an excessive number of oxidation species selected when oxidation and NO_x targets are handled together.

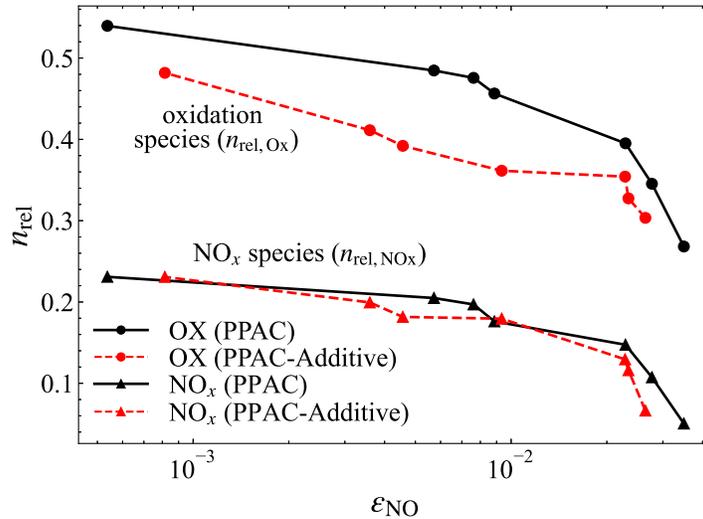


Figure 4.6: Relative number of oxidation ('OX' in legend) and NO_x species as a function of incurred error in NO mass fraction for PPAC and PPAC-Additive.

Two full-scale Sandia D flames are then simulated using set of models obtained with PPAC and PPAC-Additive generating less than 0.5% incurred error in NO mass fraction (Fig. 4.5). Figure 4.7 shows the instantaneous particle distribution for the PPAC-Additive (or PPAC)-ISAT LES/PDF simulation colored with temperature on the left and NO mass fraction in the middle. The right panel shows the same particle distribution colored with the index of the reduced kinetic model used by the corresponding particle for the chemical source term integration. Figure 4.8a and 4.8b show a break down of the number of oxidation and NO_x species in each of the models for PPAC and PPAC-Additive, respectively, using the same color map as the right panel of Fig. 4.7.

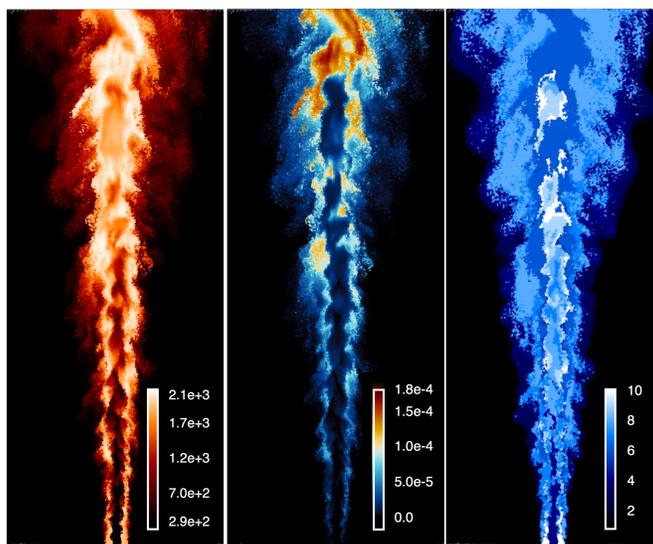


Figure 4.7: Instantaneous particle distribution colored by temperature (left), NO mass fraction (middle) and index of the model used for reaction source term integration (right).

We observe that the particles in the co-flow region use the reduced model involving the smallest number of species for both PPAC and PPAC-Additive (model 1). The maximum NO mass fraction pockets coincide with regions of higher temperature across the flame brush. These regions can be attributed to the thermal route of NO formation, which is clearly the most important NO_x pathway in Sandia Flame D. Additionally, most of the particles in these high temperature pockets are integrated by model 6, which includes the chemistry associated with interaction of N_2 species with O or OH, important in NO formation. In case of both model 1 and 6, which are the two most used models in the LES/PDF runs, PPAC models contain 35% more oxidation species than PPAC-Additive models, respectively (Fig. 4.8). On the other hand, the prompt NO pathways, which are dominated by the production of HCN from CH (Fenimore mechanism), are mostly covered by models

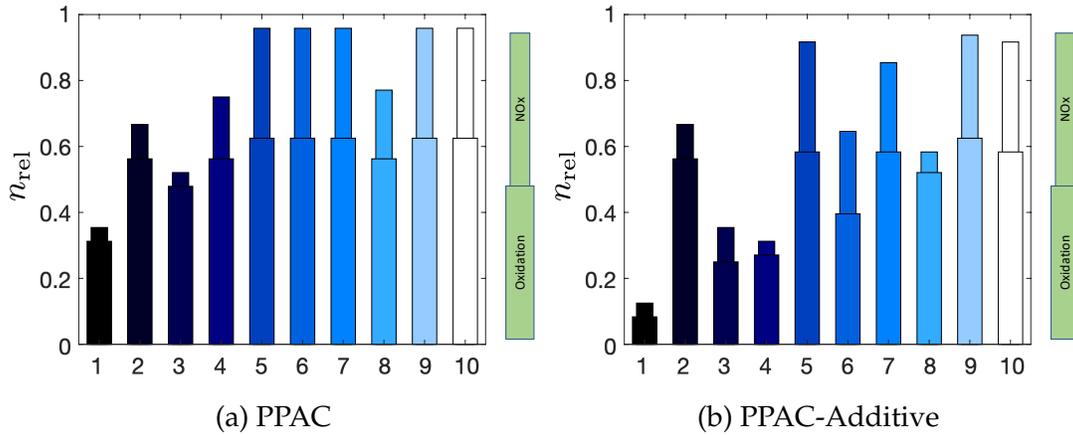


Figure 4.8: Relative number of oxidation and NO_x species in each model with respect to the detailed mechanism for (a) PPAC, and (b) PPAC-Additive. The color pattern matches that of the right panel of Fig. 4.7.

5 and 9. The regions with maximum N_2O mass fraction coincide with particles using model 8, which covers the less important N_2O -related pathways. The other models with high NO_x species observed in Fig. 4.8 share both thermal and prompt NO pathways. Finally, the models with the fewer number of nitrogen-containing species are primarily used near the fuel jet core and in the surrounding area of the flame, where the temperature is low. Overall, the PPAC run utilizes approximately 18% more relative number of species (Eq. 4.6) than PPAC-Additive for the reaction fractional step over two flow-through times, resulting in a 5% reduction in wall clock time per time step for the latter.

Figure 4.9 compares the radial profiles of the resolved mean and RMS of temperature and NO mass fraction obtained from the detailed, PPAC-Additive and PPAC simulations at two different axial locations, $15D$ and $30D$ from the burner exit (D is the burner diameter). Statistics have been collected for two flow-through

times. To quantify the difference between the adaptive (PPAC-Additive or PPAC) and detailed simulations, we use the normalized root mean square difference (RMSD) [39], defined as:

$$\mathcal{E}(\xi) = \frac{[\xi^A - \xi^D]_{rms}}{\xi_{ref}}, \quad (4.7)$$

where ξ^A and ξ^D denote the quantities obtained from adaptive and detailed simulation, respectively, ξ_{ref} is the reference value of each quantity of interest. ξ^{ref} is considered to be 1000K for temperature, and the maximum NO mass fraction among all the radial profiles at both axial locations: $15D$ and $30D$. The $[\cdot]_{rms}$ is computed over all the radial locations at all the considered axial locations. We observe that the RMSD values of resolved mean and RMS of temperature and NO mass fraction are below 3% and 5% of the detailed simulation from both PPAC-Additive and PPAC simulations, respectively.

4.6 Conclusions

A new approach for the generation of reduced chemistry models describing distinct, but coupled chemical processes such as fuel oxidation and NO_x formation has been presented. It relies on a sequential approach, in which reduced models for the primary kinetic process are first developed, followed by a selective addition, directly at the reduced level, of the reactions necessary to capture the secondary chemistry process up to a user-level error tolerance. The procedure has been assessed for NO_x formation in two different configurations: a propane/air

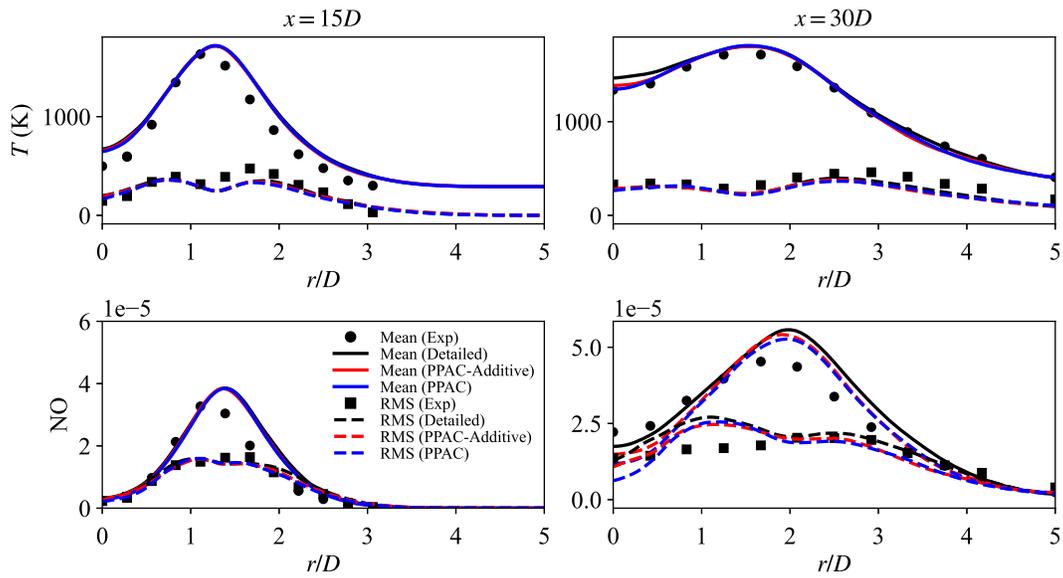


Figure 4.9: Comparison of radial profile of the experimental (Exp) and the computed (Detailed, PPAC and PPAC-Additive) resolved mean and RMS of temperature and NO mass fraction at two different axial locations: $15D$ and $30D$.

PaSR and a LES/PDF of Sandia Flame D. The additive approach was found to be much more selective in identifying the most important reactions pertaining to the secondary process, compared to a traditional reduction approach where targets from both chemical processes are handled simultaneously. The difference between both approaches can be attributed nearly exclusively to an excess number of oxidation species in the traditionally reduced models that are not related to the NO_x formation, and not necessary for the proper prediction of the main oxidation pathways.

CHAPTER 5
A DYNAMICALLY PARTITIONED ADAPTIVE CHEMISTRY
METHODOLOGY FOR EFFICIENT IMPLEMENTATION OF COMBUSTION
CHEMISTRY

Probability Density Function (PDF) methods are well established to accurately simulate turbulent flames with strong turbulence-chemistry interactions. However, these methods are computationally expensive restricting their use to simple fuels with small detailed mechanisms. To mitigate their cost both in terms of CPU time and memory requirements, a dynamically partitioned adaptive chemistry (DPAC) method is presented here which provides a reduced model to each particle at runtime adjusted to its changing composition over time. DPAC starts with a pre-processing stage, where the composition space based on compositions from any low-dimensional simulation that is not necessarily to be accessed during the actual combustion simulation is partitioned into a user-specified number of regions. The regions are initialized with reduced models involving zero reactions and a few species. During the adaptive simulation, the current accessed region of the composition space is re-partitioned, the reduced models are built using selectively chosen important reactions from the detailed model using a Direct Relation Graph with Error Propagation (DRGEP) based building algorithm. Each composition is assigned to a region of the current partition using a low-dimensional binary tree search algorithm, and then advanced in time using the most updated models available for the region. The performance of the algorithm is validated

for a propane/air non-premixed combustion in PaSR and 2D LES/PDF simulation of a lifted n-dodecane flame. The results exhibit the flexibility of the algorithm to adjust the partition and reduced models to the new compositions encountered at runtime. The adaptive framework also shows significant gains in CPU cost and memory requirement compared to pre-partitioned adaptive chemistry (PPAC) and detailed simulation.

5.1 Introduction

With recent developments in chemical kinetics for hydrocarbon fuel combustion, a significant disparity exists between the size of the available detailed kinetic schemes and those that can be used in computational fluid dynamics (CFD) application for realistic fuels and its surrogates. Especially for turbulent combustion models, such as particle PDF methods, which are agnostic to the flame topology, the CPU cost and memory requirements to retain the full species state vector for each particle impose a low upper limit of the number of variables that can be used to describe the chemistry [8].

Out of the numerous techniques developed over the last few decades to reduce the computational cost, adaptive chemistry approaches have gained significant attention. Adaptive chemistry approaches tailor the fidelity and size of the kinetic models used for reaction integration to the local flame conditions. As there exist a lot of non-reactive compositions, a large fraction of the composition space can be described properly with significantly smaller reduced mechanisms. The

reduced mechanisms can be developed on-the-fly [100, 106, 156] or in an offline pre-processing stage [109, 113, 114, 119]. The pre-partitioned adaptive chemistry (PPAC) [119] follows the latter approach. The PPAC pre-processing stage develops a set of reduced models using the DRGEP technique [51] based on a sample composition database. At runtime, each particle evolves in time according to the reduced model corresponding to the region it belongs to. The dynamic use of reduced chemistry instead of a detailed model decreases the overall CPU cost significantly. This approach also mitigates the memory requirement, which is a concern for the particle PDF approaches that involve higher memory requirement compared to flamelet type models [8]. Recently, PPAC is also combined with *in-situ* adaptive tabulation (ISAT) [68] in an LES/PDF simulation of Sandia Flame D [42] to accelerate the chemistry integration step even further.

However, the most important drawback of this PPAC approach is its dependence on the initial database of compositions during the pre-processing stage, which is assumed to be representative of the actual composition space accessed at runtime. As a result, the generation of an accurate database is crucial for the success of PPAC. For example, while demonstrating the PPAC approach in a partially stirred reactor (PaSR) [119, 120], the initial composition database is downsampled from a detailed simulation of the same configuration. Similarly in PPAC-ISAT LES/PDF simulation of Sandia Flame D [42], the compositions required for PPAC are obtained from one time instant of the same flame using the detailed mechanism. The problem with this dependence on the initial composition database is two fold. First, the use of detailed model to generate the database defies the

whole purpose of using adaptive chemistry approaches. A detailed simulation for a more realistic simulation configuration than PaSR with a more complex fuel than methane in Sandia Flame D, is still an unaffordable task. Secondly, in case of some selective well-studied flames, such as Sandia Flame D, it is still possible to run a PaSR with well-tuned parameters to predict the compositions encountered in the actual flame. However, for a less-understood and/or complex flame, there is no guarantee of replicating the compositions *a-priori*. For instance, the slightest of differences in the inflow parameters in a gas-turbine engine, or the typical non-uniformity observed in solid fuel properties, such as in coal or wood, can drastically change the compositions encountered during a simulation.

To address the aforementioned shortcoming of PPAC, a novel dynamically partitioned adaptive chemistry (DPAC) approach is introduced in this work. DPAC also involves a pre-processing stage similar to PPAC, where the composition space is partitioned into regions based on any random database of compositions. The initial database is collected from less expensive low-dimensional simulations, such as 0D auto-ignition or PaSR, which do not necessarily represent of the accessed region of the composition space at runtime. All the reduced models are initialized with zero reactions and the species included in the inflow streams of the combustion simulation. Then at runtime, each reduced model is progressively augmented with reactions chosen from the detailed model using a recently developed direct relation graph with error propagation (DRGEP) based building algorithm [59]. The algorithm also checks the quality of the current partition at runtime, and re-partitions the composition space, if required. Each particle com-

position is assigned to a region in the current partition available, and is advanced in time using the most updated reduced model available for its region.

The remainder of the paper is organized as follows. In Sec. 5.2, we briefly introduce the two numerical configurations used in this study: PaSR and the LES/PDF framework. Section 5.3 provides the overview of adaptive chemistry methodology used in DPAC involving time advancement of each composition, updating the reduced models and partitioning the composition space. This overview is followed by the detailed descriptions of the two important components of DPAC: the building algorithm (Sec. 5.4) and the partitioning algorithm (Sec. 5.5). The algorithm is then applied to propane/air non-premixed combustion in PaSR, and a 2D LES/PDF simulation of a lifted n-dodecane flame. The results are provided and discussed in Sec. 5.6, followed by the conclusions in Sec. 5.7.

5.2 Numerical Framework

Two numerical configurations are used in this work: a zero-dimensional partially stirred reactor (PaSR), and an LES/PDF two-dimensional turbulent flame. The most relevant information regarding these two configurations are briefly discussed in this section, more detailed being available in the cited references.

5.2.1 Partially stirred reactor (PaSR)

A PaSR represents a statistically spatially homogeneous flow-field with a constant number of particles that evolve in time by three fractional steps accounting for inflow/outflow/pairing, mixing and reaction. In the inflow/outflow step, an even number of particles according to a user-specified residence time are selected randomly with equal probability, and their compositions are replaced by inflow compositions. Each particle is then paired with a partner, which are denoted by particle n and $n + 1$. In the mixing fractional step the particle compositions evolve according to the pairwise mixing model [68]:

$$\begin{aligned}\frac{d\Phi^{(n)}}{dt} &= -\frac{\Phi^{(n)} - \Phi^{(n+1)}}{\tau_{\text{mix}}} \\ \frac{d\Phi^{(n+1)}}{dt} &= -\frac{\Phi^{(n+1)} - \Phi^{(n)}}{\tau_{\text{mix}}}\end{aligned}\quad (5.1)$$

where τ_{mix} is the mixing timescale, and a *composition* Φ is an $(n_s + 1)$ dimensional vector containing n_s number of species mass fractions and temperature in the full composition space, \mathbb{F} . Finally, all the compositions evolve in time due to reaction under adiabatic, isobaric conditions in the reaction fractional step:

$$\frac{d\Phi^{(n)}}{dt} = S(\Phi^{(n)}) \quad (5.2)$$

The PaSR initially runs for a few residence times before reaching a statistically stationary state. The details of the PaSR implementation are described in Liang *et al.* [119].

5.2.2 LES/PDF flow solver

The LES/PDF is a hybrid mesh-particle method, which is performed here using the variable density low-Mach solver NGA [157]. The filtered velocity field is computed by the LES solver with second order accuracy in both space and time. The Germano dynamic subgrid-scale model [164] is used to compute the turbulent viscosity and diffusivity. The resolved velocity field and turbulent diffusivity from the LES solver are used in the Lagrangian PDF solver.

For the reacting flows, a Lagrangian particle PDF method has been implemented in NGA [42] to solve for the one-point one-time joint density weighted filtered PDF of species mass fractions and enthalpy. In this approach, the modeled PDF equation is solved using a particle Monte Carlo scheme [7], where notional particles evolve in time in both physical and composition space due to transport, mixing, and reaction following stochastic differential equations (SDE) [37]. These SDEs are integrated using a simple first-order splitting method. The transport fractional step updates the particle position in physical space based on the resolved velocity field from the Eulerian mesh using a simple forward Euler method. The mixing fractional step handles changes occurring in particle compositions (including temperature) due to molecular mixing by using the IEM model [27]. Finally the reaction fractional step advances particle compositions in time by accounting for the chemical reactions. The reaction step is performed efficiently using a dynamic load balancing strategy [159]. The resolved density computed from the PDF solution is then used in the LES solver. Additionally,

the kinematic viscosity and molecular diffusivity are computed from the resolved composition from the PDF solver.

5.3 Overview of DPAC

The overall dynamically partitioned adaptive chemistry methodology consists of two stages: an offline *pre-processing stage*, during which a sample composition database is used to partition the accessed region, and an online *adaptive simulation stage*, which re-partitions the accessed region, builds the reduced models dynamically from scratch during the simulation. Similar to any pre-processing based adaptive chemistry approaches, DPAC identifies the reduced model suitable for each particle at time t to advance in time.

5.3.1 Pre-processing stage

The main purpose of the pre-processing stage is to generate an initial partition and initial reduced models using a sample composition database with detailed mechanism. The pre-processing stage consists of the following tasks:

1. *Database creation*: A sample composition database, \mathcal{D}^0 is generated using the detailed mechanism with any low-dimensional simulation, such as 0D auto-ignition or PaSR. More importantly, the accessed region of the composition

space does not have to be representative of the actual compositions that occur during the simulation of interest, as DPAC has the ability to adjust the partition based on the current accessed region during adaptive simulation, discussed later.

2. *Partitioning*: The initial accessed region of the full composition space, \mathbb{F}^0 , based on \mathcal{D}^0 , is first mapped into a low dimensional classifying space, \mathbb{C}^0 . The partitioning algorithm then partitions the classifying space \mathbb{C}^0 into a user-specified N_R number of regions. The partition in \mathbb{C}^0 space implicitly partitions the \mathbb{F}^0 space as well, thus the J th region is denoted by R_J in both \mathbb{C}^0 and \mathbb{F}^0 space. A brief description of the classification and partitioning algorithm is provided in Sec. 5.5 with more details available in [119]. The objective of the partitioning algorithm is to create regions with kinetically homogeneous compositions, i.e., compositions that display similar short time kinetics. As \mathcal{D}^0 is not representative of the accessed region of the composition space at runtime, this partitioning is not necessarily the most optimal choice for adaptive simulation.
3. *Reduced model initialization*: With the initial partition, region R_J is initialized with a reduced model, \mathcal{M}_J , consisting of zero reactions and the set of species included in the inflow streams of the simulation of interest.

5.3.2 Adaptive simulation

The adaptive simulation consists of two different modes, i.e., *building* and *surveillance*, and the algorithm is capable of switching between these modes, if needed. The purpose of the *building mode* is to obtain an optimal partition and a set of reduced models suitable for the composition space accessed at runtime. Typically, the building mode overlaps with the transient part of the simulation. On the other hand, the *surveillance mode* coincides with the statistically stationary part of the simulation, when the accessed region does not change, and therefore the models and partition are not expected to be modified significantly. In both modes, all the particles in the reactor evolve in time due to mixing and reaction fractional steps. The adaptive algorithm to advance the particle compositions proceeds in the following sub-steps:

1. **Mixing:** In the mixing fractional step, the compositions, $\Phi^{(n)}$ and $\Phi^{(n+1)}$, are mixed following Eq. 5.1 to yield mixed compositions in case of PaSR. In LES/PDF simulation, the particle compositions follow the IEM model [27]. After mixing, the compositions are denoted by $\Phi^{(n),m}$.
2. **Re-partitioning:** If the quality of the current partition of the accessed region is considered to be extremely poor, the accessed region of the composition space, F^t at time t , is re-partitioned at runtime using the same algorithm as in the pre-processing stage. A re-partition is always followed by re-initializing the kinetic models with zero reactions (similar to step 3 of the pre-processing stage) and re-building them. The criteria to decide if a re-partitioning is required or not is

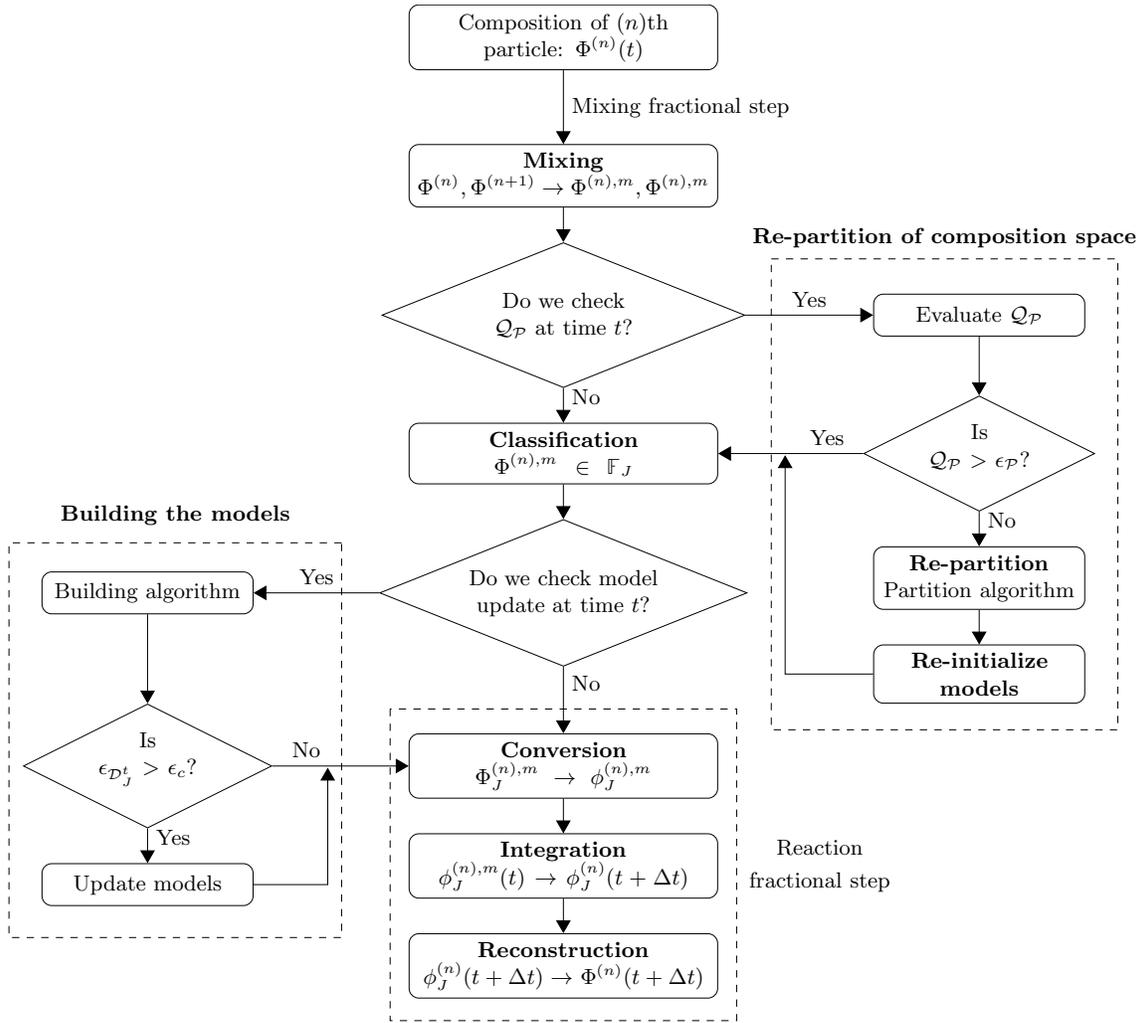


Figure 5.1: Flowchart depicting actions and decision criteria of the adaptive simulation of DPAC at time t .

discussed in Sec. 5.5. This check for re-partitioning is performed after every n_i^P time steps during the simulation.

3. **Classification:** Each particle composition after mixing is assigned to one of the regions by the classification algorithm based on a low-dimensional binary-tree search. A composition in R_j is denoted by $\Phi_j^{(n),m}$.

4. **Model update:** As the simulation progresses in time, region R_J has access to the compositions, $\Phi_J^{(n),m}$, which covers a certain subset of the current accessed region of the composition space encountered at time t , \mathbb{F}_J^t (subset of \mathbb{F}^t). Based on $\Phi_J^{(n),m}$ (or \mathbb{F}_J^t) that belongs to region J , each model \mathcal{M}_J^t is iteratively augmented with individual species and reactions chosen from the detailed mechanism using the building algorithm [59]. The criteria to decide if the model update is necessary or not is discussed in Sec. 5.4 along with the details of the building algorithm itself. Once the model is updated at time t , the check for model update continues for the following time steps, until none of the models are updated for a consecutive n_t^C number of time steps. The next check for an update of models is performed either after n_t^B time steps from the last check or if a re-partition occurs in step 2.
5. **Reduced representation:** Particle representation in full composition space ($\Phi^{(n),m}$) is converted to the reduced skeletal representation ($\phi_J^{(n),m}$) corresponding to the reduced model, \mathcal{M}_J^t , of its assigned region J through classification.
6. **Reaction:** The particle compositions in their reduced representations are then integrated over time Δt following Eq. 5.2 in both PaSR and LES/PDF using \mathcal{M}_J^t for region J .
7. **Reconstruction:** At the end of the reaction fractional step, each particle composition is reconstructed to its detailed representation ($\Phi^{(n)}$) from its reduced skeletal representation ($\phi_J^{(n)}$).

The reconstructed compositions at the end of the reaction fractional step, $\Phi^{(n)}(t +$

Δt), are then used for the next time step. At a time step t , if neither re-partition nor model update requirement is checked, the adaptive simulation algorithm skips step 2 and 4, and advances in time with the most updated partition and set of reduced models. The details of the conversion of particle representation to the reduced representation (step 5) and its reconstruction (step 7) are discussed in Liang *et al.* [119]. These sub-steps of the adaptive simulation is applicable in both *building* and *surveillance* mode of the adaptive simulation.

During the building mode, the partition is modified multiple times along with re-building of models. As a result, re-partition and model update are checked very frequently during the building mode. By the time the simulation reaches a statistically stationary condition, it is expected that the accessed region of the composition space is partitioned optimally, and consequently the frequency of model update drops significantly. At this point, the algorithm switches to the *surveillance mode*, where the re-partition and model update are checked less frequently than the building mode, i.e., n_i^P and n_i^B are significantly larger during surveillance mode than the building mode.

5.4 Building algorithm

The building algorithm is a DRGEP based iterative bottom-up approach to expand the predictive capabilities of reduced kinetic models by selectively adding species and reactions chosen from the detailed mechanism directly at the reduced

level. A summary of the algorithm is provided here, with more details available in Heberle *et al.* [59]. In DPAC, a building step is defined as a small increment in the compositions encountered during a time interval, Δt_b , for an homogeneous reactor simulation. The algorithm works similarly for all regions, thus the reduced mechanism is denoted by \mathcal{M}_J^t at time t . The goal of the building step is then to identify the additional set of reactions required to properly capture the evolution of the particle compositions from t to $t + \Delta t_b$. Starting from \mathcal{M}_J^t and a set of particle compositions encountered at time t in region J , \mathcal{D}_J^t (each composition is denoted by $\Phi_J^{(n)}(t)$), the building algorithm proceeds with the following steps:

1. The edge of the mechanism \mathcal{M}_J^t , denoted by $\delta\mathcal{M}_J^t$, is generated by a set of reactions from the detailed mechanism, which satisfies either of the two conditions: (a) every reactant in the reaction is an element of \mathcal{M}_J^t , or (b) every product in the reaction is an element of \mathcal{M}_J^t . The union of \mathcal{M}_J^t and $\delta\mathcal{M}_J^t$ is called the test mechanism ($\mathcal{M}_J^{*,t}$).
2. Based on the compositions, $\Phi_J^{(n)}(t)$ in \mathcal{D}_J^t , the reaction-specific DRGEP coefficients [51] are computed for all the reactions in $\mathcal{M}_J^{*,t}$. Any reaction from $\delta\mathcal{M}_J^t$ with a DRGEP coefficient larger than a user-specified threshold, ε_{th} , and its corresponding species are tagged. These species and reactions collectively define a sub-mechanism called $\delta\mathcal{M}_J^{+,t}$. We define an updated model at this intermediate stage as $\mathcal{M}_J^{',t} = \mathcal{M}_J^t + \delta\mathcal{M}_J^{+,t}$.
3. Each composition, $\Phi_J^{(n)}(t)$, is advanced in time twice with two different models: \mathcal{M}_J^t and $\mathcal{M}_J^{',t}$. An error is defined as the difference between these two sets of

integrated compositions at $t + \Delta t_b$:

$$\epsilon_{\mathcal{D}'_J} = \max \left(\frac{\|\mathbf{Y}_{\mathcal{T}}^{\mathcal{M}_J} - \mathbf{Y}_{\mathcal{T}}^{\mathcal{M}'_J}\|_2}{\|\mathbf{Y}_{\mathcal{T}}^{\mathcal{M}_J}\|_2}, \left| \frac{T^{\mathcal{M}_J} - T^{\mathcal{M}'_J}}{T^{\mathcal{M}_J}} \right| \right), \quad (5.3)$$

where $\mathbf{Y}_{\mathcal{T}}$ and T are the mass fraction vector of the set of targets, \mathcal{T} , and temperature after time step integration, respectively.

4. At this point, a model update check is performed. If the error, $\epsilon_{\mathcal{D}'_J}$, is greater than a user-specified error threshold, ϵ_c , then \mathcal{M}_J^t is updated as: $\mathcal{M}_J^t \leftarrow \mathcal{M}'_J^t$, and moves on to step 5. Otherwise, the algorithm considers the current model, \mathcal{M}_J^t , to be adequate for all the compositions in region J at time t , and skips to step 6.
5. Steps 1 to 4 are repeated until either (a) there is no reaction left in $\delta\mathcal{M}'_J$ with a DRGEP reaction coefficient larger than ϵ_{th} , i.e., $\delta\mathcal{M}'_J$ is empty, or (b) the error evaluated by Eq. 5.3 goes below ϵ_c .
6. At the end of this iterative process (step 1-5), a standard DRGEP based species reduction [51] is performed starting from the most updated \mathcal{M}'_J . Based on $\Phi_J^{(n)}(t)$, one species at a time and its corresponding reactions are removed from \mathcal{M}'_J . Again, we define this intermediate reduced model as \mathcal{M}''_J . The same error following Eq. 5.3 is evaluated based on two sets of integrated compositions. We continue to remove species until the error, $\epsilon_{\mathcal{D}'_J}$, goes above ϵ_c .

The same procedure of updating the reduced models is repeated for all regions, $J = 1$ to N_R .

5.5 Partitioning algorithm

The objective of the partitioning algorithm is to partition the accessed region of the composition space into a user-defined N_R number of regions so that each composition belonging to region J , $\Phi_J^{(n)}$ exhibit similar short term kinetic characteristics. Consequently, all compositions in region J can be described by a common reduced model, \mathcal{M}_J^t at time t , which contains a smaller number of species and reactions than the detailed mechanism, \mathcal{M}_D . Given a database \mathcal{D} with n_D number of compositions at time t , the most important aspects of the partitioning algorithm are described below. The details of the algorithm are available in [119]. For clarity, the superscript t to represent the current time step is removed from all notations in this section.

1. *Classifying space*: The accessed region of the full composition space, \mathbb{F} , based on \mathcal{D} , is first mapped into a n_C dimensional classifying space, \mathbb{C} , where n_C is a subset of the composition variables (n_s species mass fractions and temperature). These n_C components are identified such that these variables span the n_C dimensional subspace in \mathbb{F} closest to the first n_C principal components of the compositions in \mathcal{D} using principal component analysis (PCA).
2. *Partition structure*: The partitioning algorithm cuts \mathbb{C} using hyperplanes of dimension $n_C - 1$. The structure of the partition is stored in a binary tree. The nodes of the tree contain the equation of the hyperplane, determined by its normal vector and an anchor point, and its leaves contain the subset of sam-

ple compositions in increasingly refined regions under the node. For instance, starting from a region \mathcal{D}_m , which is one of the sub-regions in the previous iteration or initially $\mathcal{D}_m = \mathcal{D}$, a hyperplane is identified that divides \mathcal{D}_m into two sub-regions, $\mathcal{D}_{m,L}$ and $\mathcal{D}_{m,R}$. The node keeps the information of the hyperplane, and its leaves carry the information about $\mathcal{D}_{m,L}$ and $\mathcal{D}_{m,R}$, and their corresponding compositions.

3. *Optimal hyperplanes*: An optimal hyperplane divides \mathcal{D}_m into $\mathcal{D}_{m,L}$ and $\mathcal{D}_{m,R}$ such that both sub-regions are more kinetically homogeneous than \mathcal{D}_m . The kinetic homogeneity of \mathcal{D}_m is estimated by $\mathcal{H}_{\mathcal{D}_m}$ based on species DRGEP co-efficient vector $\mathbf{R}^{\mathcal{D}_m}$ evaluated using the compositions in \mathcal{D}_m :

$$\mathcal{H}_{\mathcal{D}_m} = \|\mathbf{R}^{\mathcal{D}_m}\|_1 = \sum_{i=1}^{n_s} R_i^{\mathcal{D}_m} \quad (5.4)$$

A smaller value of \mathcal{H} for a set of compositions is associated with lower DRGEP coefficients, and consequently a more reduced kinetic model for that set of compositions. Thus, the objective of each iteration is to identify the hyperplane that minimizes the following cost function:

$$\mathcal{C}_{\mathcal{D}_m} = n_{p,\mathcal{D}_{m,L}} \mathcal{H}_{\mathcal{D}_{m,L}} + n_{p,\mathcal{D}_{m,R}} \mathcal{H}_{\mathcal{D}_{m,R}} = n_{p,\mathcal{D}_{m,L}} \|\mathbf{R}^{\mathcal{D}_{m,L}}\|_1 + n_{p,\mathcal{D}_{m,R}} \|\mathbf{R}^{\mathcal{D}_{m,R}}\|_1 \quad (5.5)$$

where $n_{p,\mathcal{D}_{m,L}}$ and $n_{p,\mathcal{D}_{m,R}}$ are the number of compositions in $\mathcal{D}_{m,L}$ and $\mathcal{D}_{m,R}$, respectively. With this objective function in mind, first the normal vector of the hyperplane is determined by the first principal component of the ensemble of DRGEP coefficients of all compositions in \mathcal{D}_m . Then the exact position of the hyperplane along the direction of the normal vector, or the anchor point is identified by minimizing the cost function in Eq. 5.5. The normal vector and the

anchor point fully define the hyperplane that provides the optimal $\mathcal{D}_{m,L}$ and $\mathcal{D}_{m,R}$. The algorithm moves on with dividing each of these two sub-regions in the next iteration.

4. *Consolidation into N_R regions:* Without any constraint on number of regions, \mathbb{C} is partitioned into $n_{\mathcal{D}}$ regions, each containing a single Φ from \mathcal{D} . To consolidate the number of regions to a user-specified value N_R , the full tree is analyzed and an upper tree with N_R leaves is identified that minimizes the overall cost of the tree:

$$C_{\text{tree}} = \sum_{J=1}^{N_R} n_{p,\mathcal{D}_J} \mathcal{H}_{\mathcal{D}_J} \quad (5.6)$$

Once the optimal upper tree is identified, all hyperplanes below the terminal points of the upper tree are removed to consolidate the partition into N_R regions.

Re-partitioning criteria: During the simulation of interest, the quality of the partition is examined after every n_t^p time steps. If the partition is considered to be below a certain quality, the DPAC algorithm re-partitions the current accessed region of the composition space following the partitioning algorithm (step 1-4). The quality of the partition is also estimated by the concept of kinetic homogeneity. The formulation of \mathcal{H} in Eq. 5.4 based on DRGEP coefficients, by construction, ensures that the largest value of \mathcal{H} is observed in case of a full composition database \mathcal{D} without any sub-regions, and the value of \mathcal{H} of any subset of \mathcal{D}_m , e.g., $\mathcal{D}_{m,L}$, is always less than or equal to $\mathcal{H}_{\mathcal{D}_m}$ [119]. As a result, the maximum cost (the worst case scenario) corresponding to placing all compositions in \mathcal{D} in a single region

will be

$$C_{\max} = C_{\mathcal{D}} = n_{p,\mathcal{D}} \mathcal{H}_{\mathcal{D}} = n_{p,\mathcal{D}} \sum_{i=1}^{n_s} \mathbf{R}_i^{\mathcal{D}}. \quad (5.7)$$

Then we define the quality of a partition, $Q_{\mathcal{P}}$ as:

$$Q_{\mathcal{P}} = \frac{C_{\max}}{C_{\text{tree}}}, \quad (5.8)$$

where C_{tree} is the cost of the current partition or the binary tree. A higher value of $Q_{\mathcal{P}}$ indicates better quality of the current partition. If this ratio, $Q_{\mathcal{P}}$, goes below a user-defined threshold, $\epsilon_{\mathcal{P}}$, the algorithm decides to re-partition the current accessed region. During statistically stationary condition of a simulation, as the accessed region of the composition space does not change significantly, DPAC is expected not to re-partition the accessed region during the surveillance mode. However, during the transient part of the simulation (building mode), multiple re-partitions are typically required before an optimal partition is achieved.

5.6 Results and discussion

We utilize two different configurations, as mentioned before. The 0D PaSR is the first configuration, which is used here to study different features of the DPAC algorithm. The full scope of the automatic DPAC algorithm is explored in case of an LES/PDF simulation of a 2D turbulent flame.

5.6.1 PaSR configuration

The first configuration considered is a non-premixed piloted propane-air PaSR. The detailed propane mechanism used here is from Petersen *et al.* [160] with 115 species and 1308 reactions. The PaSR parameters are those used in [119], with 100 particles and three inflow streams: pure fuel, air and a burnt gas pilot. The PaSR is initially run for three residence times after which it reaches a statistically stationary state. The relevant DPAC inputs used for the PaSR study are summarized here:

- The number of regions is specified to be 10.
- The targets for building models at runtime are chosen to be propane, CO₂, CO, OH, and temperature.
- Each reduced model either at the end of pre-processing stage or after re-partitioning is initialized with 14 species, including the species from inflow streams and the targets.
- The building time step: $\Delta t_b = 2\Delta t$, where Δt is the PaSR time step.
- All compositions, i.e., 100 particles are used to build the models at time t .
- n_t^B is chosen to be 1 and 100 in the building and surveillance mode, respectively.
- The value of n_t^C is 10 in all simulations.
- The DRGEP reaction threshold, ϵ_{th} is varied, and the error threshold, ϵ_c , during building is kept as one order of magnitude lower than ϵ_{th} .

- The threshold, ϵ_p , for the quality of a partition Q_p is not very intuitive to guess. Therefore, first the variation of Q_p is explored, and then based on that analysis, an ϵ_p is chosen and used for all the DPAC runs.
- n_i^p is chosen to be 100 and 200 in the building and surveillance mode, respectively.
- To compare the impact of the initial database and partition, three different initial databases (\mathcal{D}^0) are used in the pre-processing stage. The simulations used to generate those databases are:
 - (1) \mathcal{D}_1^0 : detailed PaSR,
 - (2) \mathcal{D}_2^0 : 0D isobaric homogeneous reactor including both low and high temperature chemistry,
 - (3) \mathcal{D}_3^0 : 0D isobaric homogeneous reactor with only low temperature chemistry.

Re-partition criteria

Finding the right criteria for re-partitioning is extremely important in DPAC algorithm. As the compositions evolve during the transient stage of a simulation, re-partitioning at almost every check (after every n_i^p time steps) leads to a better partition than the previous one. However the cost of re-partitioning and subsequent rebuilding of models from scratch are computationally expensive. The DPAC algorithm only works if we need to re-partition and rebuild only a few times to reach the optimal partition and models, so that the cost of them becomes

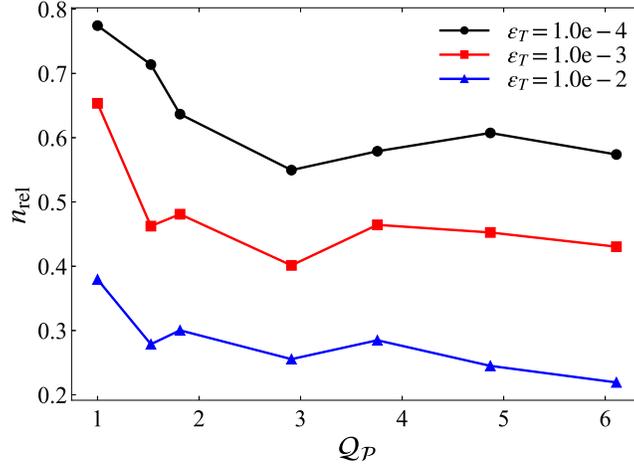


Figure 5.2: Relative number of species as a function of the quality of partition for three different incurred error in temperature.

insignificant compared to the overall simulation. Consequently, the right criterion ensures that a re-partition is performed only when the quality of the current partition is extremely poor for the current accessed region, and re-partitioning may provide a significant gain in the long run. The quality of a partition, Q_p (Eq. 5.8) indicates how good a partition is compared to the case where all compositions exist in one region. To figure out a threshold in Q_p, ϵ_p , we followed these steps:

- We evaluate the cost of a partition or a tree, C_{tree} (Eq. 5.5), based on the compositions in \mathcal{D}_1^0 . In other words, we will pick several partitions or trees based on random databases, and evaluate the cost of it in the context of \mathcal{D}_1^0 .
- The first tree we choose corresponds to the optimal partition, i.e., the tree generated using \mathcal{D}_1^0 itself. This tree should have the minimum cost, thus largest Q_p , denoted by Q_p^{max} .

- The worst tree would be the one where all compositions from \mathcal{D}_1^0 are classified into one region. The cost of that tree would be the maximum, and by definition, $Q_p = 1$ for that tree.
- Any tree other than the optimal and the worst one should have a Q_p value between 1 and Q_p^{\max} . We randomly pick five trees based on different composition databases, which cover the whole range of Q_p .
- We run the DPAC adaptive PaSR simulations using each of the seven different trees mentioned above, and compare their accuracy with respect to the PaSR run using detailed mechanism.

To compare the quality of each partition, the relative number of species used in each adaptive simulation is plotted against the quality of each tree in Fig. 5.2. The relative number of species used throughout the simulation is defined as:

$$n_{\text{rel}} = \frac{1}{n_t n_p n_s} \sum_{k=1}^{n_t} \sum_{n=1}^{n_p} n_{s,k}^{(n),A}, \quad (5.9)$$

where n_s is the number of species in the detailed model, and $n_{s,k}^{(n),A}$ is the number of species in the reduced model used by particle n at time step k . The incurred error for a quantity X is defined as:

$$\varepsilon_X = \frac{\sum_{k=1}^{n_t} \sum_{n=1}^{n_p} |X_k^{(n),A} - X_k^{(n),D}|}{\sum_{k=1}^{n_t} \sum_{n=1}^{n_p} |X_k^{(n),D}|}, \quad (5.10)$$

where n_t is the number of time steps and n_p , the number of particles. $X_k^{(n),A}$ and $X_k^{(n),D}$ represent the values of quantity X for particle n at the k -th time step using the adaptive and detailed chemistry, respectively. The figure shows the variation

of n_{rel} for three different incurred errors in temperature, i.e., 10^{-2} , 10^{-3} and 10^{-4} . Those different incurred errors for a given tree are obtained by varying the DRGEP reaction coefficient threshold, ϵ_{th} , in the DPAC runs.

In all three curves, the left most and the right most points correspond to the lowest quality tree, i.e., $Q_{\mathcal{P}} = 1$, and the best quality tree, i.e., $Q_{\mathcal{P}} = Q_{\mathcal{P}}^{\text{max}}$, respectively. As expected, if all compositions are classified to one region, DPAC requires a larger number of species to achieve a certain ϵ_T than any other trees. However, beyond a $Q_{\mathcal{P}}$ value in between 1 and 2, n_{rel} does not vary significantly even if the quality of the partition increases, consequently the average cost of integrating the chemistry source term will not vary significantly. On the other hand, choosing a higher value of $Q_{\mathcal{P}}$ than 2 as $\epsilon_{\mathcal{P}}$ will only increase the required number of re-partitions and rebuilding of models. In other words, if a partition, say of $Q_{\mathcal{P}} = 4$, is allowed to be re-partitioned, the new partition may have a $Q_{\mathcal{P}}$ higher than 4, but without affecting n_{rel} and the chemistry integration cost. Therefore, we choose $\epsilon_{\mathcal{P}} = 2$ to be the threshold value of $Q_{\mathcal{P}}$, below which the current accessed region of the composition space will be re-partitioned.

A couple of important points need to be mentioned here. First, in case of $\epsilon_T = 10^{-3}$ and 10^{-4} , the partition corresponding to a quality of 3, requires slightly smaller n_{rel} than the optimal partition. This may have happened due to several reasons: the nature of the simulation, the non-linearity involved in the definition of error or in combustion chemistry itself. It may change for another set of compositions. Overall, we can say that the change in n_{rel} is minimal beyond $Q_{\mathcal{P}} = 2$.

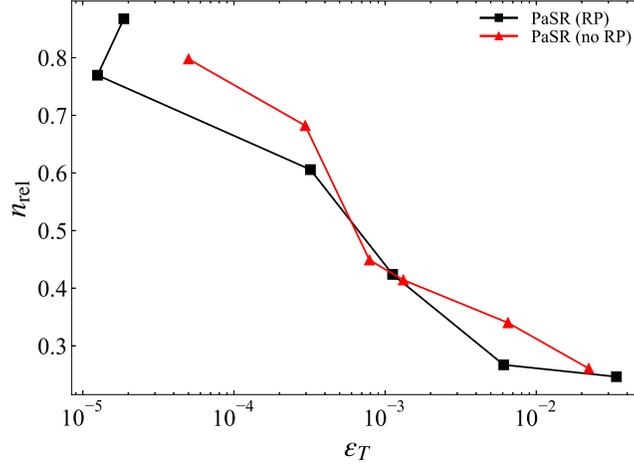


Figure 5.3: Relative number of species as a function of incurred error in temperature for the initial database \mathcal{D}_1^0 (detailed PaSR) with re-partitioning (black squares) and without re-partitioning (red triangles).

Second, and more importantly, this threshold, like any threshold, is not necessarily the optimal choice for any set of compositions from any simulation. Before running an LES/PDF simulation, a similar test with compositions from a PaSR run (same inflow streams as the LES/PDF) would give a better idea of the threshold value. However, we can conclude from the previous discussion that a value closer to unity than closer to Q_ρ should be the ideal choice, thus the threshold is not expected to differ a lot from the threshold chosen here, i.e., $\epsilon_\rho = 2$.

Importance of re-partition and dependence on initial database

To emphasize the importance of re-partition, we compare the incurred error in temperature with and without re-partitioning for all three initial databases as a function of n_{rel} (Eq. 5.9) used in the kinetic models in each case. The runs without

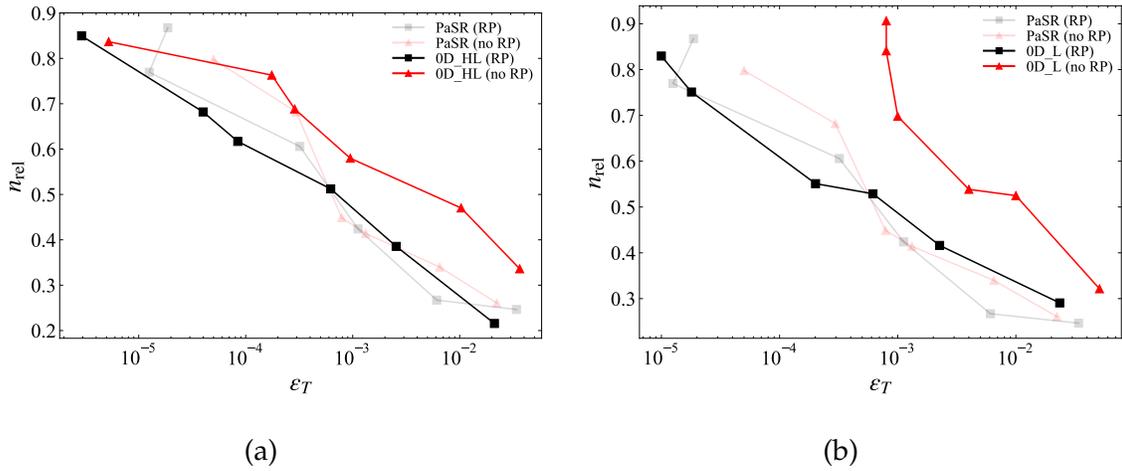


Figure 5.4: Relative number of species as a function of incurred error in temperature for the initial database (a) \mathcal{D}_2^0 (0D runs with high and low temperature chemistry) and (b) \mathcal{D}_3^0 (0D runs with only low temperature chemistry) with re-partitioning (black squares) and without re-partitioning (red triangles). The transparent lines correspond to the same two curves from Fig. 5.3

re-partitioning follows the DPAC algorithm except it keeps the initial partition from the pre-processing stage throughout the simulation.

The comparison for the initial database coming from a detailed PaSR, \mathcal{D}_1^0 , is shown in Fig. 5.3. We observe from both curves, as expected, an increase in incurred error for temperature as the relative number of species decreases, that is, as the region-specific models become more reduced. A more important observation is that both curves are very similar, indicating that a similar level of reduction will lead a similar error in temperature prediction even if we do not re-partition. The compositions in \mathcal{D}_1^0 from a detailed PaSR run are expected to be very similar to the compositions encountered during the adaptive run, as both of them are the exact same simulation configuration. Therefore the initial partition from the

pre-processing stage is expected to divide the accessed region of the composition space optimally, leading to a very similar required n_{rel} for a given ϵ_T .

However, the trend changes noticeably, when looking at the same comparison in Fig. 5.4a and 5.4b for \mathcal{D}_2^0 and \mathcal{D}_3^0 , respectively. The translucent lines, corresponding to the curves from Fig. 5.3, are shown in both plots for a succinct comparison. For both databases coming from 0D isobaric runs, the relative number of species required to reach a given error in the prediction of temperature is observed to be significantly smaller with re-partitioning than without. As the accessed region of compositions from the 0D runs are different from PaSR compositions, re-partition based on the compositions encountered at runtime always increases the efficiency.

As mentioned before, the idea of DPAC is to eliminate the impact of the initial database and its corresponding partition by adaptively adjusting to the current accessed region of the composition space during the simulation of interest. Consequently, it is expected that the incurred error as a function of n_{rel} would exhibit similar trend in case of any initial database for DPAC. To confirm that, all three DPAC curves ('RP' lines from Fig. 5.3 and 5.4) corresponding to three initial databases are plotted in Fig. 5.5. As expected, all three curves overlap, which means to achieve a certain error in temperature in a PaSR run, the effective number of species required is independent of the initial database used in the pre-processing stage.

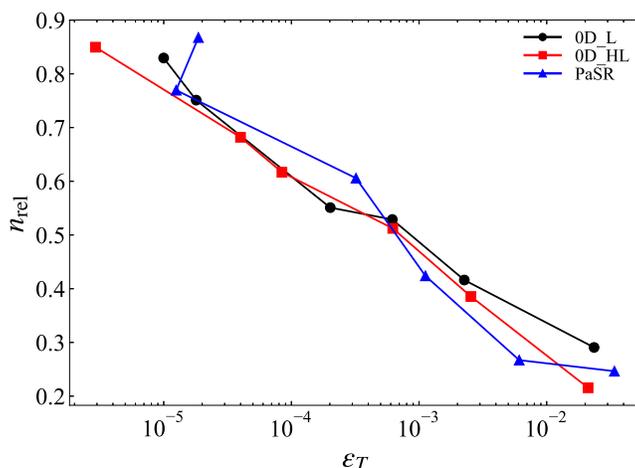


Figure 5.5: Relative number of species as a function of incurred error in temperature during DPAC runs with all three initial databases. Clearly, the accuracy of DPAC is independent of the initial database.

Comparison with pre-partitioned adaptive chemistry

Similar to DPAC, pre-partitioned adaptive chemistry (PPAC) also consists of two stages: a pre-processing stage, and an adaptive stage. During the pre-processing stage, a partition is obtained based on an initial database, and then reduced models are generated using traditional top-down DRGEP [51] methodology for each region. In the adaptive stage, before the reaction step, each particle is first assigned to a region based on the classification algorithm, and then advanced in time using the reduced model generated for that region in the pre-processing stage. The details of the PPAC algorithm is available in [119]. The initial database in the pre-processing stage is an important component of the PPAC algorithm, as it must be representative of the actual compositions encountered at runtime for an accurate adaptive simulation. In this section, we compare the accuracy of PPAC and DPAC

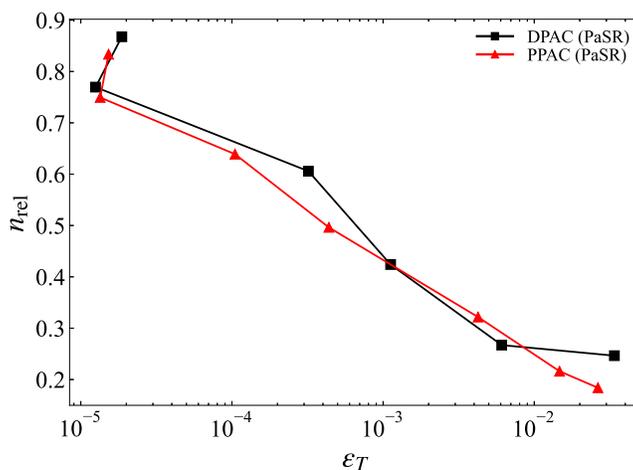


Figure 5.6: Relative number of species as a function of incurred error in temperature for DPAC (black squares) and PPAC (red triangles), both with initial database, \mathcal{D}_1^0 (detailed PaSR).

algorithm in the context of all three initial databases.

The first comparison is performed for \mathcal{D}_1^0 , which is obtained from detailed PaSR, and plotted in Fig. 5.6. As previously explained, because the detailed PaSR compositions are true representative of the compositions encountered during adaptive PaSR simulation, or in other words their accessed regions are similar, the PPAC algorithm exhibits very similar results compared to DPAC. Both of them require similar relative number of species to achieve a given incurred error in temperature. The idea of DPAC is more relevant when a set of initial compositions which is representative of the actual compositions is not easy to guess or rather guarantee. For such cases, the performance of PPAC drops significantly. Figure 5.7a shows the same comparison for \mathcal{D}_2^0 . Clearly, to achieve an incurred error less than 10^{-4} , PPAC requires more than 80% of the species in the detailed

mechanism. The efficiency drops even further in case of \mathcal{D}_3^0 (Fig. 5.7b). As \mathcal{D}_2^0 covers a more extensive set of 0D simulations than \mathcal{D}_3^0 , PPAC works slightly better in case of the former, as can be seen in Fig. 5.7.

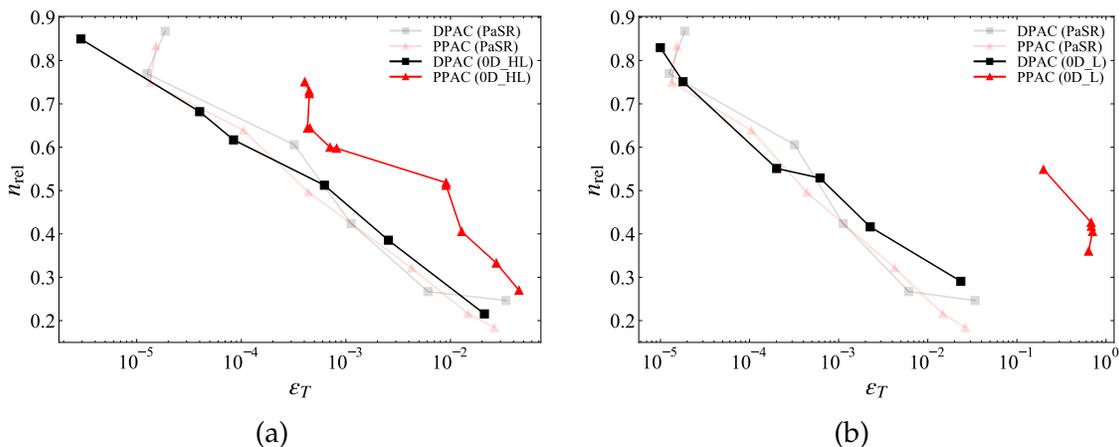


Figure 5.7: Relative number of species as a function of incurred error in temperature for DPAC (black squares) and PPAC (red triangles), both with initial database, (a) \mathcal{D}_2^0 (0D runs with high and low temperature chemistry) and (b) \mathcal{D}_3^0 (0D runs with only low temperature chemistry).

Features of DPAC

To emphasize on different characteristics of DPAC, such as building the models and re-partitioning, we focus on two full PaSR runs using DPAC, and compare them with the PaSR run using detailed kinetic model. These two DPAC runs correspond to two different reaction DRGEP threshold, ϵ_{th} , used during building the models, i.e., 10^{-2} and 10^{-4} . The evolution of temperature for both DPAC runs and the detailed run are plotted in Fig. 5.8 (top). Clearly, both DPAC runs are indistinguishable from the detailed simulation. To differentiate the DPAC runs,

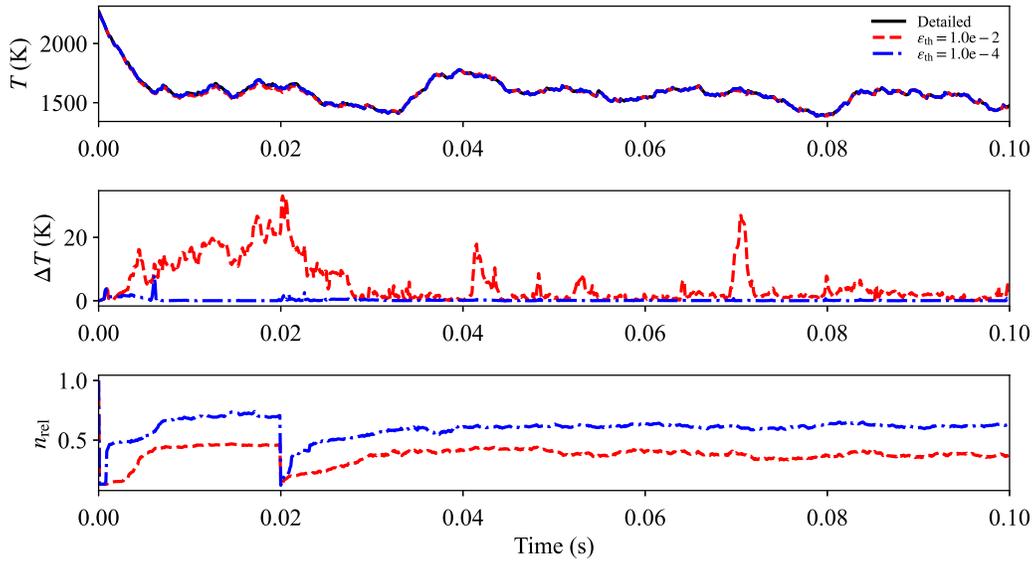


Figure 5.8: (a) The evolution of mean temperature in a PaSR with detailed mechanism (black solid line), DPAC with $\epsilon_{th} = 10^{-2}$ (red dashed line) and $\epsilon_{th} = 10^{-4}$ (blue dash dot line), (b) The absolute difference in mean temperature between detailed run and two DPAC runs, (c) Relative number of species used at each time step in case of both DPAC runs.

the middle plot of Fig. 5.8 shows the absolute difference in temperature between the detailed and the DPAC runs. The evolution of the relative number of species, n_{rel} , used at each time step for both DPAC runs is plotted in the bottom figure. The n_{rel} evaluation is similar to Eq. 5.9, except it is only averaged over all particles at each time step.

There are three important observations from these three plots in Fig. 5.8. First, as ϵ_{th} drops from 10^{-2} to 10^{-4} , the building algorithm adds more species in the reduced models, resulting in a higher n_{rel} (bottom plot) at every time step. A higher n_{rel} leads to better accuracy, as expected. Thus, the difference in temperature (middle plot) is lower for $\epsilon_{th} = 10^{-4}$. This difference in ΔT over the full PaSR run be-

tween two ϵ_{th} values results in a difference of more than two orders of magnitude in ϵ_T (Eq. 5.3). Secondly, the re-partition in both DPAC runs occurs at $t = 0.02$ seconds (after 2 residence times), where the models are re-initialized, and as a result n_{rel} drops to the initial number of species. Due to re-initialization of models and the absence of important species and reactions in the next few time steps, the difference in temperature reaches a maximum near $t = 0.02$ seconds for $\epsilon_{\text{th}} = 10^{-2}$, which then drops again as the models are sufficiently built. Finally, as the simulation reaches a statistically stationary state, DPAC does not need to re-partition any more, neither do the models update significantly (n_{rel} remains constant in both DPAC runs).

5.6.2 LES/PDF configuration

The configuration simulated in this study is similar to the flame studied experimentally in [165] and numerically in a 2D DNS [154]. The configuration consists of a n-dodecane liquid jet injected with 16% oxygen concentration by volume at an initial pressure of 3.4 MPa and temperature of 800 K. Similar to Tagliante *et al.* [154], the computational domain is chosen to be 20 mm downstream of the liquid injector, thus neglecting the liquid spray simulation completely and only simulating the reactive zone with gaseous reactions. There are three major modifications made in the current configuration compared to the study in [154]. First, here the jet diameter (D) is 20 mm, where the inflow velocity follows the same velocity profile as [154], but with a co-flow velocity of 10 m/s (as opposed to 1 m/s) and

a maximum axial velocity of 71 m/s (as opposed to 80 m/s). Outside the jet area, there is a uniform co-flow of 10 m/s. Secondly, the temperature profile at the inlet follows a step function, i.e., the jet stream and co-flow are at 600 K and 800 K, respectively. A similar step function for the inflow compositions is used as well. The inflow compositions are tabulated in table 5.1. Third, we perform a 2D LES/PDF simulation with DPAC algorithm and compare it with either PPAC or the detailed mechanism, as opposed to their 2D DNS case with a detailed mechanism. The objective of this study is not to replicate or compare with the simulation in [154], rather it is to check the efficiency and accuracy of the DPAC algorithm, and compare the computational cost with using PPAC or a detailed mechanism.

Table 5.1: Inlet species mass fractions.

	N ₂	O ₂	CO ₂	H ₂ O	nC ₁₂ H ₂₆
Co-flow	0.7016	0.1746	0.1001	0.0237	0
Jet	0.5943	0.1479	0.0848	0.02	0.153

The computational domain is $20D$ long in both axial and radial directions. A non-uniform 128 (axial) \times 128 (radial) mesh is used. The mixing model constant is specified to be 4, and the number of particles per cell is chosen to be 20. A reduced n-dodecane mechanism [154] with 47 species and 198 reactions is used as the detailed model in this study. The relevant DPAC inputs (relevant ones are used in PPAC as well) are listed below:

- The number of regions is specified to be 10.
- The targets are chosen to be n-dodecane, CO₂, CO, OH, CH₂O and tempera-

ture for building models at runtime. The introduction of CH_2O is important to capture the low temperature chemistry, as observed in [154].

- Each reduced model either at the end of pre-processing stage or after re-partitioning is initialized with 8 species, including the species from inflow streams and the targets, which corresponds to 17% of the detailed model.
- The first two and the next five flow-through times are considered to be the building and surveillance mode, respectively.
- The building time step: $\Delta t_b = 10\Delta t$, where Δt is the CFD time step.
- A set of downsampled compositions (1000 particles) are used to build the models at time t .
- $n_i^{\mathcal{B}}$ is chosen to be 500 and 1000 in the building and surveillance mode, respectively.
- The value of $n_i^{\mathcal{C}}$ is 10.
- The DRGEP reaction threshold, ϵ_{th} is 10^{-2} .
- To capture the auto-ignition phenomenon and low temperature chemistry (LTC) accurately, an indicator, Y_{LTC} , is introduced here, and defined by $Y_{\text{LTC}} = Y_{\text{C}_{12}\text{H}_{25}\text{O}_2} + Y_{\text{O}_2\text{C}_{12}\text{H}_{24}\text{OOH}} + Y_{\text{OC}_{12}\text{H}_{23}\text{OOH}} + Y_{\text{C}_{12}\text{OOH}}$, where these 4 species in the detailed mechanism correspond to the low temperature pathways. Unlike the previous PaSR case, in the LES/PDF simulation, the error threshold, ϵ_c , is dynamically chosen. If more than 10% of the compositions in a region have $Y_{\text{LTC}} > 10^{-4}$, we consider ϵ_c to be 10^{-10} , otherwise it remains 10^{-3} , which is one order magnitude lower than ϵ_{th} . As the change in composition of a particle over a CFD time step

at LTC is always very small until it ignites, such a small ϵ_c is important to capture LTC, otherwise the building algorithm does not add necessary reactions, and the flame does not ignite.

- Based on the previous analysis in case of PaSR, ϵ_p is chosen to be 2.
- A set of downsampled compositions (5000 particles) are used to check the quality of the current partition, and to re-partition, if required.
- n_i^p is chosen to be 200 and 1000 in the building and surveillance mode, respectively.
- To compare the impact of the initial database and partition, three different initial databases (\mathcal{D}^0) are used in the pre-processing stage. The simulations used to generate those databases are:
 - (1) \mathcal{D}_1^0 : detailed LES/PDF simulation,
 - (2) \mathcal{D}_2^0 : detailed PaSR simulation with two streams similar to the current configuration.
 - (3) \mathcal{D}_3^0 : 0D isobaric homogeneous reactor with only low temperature chemistry.

These three databases are only used in PPAC runs. In case of DPAC, we only use \mathcal{D}_2^0 in the pre-processing stage, as DPAC is independent of the initial database and partition.

The difference between the adaptive simulations (DPAC or PPAC) and the detailed simulations is quantified by the normalized root mean square difference

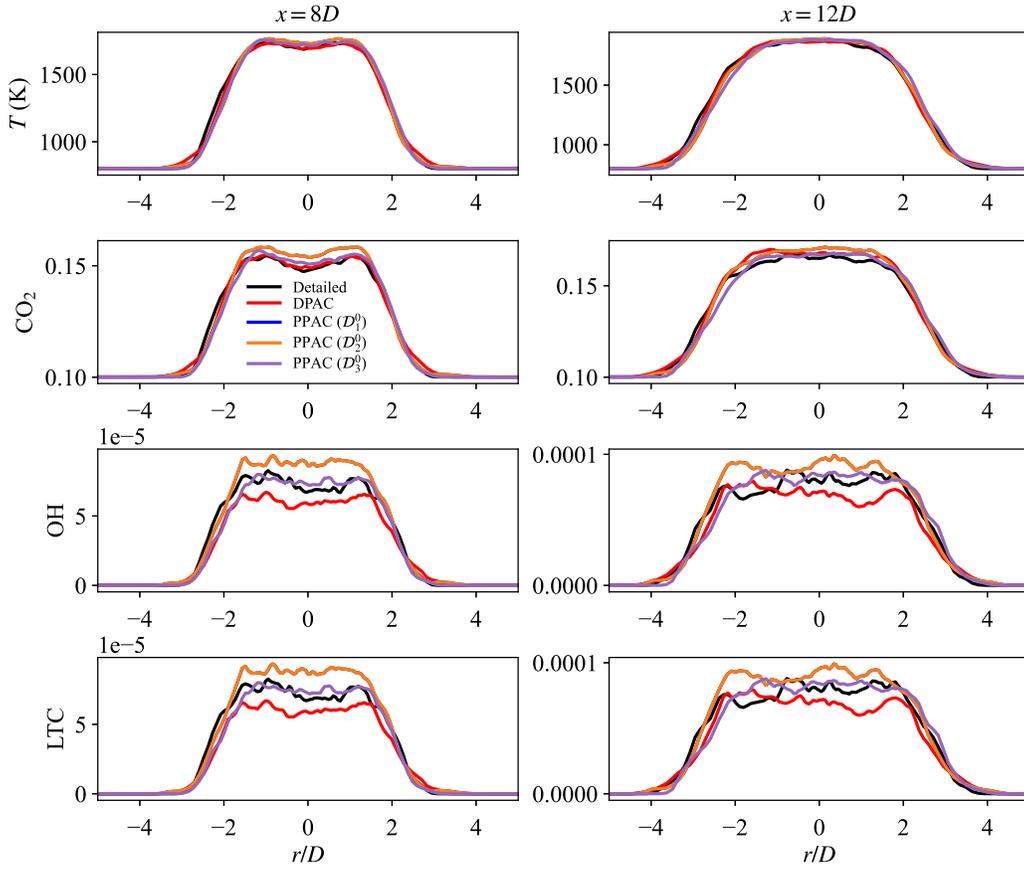


Figure 5.9: Comparison of resolved mean of temperature and species mass fractions of CO_2 , OH and LTC between LES/PDF with detailed mechanism and adaptive runs corresponding to DPAC and PPAC with three different databases.

(RMSD) [39], defined as:

$$\mathcal{E}(\xi) = \frac{[\xi^A - \xi^D]_{rms}}{\xi_{ref}}, \quad (5.11)$$

where ξ^A and ξ^D denote the quantities obtained from adaptive and detailed simulation, respectively, ξ_{ref} is the reference value of each quantity of interest. ξ^{ref} is considered to be 1000K for temperature, and the maximum species mass fraction among all the radial profiles at both axial locations: $8D$ and $12D$ (D is the burner

diameter). The $[\cdot]_{rms}$ is computed over all the radial locations at all the considered axial locations. The simulations with RMSD value below 3% in resolved mean of temperature are chosen for the subsequent analysis of performance and accuracy in case of both PPAC and DPAC. Figure 5.9 compares the radial profiles of the resolved mean of temperature and mass fractions of CO_2 , OH and LTC obtained from the detailed, DPAC and PPAC (based on all three databases) simulations at two different axial locations, $8D$ and $12D$ from the burner exit. Statistics have been collected for five flow-through times (surveillance mode) evaluated based on the co-flow velocity.

Table 5.2: Normalized RMSD (%) of resolved mean quantities between LES/PDF using detailed mechanism and DPAC or PPAC. The relative number of species required to achieve those RMSD errors are provided in the last row. All the quantities are evaluated over five flow-through times in their statistically steady period.

Quantity (ξ)	DPAC	PPAC (\mathcal{D}_1^0)	PPAC (\mathcal{D}_2^0)	PPAC (\mathcal{D}_3^0)
$\mathcal{E}(T)$	2.24	2.89	1.98	1.88
$\mathcal{E}(\text{CO}_2)$	1.36	2.12	1.45	1.44
$\mathcal{E}(\text{OH})$	8.73	10.73	7.21	7.01
$\mathcal{E}(\text{LTC})$	9.67	11.76	8.69	7.83
n_{rel}	0.43	0.42	0.96	0.98

The RMSD errors of a few important species mass fractions (plotted in Fig. 5.9) and temperature are provided in Table 5.2. The table also provides the relative number of species, n_{rel} (Eq. 5.9), required for all the adaptive runs to achieve the RMSD of 3% in temperature. DPAC and PPAC with \mathcal{D}_1^0 reach similar level of accuracy using approximately 40% of the number of species in the detailed model. As \mathcal{D}_1^0 is generated from a detailed LES/PDF simulation of the same configuration, it is expected that both PPAC and DPAC would exhibit similar performance and

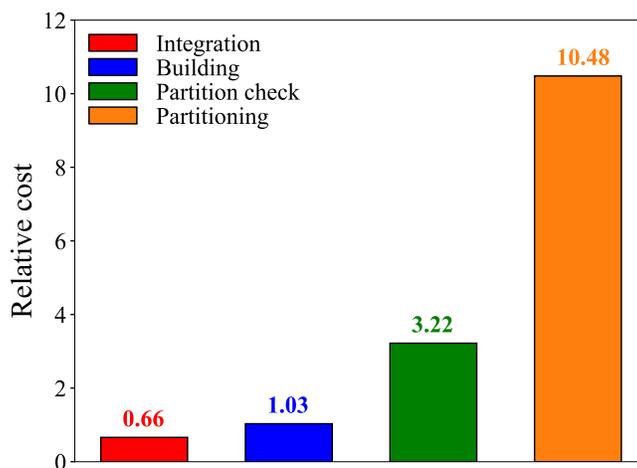


Figure 5.10: Relative computational cost of one time-step of individual components of DPAC: chemistry source term integration, building models, partition quality check and re-partitioning with respect to the cost of chemistry source term integration with detailed mechanism.

accuracy. However, for the other two databases, when the partition is not optimal, PPAC requires almost all the species from the detailed mechanism to achieve the same level of accuracy. An RMSD error of more than 10% in OH and LTC for PPAC with \mathcal{D}_1^0 might be considered unacceptable even though the error in temperature is below 3%. In such a case, a larger set of reduced models is required to improve the level of accuracy, which will be verified in future.

From the analysis on accuracy of LES/PDF simulation and the overall PaSR comparison in the previous section, DPAC is found to provide accurate solution using much smaller relative number of species, n_{rel} , compared to PPAC. However, n_{rel} only indicates the cost of chemistry source term integration. Unlike PPAC, in case of DPAC, the adaptive run consists of components other than the integration, such as building models and re-partitioning. Even when we do not re-

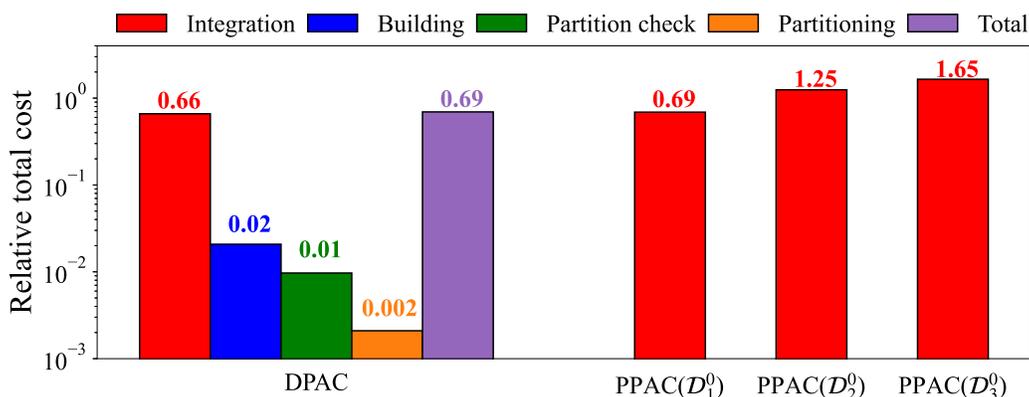


Figure 5.11: Relative total computational cost of each of the components of DPAC and overall DPAC, and PPAC with three different databases with respect to the total cost of detailed simulation during the building mode.

partition, we keep checking the quality of the partition, Q_p . Thus, a cost analysis of each of these components is extremely important to appreciate the essence of DPAC. Figure 5.10 shows the cost of each of these components relative to the cost of chemistry source term integration with a detailed mechanism. As n_{rel} is significantly smaller than unity in DPAC, the relative cost of integration is 66% of the detailed integration. However, the average CPU cost of each building step is approximately same as the detailed integration. The partition quality check and re-partitioning are approximately 3 and 10 times more expensive than a detailed integration step, respectively. Although the cost of individual building or re-partition is large, DPAC builds the models or re-partitions only a few times before it reaches an optimal partition and set of reduced models. For instance, in this particular DPAC run, the composition space is re-partitioned only 5 times during the building mode of the simulation, and then it does not change during the surveillance mode.

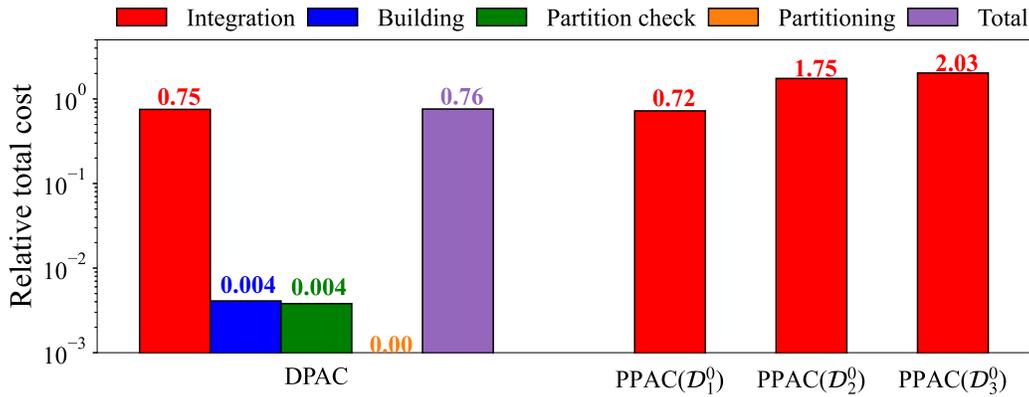


Figure 5.12: Relative total computational cost of each of the components of DPAC and overall DPAC, and PPAC with three different databases with respect to the total cost of detailed simulation during the surveillance mode.

The total cost of all the individual components of DPAC relative to the total cost of detailed simulation during the building and surveillance mode are shown in Fig. 5.11 and 5.12, respectively. Both plots also compare the relative cost of each of the PPAC runs based on three different databases. There are a few important observations from these two plots. First, during the building mode, the total cost is 69% of the detailed simulation, and its counterpart during the surveillance mode takes about 77%. Therefore, overall we observe a 31% and 23% reduction in CPU cost using DPAC compared to the detailed simulation. Second, the cost of PPAC with \mathcal{D}_1^0 is similar to the cost of DPAC, as expected. But for the other two PPAC runs, the cost is even higher than a detailed simulation. This can be attributed to the fact that an adaptive simulation includes the additional cost of classification, conversion to reduced representation and reconstruction of the mass fraction vector for each particle at each time step. As both PPAC (\mathcal{D}_2^0 and \mathcal{D}_3^0) runs with nearly detailed mechanism, the overall cost becomes larger

than the detailed integration. Third, most of the model update, partition quality check and re-partitioning occur during the building mode, as overall they take about 3% cost of the simulation. Re-partitioning, which is the most expensive step (Fig. 5.10), takes only 0.2% of the cost during building mode. As the frequency of model update and partition quality check drop during surveillance mode, they only take about 1% of the total cost.

Figure 5.13 shows the instantaneous particle distribution of temperature and index of the model used by each particle for reaction integration step at four different time instants of the building mode. The final snapshot is at $t = 0.01$ s, which corresponds to one-third of a flow-through time. During this time, the jet mixes with the co-flow, the LTC pathways are activated near the mixing layer, a few particles start to ignite, and eventually the flame evolves. This series of figures exhibit the ability of DPAC to adapt to the evolving accessed region of the composition space as it re-partitions and rebuilds the models multiple times during the building mode. In Fig. 5.13a, we observe all the particles are classified into two regions, one corresponding to the co-flow and the other to the jet compositions. Soon after that, in Fig. 5.13b, as the jet and co-flow compositions mix, the accessed region of the composition space is re-partitioned and a third region emerges, which is allocated for the mixed compositions. After this re-partition, the model corresponding to the new region starts to grow to capture the LTC. The other two regions corresponding to jet and co-flow does not add any reactions as they contain the non-reactive compositions. Then in Fig. 5.13c, a few particle compositions ignite and exhibit high temperature kernels, and the accessed region grows as both low

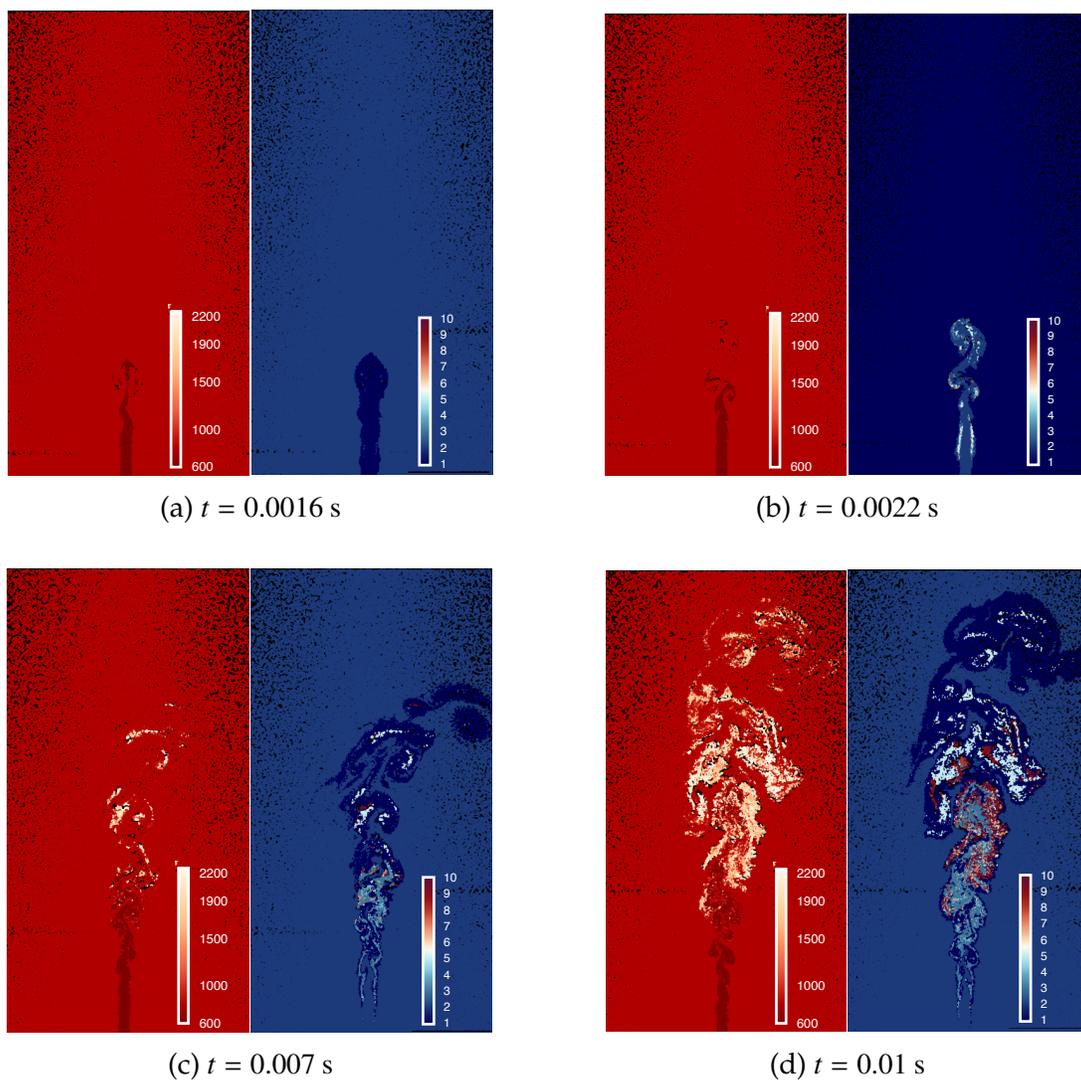


Figure 5.13: Instantaneous particle distribution colored by temperature and index of the model used at four different time instants of the building mode. (a) Towards the beginning, only two models are being used for jet and co-flow. (b) As the jet and co-flow compositions mix, the accessed region is re-partitioned into 3 regions, the third one being allocated for the mixed compositions, which captures the LTC. (c) A few ignition kernels start to appear. There is enough variance in the composition space for DPAC to use ten models, (d) The partition keeps using all models as the flame evolves, and finally finds the optimal partition.

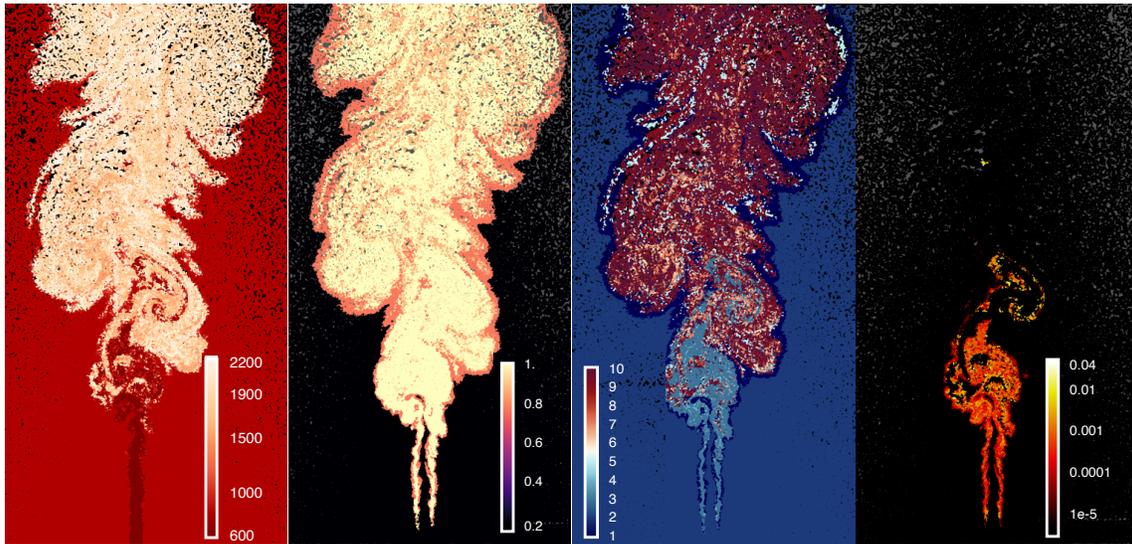


Figure 5.14: Instantaneous particle distribution colored by temperature, relative number of species being used by each particle, index of the model used for reaction source term integration and mass fraction of LTC from left to right, respectively, during surveillance mode.

and high temperature pathways are activated. As a result, the quality of the current partition from Fig. 5.13b deems to be poor according to the re-partitioning criteria, and DPAC re-partitions the composition space. This time all non-reactive compositions from both jet and co-flow are allocated to a single region (model 2 Fig. 5.13c and 5.13d), and all the important LTC compositions are classified into another (model 3 Fig. 5.13c and 5.13d). Additionally, as the flame evolves, there is a separate region (model 1 in Fig. 5.13c and 5.13d) allocated for the compositions near the outer shear layer of the flame, where non-reactive compositions mix with the reactive low or high temperature compositions. After $t = 0.01$ s, the partition does not change in either building or surveillance mode.

The instantaneous particle distributions of temperature, relative number of species required for each particle to advance in time, the model each particle is assigned to and Y_{LTC} are shown in Fig. 5.14 after the flame has reached a statistically stationary state. The maximum Y_{LTC} compositions are assigned to a separate model (model 3 in this case), as mentioned before. There are two important observations here. First, apart from model 2, which corresponds to the non-reactive compositions, all other models require almost all species from the detailed mechanism. This can be attributed to the fact that the detailed mechanism contains only 47 species, which is an extremely reduced model for a fuel such as n-dodecane including LTC pathways. Even removing a few species results in flame extinction for such a reduced model. A better detailed model will definitely exhibit more variability among the reactive compositions, leading to a better speed up. Second, DPAC does not add any reactions in region 2. Consequently, the non-reactive compositions are integrated using a reduced model with the initial 8 species (17% of the detailed model) and no reactions. The reduction in CPU cost observed in DPAC (or in PPAC with \mathcal{D}_1^0) is a result of not adding any additional species for the non-reactive compositions, which is more than 60% of all compositions, in the computational domain.

5.7 Conclusions

A novel adaptive chemistry algorithm, referred as dynamically partitioned adaptive chemistry (DPAC), especially tailored for particle PDF methods is introduced in this study. This algorithm starts with a partition and a set of initial models, assembled *a-priori*, then adaptively adjusts both the partition and the reduced models based on the compositions encountered during the simulation of interest. Each particle is assigned to one of the regions in the partition and its corresponding reduced model during the adaptive run. The algorithm includes two steps:

- In the pre-processing stage, based on a composition database, the composition space is partitioned into user-specified number of regions, and each region is initialized with a reduced model including a few species and no reaction. The initial composition database does not need to be representative of the compositions we may encounter during the actual simulation.
- In the adaptive stage, based on the current set of compositions observed at runtime, each model is augmented by selectively chosen reactions from the detailed mechanism using a building algorithm. As the initial partition is not optimal for the current accessed region of the composition space, DPAC repartitions the composition space multiple times at runtime, and rebuilds the models from scratch. By the time the flame reaches a statistically stationary state, we have an optimal partition and a set of reduced models, which are not significantly modified any further. All the particles are advanced in time using

the most updated reduced model at that time corresponding to the region it is assigned to.

The efficiency and accuracy of DPAC is demonstrated for a propane/air non-premixed partially stirred reactor and a 2D LES/PDF of lifted dodecane flame. The most important feature of DPAC is that it is independent of the initial database used in the pre-processing stage. Consequently, the initial database can be generated using less expensive low dimensional simulations, such as 0D auto-ignition. The cost of different components of DPAC such as building models and re-partitioning is observed to be insignificant compared to the overall cost of the simulation. In case of LES/PDF simulation, we observe a speed up of 24% (Fig. 5.12) compared to the same simulation with detailed mechanism. The speed up is expected to be larger with larger detailed mechanisms. The performance of DPAC is also compared to a recently developed pre-partitioned adaptive chemistry (PPAC) algorithm. The key disadvantage of PPAC is its dependence on the initial database used in the pre-processing stage. The results show that the performance of PPAC with a proper initial database is comparable with DPAC. However, for a different initial database that is not representative of the compositions at runtime, the performance of DPAC is significantly better than PPAC.

CHAPTER 6

CONCLUSIONS

6.1 Summary

The analytical Jacobian is an important component of any reactive flow simulation to accelerate the stiff chemistry solver. In this work, a unifying analytical framework to formulate Jacobian matrices has been developed, which can handle a state vector comprising all species mass fractions and temperature without violating the constraint that the sum of mass fractions must be unity. To keep the mass fraction vector evaluation at all times during any reactive flow simulation in the realizable simplex, defined by the mass conservation constraint, we use a projection matrix directly in the analytical Jacobian evaluation. Three different projections have been investigated for this purpose. The benefits and drawbacks of each option have been examined and compared in the context of three different configurations: an auto-ignition model, an auto-ignition with mixing model, and PaSR. In terms of the individual boundedness, the projection that normalizes the mass fractions exhibits better accuracy than other projections, whereas the orthogonal projection is the least accurate. The projection approach also predicts the spectral properties of the exact Jacobian matrix, which is important in the context of CSP or CEMA. The analytical Jacobian with all mass fractions is also able to provide accurate solutions of the governing equations not only with Newton-Krylov solvers, such as DVODE, but also linear implicit solvers, such as

Rosenbrock solvers.

The concept of Analytical Jacobian is extended for a reduced kinetic mechanism with quasi-steady species (QSS) assumption. The QSS species are solved using algebraic expressions as a function of all other non-QSS species mass fractions. As a result, the analytical Jacobian needs to take care of the extra derivatives appearing from those algebraic expressions. The QSS species is assumed to have a linear relationship with other QSS, neglecting any quadratic relations, which increases the complexity of the formulation and the computational cost. The algorithm developed here provides all necessary subroutines pertinent to the QSS concentration expressions, and can be easily integrated with any reactive flow simulation. The algorithm is verified in a PaSR simulation with a 47 species dodecane mechanism, out of which 18 species are considered to be QSS. The CPU cost of using an exact analytical Jacobian is observed to 20% less compared to using a numerical Jacobian, as the solver needs to evaluate the source term significantly less number of times in case of analytical Jacobian compared to the numerical one.

The pre-partitioned adaptive chemistry methodology (PPAC) method [119] has been proposed recently to mitigate the computational cost of using particle PDF methods. In this work, we investigate the ability of the PPAC algorithm in predicting the secondary chemistry pathways along with the main combustion chemistry. We observed that PPAC provides a very conservative set of reduced models while trying to capture the secondary chemistry as it adds a lot of unimportant oxidation species before adding necessary secondary pathways. Hence a

PPAC-Additive algorithm is proposed, where the reduced models are generated in two steps: first the reduced models are created only for oxidation targets using a traditional top-down DRGEP algorithm, and then in the second stage, all these reduced models are augmented with relevant secondary chemistry pathways using a recently developed building algorithm [59]. The PPAC-Additive algorithm is verified for a PaSR simulation using a 140 species propane oxidation model including NO_x chemistry. To gain a 3% accuracy in NO prediction, PPAC-Additive showed a 23% reduction in wall clock time compared to PPAC. The algorithm is also validated for a LES/PDF simulation of Sandia Flame D using GRI 2.11 mechanism. For this case, PPAC-Additive exhibited 5% reduction in CPU cost compared to PPAC. The idea of this thesis is to demonstrate a method to decouple the primary and secondary chemistry, and to efficiently implement it in an adaptive chemistry framework. The LES/PDF simulation performed here is only a test case to showcase PPAC additive's different features and accuracy. Consequently, a 5% reduction in CPU time may not be significant for this particular example, however, a more practical turbulent combustion simulation (than Sandia Flame D) with more realistic fuel (than methane) and larger detailed mechanism (than only 48 species) will exhibit a significantly higher speed up in terms of CPU cost.

A novel adaptive chemistry approach, referred as dynamically partitioned adaptive chemistry (DPAC) is developed in this study. The algorithm starts with a pre-processing stage, where an initial partition is developed with an initial database, which does not have to be representative of the compositions encountered at runtime. Typically the database can be generated using any less expen-

sive low dimensional simulation, such as 0D auto-ignition. A reduced model with only a few species and no reactions is initialized for each region of the initial partition. During the adaptive simulation, the models are built from scratch using the building algorithm. Based on the current compositions observed at runtime, the accessed region of the composition space is partitioned during the simulation for a more optimal set of reduced models. Every re-partition is followed by re-initialization of reduced models and rebuilding. Proper criteria to decide if a rebuilding or a re-partition is required or not are also developed. The DPAC algorithm is validated in a propane/air non-premixed PaSR simulation first, followed by a 2D LES/PDF simulation of lifted dodecane flame. The results show that DPAC is independent of the initial database. For the LES/PDF run, we observe a 23% speed-up in CPU cost compared to the same simulation with detailed mechanism. A better performance is expected for a more practical turbulent combustion configuration with a larger detailed mechanism, as mentioned in case of PPAC Additive. The efficiency is comparable to the PPAC run with proper initial database, however, DPAC performs significantly better than PPAC if the initial database is not representative of the actual compositions at runtime. As DPAC is not dependent on the initial database, and works adaptively based on the compositions observed during the simulation, it has the potential to accurately simulate turbulent reacting flows where it is not possible to compare the accuracy with a corresponding detailed simulation. Similar to a typical grid convergence study, it will require multiple DPAC runs with progressively finer DRGEP thresholds to check if the results converge to one solution, which ensures the accuracy of DPAC

without running the simulation with detailed model. Additionally, an adaptive method such as DPAC is applicable not only in the field of combustion, but also in any dynamical system with a varying spectrum of time scales. For instance, in atmospheric chemistry such as photochemical reactions and ionizations, mechanisms of ozone destruction or halogen release at the marine boundary layer, in biological systems such as blood coagulation or any enzyme kinetics, the idea of DPAC has the potential to significantly impact the simulation efficiency.

6.2 Future work

A few followup works can significantly improve the efficiency of the of the DPAC algorithm in the context of LES/PDF computations.

- One advantage of PPAC is that the partition and models are fixed throughout the adaptive simulation. There are dedicated source term evaluation subroutines available for each model, which makes it very efficient. In case of DPAC, the model sizes change continuously. As a result, the source term evaluation always involves the use of detailed representation of the mass fraction vector. Although the reaction integration is performed in its corresponding reduced representation, the source term evaluation is very crude (however, accurate) at this stage. A more efficient source term evaluation can reduce the cost of the DPAC simulation significantly. The same problem is applicable for the analytical Jacobian evaluation.

- Previously, we have used ISAT with PPAC, which significantly improves the CPU cost. However, in DPAC, as the partition and the models are changing during the adaptive simulation, the ISAT implementation is tricky. Every time the partition changes, the ISAT table needs to be built from scratch. The good part about DPAC is that it re-partitions the accessed region only a few times. Maybe it is possible to start using ISAT after the algorithm settles on a partition. In any case, an efficient implementation of ISAT can reduce the CPU cost significantly.
- The building and partitioning at runtime in DPAC is dependent on the set of compositions, being used for them. The compositions need to be downsampled from all the compositions in the domain for fast rebuilding and re-partitioning. A random downsample does not work. In this study, we downsampled the compositions based on mixture fraction. In our experience, this is not only an extremely crude way to choose compositions, but also imposes a restriction to only non-premixed flames. A more accurate, but fast, implementation of downsampling compositions is required for better partitioning and building of models. A potential remedy is to estimate a probability map of the original dataset and then use it to uniformly sample phase-space [166]. This approach predicts the probability map without accessing the full dataset, which makes it very efficient.
- In PPAC algorithm, we observe that DRGEP based reduction provides larger set of reduced models if both primary and secondary targets are kept during the reduction process. PPAC-Additive resolves that issue with the use of build-

ing algorithm as an additional step to decouple the oxidation and secondary pathways. However, in DPAC, where we use only building algorithm with user-specified targets, it should be tested in future how it behaves when secondary species are included in the target list along with the oxidation species.

- Apart from large computational cost, the memory requirement for PDF type approaches is also very high. Before the reaction step, the reduced representation helps to reduce the cost of chemistry integration, but as the model allocation of a particle changes continuously in an adaptive simulation, each particle needs to carry the information about all species. In PPAC, as the models are fixed, and their corresponding species are known *a-priori*, each particle needs to carry the information about only the set of species consisting of the union of all models. However in DPAC, as the model size is changing, it is not easy to predict how many species information is good enough to carry. In this implementation, we carry the mass fractions of all species. This should be modified in order to adjust the particle memory allocation at runtime. As the models are not typically updated a lot once it reaches a statistically stationary state, it is expected that the cost of multiple allocation and deallocation of particle memory should be insignificant compared to the gain from carrying a significantly reduced mass fraction vector throughout the simulation.
- PDF methods provide a single framework which is applicable to all combustion modes: premixed, non-premixed and partially-premixed. The only caveat is its high computational cost. Several methods, such as model reduction, tabulation, dimension reduction, have been developed over the last decade to make

PDF approaches more tractable, some of which are developed in this study. It is probably time to go beyond simulating a Sandia Flame D or a lifted dodecane flame with ~50 species. An attempt to simulate a real combustion device with a more complex fuel in a PDF simulation with all additional components such as analytical Jacobian, QSS species (or maybe even RCCE [120]), ISAT and adaptive chemistry is long due. As a step forward in that direction, the same lifted dodecane flame will be simulated using a detailed mechanism containing ~100 species with DPAC.

BIBLIOGRAPHY

- [1] U. S. energy information administration, annual energy outlook 2022, tech. rep., 2022.
- [2] The national aeronautics and space administration, climate.nasa.gov. (Accessed: 06-18-2022).
- [3] S. C. Anenberg, J. Miller, R. Minjares, L. Du, D. K. Henze, F. Lacey, C. S. Malley, L. Emberson, V. Franco, Z. Klimont, *et al.* Impacts and mitigation of excess diesel-related nox emissions in 11 major vehicle markets. *Nature*, 545 (7655):467–471, 2017.
- [4] A. F. Ghoniem. Needs, resources and climate change: Clean and efficient conversion technologies. *Progress in energy and combustion science*, 37(1):15–51, 2011.
- [5] S. J. Jones. If electric cars are the answer, what was the question? *British medical bulletin*, 129(1):13–23, 2019.
- [6] F. N. Egolfopoulos, N. Hansen, Y. Ju, K. Kohse-Höinghaus, C. K. Law, and F. Qi. Advances and challenges in laminar flame experiments and implications for combustion chemistry. *Progress in Energy and Combustion Science*, 43:36–67, 2014.
- [7] S. B. Pope. Pdf methods for turbulent reactive flows. *Progress in energy and combustion science*, 11(2):119–192, 1985.

- [8] S. B. Pope. Small scales, many species and the manifold challenges of turbulent combustion. *Proc Comb Inst*, 34(1):1–31, 2013.
- [9] H. Pitsch. A consistent level set formulation for large-eddy simulation of premixed turbulent combustion. *Combustion and Flame*, 143(4):587–598, 2005.
- [10] V. Zimont, W. Polifke, M. Bettelini, and W. Weisenstein. An Efficient Computational Model for Premixed Turbulent Combustion at High Reynolds Numbers Based on a Turbulent Flame Speed Closure. *Journal of Engineering for Gas Turbines and Power*, 120(3):526–532, 1998.
- [11] O. Colin, A. Benkenida, and C. Angelberger. 3d modeling of mixing, ignition and combustion phenomena in highly stratified gasoline engines. *Oil & gas science and technology*, 58(1):47–62, 2003.
- [12] J. Duclos, D. Veynante, and T. Poinso. A comparison of flamelet models for premixed turbulent combustion. *Combustion and flame*, 95(1-2):101–117, 1993.
- [13] J. Van Oijen, A. Donini, R. Bastiaans, J. ten Thijsse Boonkcamp, and L. De Goey. State-of-the-art in premixed combustion modeling using flamelet generated manifolds. *Progress in Energy and Combustion Science*, 57: 30–74, 2016.
- [14] O. Gicquel, N. Darabiha, and D. Thévenin. Liminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ildm with dif-

- ferential diffusion. *Proceedings of the Combustion Institute*, 28(2):1901–1908, 2000.
- [15] N. Peters. Laminar diffusion flamelet models in non-premixed turbulent combustion. *Progress in energy and combustion science*, 10(3):319–339, 1984.
- [16] 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.
- [17] 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):055110, 2008.
- [18] V. Bykov and U. Maas. The extension of the ildm concept to reaction-diffusion manifolds. *Combustion Theory and Modelling*, 11(6):839–862, 2007.
- [19] V. Bykov and U. Maas. Problem adapted reduced models based on reaction-diffusion manifolds (redims). *Proceedings of the Combustion Institute*, 32(1): 561–568, 2009.
- [20] D. Haworth, M. Drake, S. Pope, and R. Blint. The importance of time-dependent flame structures in stretched laminar flamelet models for turbulent jet diffusion flames. In *Symposium (International) on Combustion*, volume 22, pages 589–597. Elsevier, 1989.
- [21] H. Pitsch, M. Chen, and N. Peters. Unsteady flamelet modeling of turbulent

- hydrogen-air diffusion flames. In *Symposium (international) on combustion*, volume 27, pages 1057–1064. Elsevier, 1998.
- [22] A. Y. Klimenko and S. Pope. The modeling of turbulent reactive flows based on multiple mapping conditioning. *Physics of Fluids*, 15(7):1907–1925, 2003.
- [23] M. Ihme and Y. C. See. Prediction of autoignition in a lifted methane/air flame using an unsteady flamelet/progress variable model. *Combustion and Flame*, 157(10):1850–1862, 2010.
- [24] P. Kundu, T. Echehki, Y. Pei, and S. Som. An equivalent dissipation rate model for capturing history effects in non-premixed flames. *Combustion and Flame*, 176:202–212, 2017.
- [25] P. Kundu, M. M. Ameen, C. Xu, U. Unnikrishnan, T. Lu, and S. Som. Implementation of detailed chemistry mechanisms in engine simulations. *Journal of Engineering for Gas Turbines and Power*, 141(1):011026, 2019.
- [26] D. C. Haworth. Progress in probability density function methods for turbulent reacting flows. *Progress in Energy and combustion Science*, 36(2):168–259, 2010.
- [27] J. Villermaux and J. Devillon. Représentation de la coalescence et de la redispersion des domaines de ségrégation dans un fluide par un modèle d’interaction phénoménologique. *Proc. Int. Symp Chem React. Eng.*, 26:1–13, 1972.

- [28] C. Dopazo. Probability density function approach for a turbulent axisymmetric heated jet. centerline evolution. *The Physics of Fluids*, 18(4):397–404, 1975.
- [29] R. L. Curl. Dispersed phase mixing: I. theory and effects in simple reactors. *AIChE journal*, 9(2):175–181, 1963.
- [30] A. Sandu, J. Verwer, J. Blom, E. Spee, G. Carmichael, and F. Potra. Benchmarking stiff ode solvers for atmospheric chemistry problems ii: Rosenbrock solvers. *Atmospheric environment*, 31(20):3459–3472, 1997.
- [31] M. Kuron, E. R. Hawkes, Z. Ren, J. C. Tang, H. Zhou, J. H. Chen, and T. Lu. Performance of transported pdf mixing models in a turbulent premixed flame. *Proceedings of the Combustion Institute*, 36(2):1987–1995, 2017.
- [32] M. Muradoglu, P. Jenny, S. B. Pope, and D. A. Caughey. A consistent hybrid finite-volume/particle method for the pdf equations of turbulent reactive flows. *Journal of Computational Physics*, 154(2):342–371, 1999.
- [33] L. Valino. A 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.
- [34] P. J. Colucci, F. A. Jaber, 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.

- [35] V. Raman and H. Pitsch. A consistent les/filtered-density function formulation for the simulation of turbulent flames with detailed chemistry. *Proceedings of the Combustion Institute*, 31(2):1711–1719, 2007.
- [36] S. Viswanathan, H. Wang, and S. B. Pope. Numerical implementation of mixing and molecular transport in les/pdf studies of turbulent reacting flows. *Journal of Computational Physics*, 230(17):6916–6957, 2011.
- [37] H. Wang and S. B. Pope. Large eddy simulation/probability density function modeling of a turbulent CH₄/H₂/N₂ jet flame. *Proc. Combust. Inst.*, 33(1):1319–1330, 2011.
- [38] V. Hiremath, S. R. Lantz, H. Wang, and S. B. Pope. Computationally-efficient and scalable parallel implementation of chemistry in simulations of turbulent combustion. *Combustion and flame*, 159(10):3096–3109, 2012.
- [39] V. Hiremath, S. R. Lantz, H. Wang, and S. B. Pope. Large-scale parallel simulations of turbulent combustion using combined dimension reduction and tabulation of chemistry. *Proc. Combust. Inst.*, 34(1):205–215, 2013.
- [40] W. Han, V. Raman, and Z. Chen. Les/pdf modeling of autoignition in a lifted turbulent flame: Analysis of flame sensitivity to differential diffusion and scalar mixing time-scale. *Combustion and Flame*, 171:69–86, 2016.
- [41] M. Rieth, J.-Y. Chen, S. Menon, and A. M. Kempf. A hybrid flamelet finite-rate chemistry approach for efficient les with a transported fdf. *Combustion and Flame*, 199:183–193, 2019.

- [42] A. S. Newale, S. B. Pope, and P. Pepiot. Computationally-efficient and accurate particle pdf simulations of turbulent combustion using coupled pre-partitioned adaptive chemistry and tabulation. *Proc. Combust. Inst.*, 2020.
- [43] T. Turanyi. Reduction of large reaction mechanisms. *New journal of chemistry* (1987), 14(11):795–803, 1990.
- [44] A. Coussement, B. J. Isaac, O. Gicquel, and A. Parente. Assessment of different chemistry reduction methods based on principal component analysis: Comparison of the mg-pca and score-pca approaches. *Combustion and flame*, 168:83–97, 2016.
- [45] N. J. Brown, G. Li, and M. L. Koszykowski. Mechanism reduction via principal component analysis. *International journal of chemical kinetics*, 29(6):393–414, 1997.
- [46] A. S. Tomlin, M. J. Pilling, T. Turányi, J. H. Merkin, and J. Brindley. Mechanism reduction for the oscillatory oxidation of hydrogen: sensitivity and quasi-steady-state analyses. *Combustion and flame*, 91(2):107–130, 1992.
- [47] L. Whitehouse, A. Tomlin, and M. Pilling. Systematic reduction of complex tropospheric chemical mechanisms, part i: sensitivity and time-scale analyses. *Atmospheric Chemistry and Physics*, 4(7):2025–2056, 2004.
- [48] T. Lu and C. K. Law. A directed relation graph method for mechanism reduction. *Proceedings of the Combustion Institute*, 30(1):1333–1341, 2005.

- [49] T. Lu and C. K. Law. Linear time reduction of large kinetic mechanisms with directed relation graph: n-heptane and iso-octane. *Combustion and Flame*, 144(1):24–36, 2006.
- [50] T. Lu and C. K. Law. On the applicability of directed relation graphs to the reduction of reaction mechanisms. *Combustion and Flame*, 146(3):472–483, 2006.
- [51] P. Pepiot-Desjardins and H. Pitsch. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms. *Combustion and Flame*, 154(1):67–81, 2008.
- [52] W. Sun, Z. Chen, X. Gou, and Y. Ju. A path flux analysis method for the reduction of detailed chemical kinetic mechanisms. *Combustion and Flame*, 157(7):1298–1307, 2010.
- [53] X. Gao, S. Yang, and W. Sun. A global pathway selection algorithm for the reduction of detailed chemical kinetic mechanisms. *Combustion and Flame*, 167:238–247, 2016.
- [54] P. Pepiot-Desjardins and H. Pitsch. An automatic chemical lumping method for the reduction of large chemical kinetic mechanisms. *Combustion Theory and Modelling*, 12(6):1089–1108, 2008.
- [55] L. Heberle and P. Pepiot. Automatic identification and lumping of high-temperature fuel decomposition pathways for chemical kinetics mechanism reduction. *Proceedings of the Combustion Institute*, 38(1):1053–1061, 2021.

- [56] R. Sankaran, E. R. Hawkes, J. H. Chen, T. Lu, and C. K. Law. Structure of a spatially developing turbulent lean methane–air bunsen flame. *Proceedings of the combustion institute*, 31(1):1291–1298, 2007.
- [57] X. Zheng, T. Lu, and C. Law. Experimental counterflow ignition temperatures and reaction mechanisms of 1, 3-butadiene. *Proceedings of the Combustion Institute*, 31(1):367–375, 2007.
- [58] K. E. Niemeyer, C.-J. Sung, and M. P. Raju. Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis. *Combustion and flame*, 157(9):1760–1770, 2010.
- [59] L. Heberle, P. Sharma, and P. Pepiot. Automated construction of reduced mechanisms and additive reaction modules. *Combustion and Flame*, 234:111682, 2021.
- [60] N. Peters. Numerical and asymptotic analysis of systematically reduced reaction schemes for hydrocarbon flames. In *Numerical simulation of combustion phenomena*, pages 90–109. Springer, 1985.
- [61] T. Lu and C. K. Law. Systematic approach to obtain analytic solutions of quasi steady state species in reduced mechanisms. *The Journal of Physical Chemistry A*, 110(49):13202–13208, 2006.
- [62] S. Lam and D. Goussis. The csp method for simplifying kinetics. *International Journal of Chemical Kinetics*, 26(4):461–486, 1994.

- [63] S. H. Lam. Model reductions with special csp data. *Combustion and flame*, 160(12):2707–2711, 2013.
- [64] Z. Ren and S. B. Pope. Species reconstruction using pre-image curves. *Proceedings of the Combustion Institute*, 30(1):1293–1300, 2005.
- [65] J. C. Keck. Rate-controlled constrained-equilibrium theory of chemical reactions in complex systems. *Progress in Energy and Combustion Science*, 16(2): 125–154, 1990.
- [66] 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.
- [67] A. N. Al-Khateeb, J. M. Powers, S. Paolucci, A. J. Sommes, J. A. Diller, J. D. Hauenstein, and J. D. Mengers. One-dimensional slow invariant manifolds for spatially homogenous reactive systems. *The Journal of chemical physics*, 131(2):024118, 2009.
- [68] S. B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combustion Theory and Modelling*, 1(1):41–63, 1997.
- [69] L. Lu and S. B. Pope. An improved algorithm for in situ adaptive tabulation. *Journal of Computational Physics*, 228(2):361–386, 2009.
- [70] F. Christo, A. Masri, E. Nebot, and S. Pope. An integrated pdf/neural net-

- work approach for simulating turbulent reacting systems. In *Symposium (International) on Combustion*, volume 26, pages 43–48. Elsevier, 1996.
- [71] S. R. Tonse, N. W. Moriarty, N. J. Brown, and M. Frenklach. Prism: Piecewise reusable implementation of solution mapping. an economical strategy for chemical kinetics. *Israel Journal of Chemistry*, 39(1):97–106, 1999.
- [72] Y. Shi, W. H. Green, H.-W. Wong, and O. O. Oluwole. Accelerating multi-dimensional combustion simulations using gpu and hybrid explicit/implicit ode integration. *Combustion and Flame*, 159(7):2388–2397, 2012.
- [73] K. E. Niemeyer and C.-J. Sung. Recent progress and challenges in exploiting graphics processors in computational fluid dynamics. *The Journal of Supercomputing*, 67(2):528–564, 2014.
- [74] C. Safta, H. N. Najm, and O. Knio. Tchem-a software toolkit for the analysis of complex kinetic models. Technical report, Tech. Report, SAND2011-3282, Sandia National Laboratories, 2011.
- [75] F. Bisetti. Integration of large chemical kinetic mechanisms via exponential methods with krylov approximations to jacobian matrix functions. *Combustion Theory and Modelling*, 16(3):387–418, 2012.
- [76] T. Dijkmans, C. M. Schietekat, K. M. Van Geem, and G. B. Marin. Gpu based simulation of reactive mixtures with detailed chemistry in combination with

- tabulation and an analytical jacobian. *Computers & Chemical Engineering*, 71: 521–531, 2014.
- [77] K. E. Niemeyer, N. J. Curtis, and C.-J. Sung. pyjac: Analytical jacobian generator for chemical kinetics. *Computer Physics Communications*, 215:188–203, 2017.
- [78] C. Curtiss and J. O. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Sciences*, 38(3):235–243, 1952.
- [79] G. D. Byrne and A. C. Hindmarsh. Stiff ode solvers: A review of current and coming attractions. *Journal of Computational physics*, 70(1):1–62, 1987.
- [80] P. N. Brown, G. D. Byrne, and A. C. Hindmarsh. Vode: A variable-coefficient ode solver. *SIAM journal on scientific and statistical computing*, 10(5):1038–1051, 1989.
- [81] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.
- [82] T. Lu and C. K. Law. Toward accommodating realistic fuel chemistry in large-scale computations. *Progress in Energy and Combustion Science*, 35(2): 192–215, 2009.
- [83] M. R. Youssefi. Development of analytic tools for computational flame diagnostics. Master’s thesis, University of Connecticut, 2011.

- [84] J. Tolsma and P. I. Barton. Daepack: An open modeling environment for legacy models. *Industrial & Engineering Chemistry Research*, 39(6):1826–1839, 2000.
- [85] J. E. Tolsma and P. I. Barton. On computational differentiation. *Computers & chemical engineering*, 22(4-5):475–490, 1998.
- [86] D. A. Schwer, J. E. Tolsma, W. H. Green Jr, and P. I. Barton. On upgrading the numerics in combustion chemistry codes. *Combustion and Flame*, 128(3):270–291, 2002.
- [87] F. Perini, E. Galligani, and R. D. Reitz. An analytical jacobian approach to sparse reaction kinetics for computationally efficient combustion modeling with large reaction mechanisms. *Energy & Fuels*, 26(8):4804–4822, 2012.
- [88] F. Perini, A. Krishnasamy, Y. Ra, and R. D. Reitz. Computationally efficient simulation of multicomponent fuel combustion using a sparse analytical jacobian chemistry solver and high-dimensional clustering. *Journal of Engineering for Gas Turbines and Power*, 136(9), 2014.
- [89] Y. Shi, W. H. Green, H.-W. Wong, and O. O. Oluwole. Redesigning combustion modeling algorithms for the graphics processing unit (gpu): Chemical kinetic rate evaluation and ordinary differential equation integration. *Combustion and Flame*, 158(5):836–847, 2011.
- [90] M. A. Hansen and J. C. Sutherland. On the consistency of state vectors and jacobian matrices. *Combustion and Flame*, 193:257–271, 2018.

- [91] M. J. McNenly, R. A. Whitesides, and D. L. Flowers. Faster solvers for large kinetic mechanisms using adaptive preconditioners. *Proceedings of the Combustion Institute*, 35(1):581–587, 2015.
- [92] Y. Gao, Y. Liu, Z. Ren, and T. Lu. A dynamic adaptive method for hybrid integration of stiff chemistry. *Combustion and Flame*, 162(2):287–295, 2015.
- [93] C. Xu, Y. Gao, Z. Ren, and T. Lu. A sparse stiff chemistry solver based on dynamic adaptive integration for efficient combustion simulations. *Combustion and Flame*, 172:183–193, 2016.
- [94] F. Sewerin and S. Rigopoulos. A methodology for the integration of stiff chemical kinetics on gpus. *Combustion and Flame*, 162(4):1375–1394, 2015.
- [95] T. Lu, C. S. Yoo, J. Chen, and C. K. Law. Three-dimensional direct numerical simulation of a turbulent lifted hydrogen jet flame in heated coflow: a chemical explosive mode analysis. *Journal of Fluid Mechanics*, 652:45–64, 2010.
- [96] Z. Luo, C. S. Yoo, E. S. Richardson, J. H. Chen, C. K. Law, and T. Lu. Chemical explosive mode analysis for a turbulent lifted ethylene jet flame in highly-heated coflow. *Combustion and Flame*, 159(1):265–274, 2012.
- [97] R. Shan, C. S. Yoo, J. H. Chen, and T. Lu. Computational diagnostics for n-heptane flames with chemical explosive mode analysis. *Combustion and Flame*, 159(10):3119–3127, 2012.

- [98] M. Valorani, F. Creta, D. A. Goussis, J. C. Lee, and H. N. Najm. An automatic procedure for the simplification of chemical kinetic mechanisms based on csp. *Combustion and Flame*, 146(1-2):29–51, 2006.
- [99] T. Nagy and T. Turányi. Reduction of very large reaction mechanisms using methods based on simulation error minimization. *Combustion and Flame*, 156(2):417–428, 2009.
- [100] L. Liang, J. G. Stevens, and J. T. Farrell. A dynamic adaptive chemistry scheme for reactive flow computations. *Proc. Combust. Inst.*, 32(1):527–534, 2009.
- [101] H. Yang, Z. Ren, T. Lu, and G. M. Goldin. Dynamic adaptive chemistry for turbulent flame simulations. *Combustion Theory and Modelling*, 17(1):167–183, 2013.
- [102] Z. Ren, C. Xu, T. Lu, and M. A. Singer. Dynamic adaptive chemistry with operator splitting schemes for reactive flow simulations. *Journal of Computational Physics*, 263:19–36, 2014.
- [103] X. Gou, Z. Chen, W. Sun, and Y. Ju. A dynamic adaptive chemistry scheme with error control for combustion modeling with a large detailed mechanism. *Combustion and flame*, 160(2):225–231, 2013.
- [104] T. Løvs, S. Navarro-Martinez, and S. Rigopoulos. On adaptively reduced chemistry in large eddy simulations. *Proceedings of the Combustion Institute*, 33(1):1339–1346, 2011.

- [105] W. Sun, X. Gou, H. A. El-Asrag, Z. Chen, and Y. Ju. Multi-timescale and correlated dynamic adaptive chemistry modeling of ignition and flame propagation using a real jet fuel surrogate model. *Combustion and Flame*, 162(4):1530–1539, 2015.
- [106] S. Yang, R. Ranjan, V. Yang, S. Menon, and W. Sun. Parallel on-the-fly adaptive kinetics in direct numerical simulation of turbulent premixed flame. *Proc. Combust. Inst.*, 36(2):2025–2032, 2017.
- [107] D. Zhou, K. L. Tay, H. Li, and W. Yang. Computational acceleration of multi-dimensional reactive flow modelling using diesel/biodiesel/jet-fuel surrogate mechanisms via a clustered dynamic adaptive chemistry method. *Combustion and Flame*, 196:197–209, 2018.
- [108] Y. Shi, L. Liang, H.-W. Ge, and R. D. Reitz. Acceleration of the chemistry solver for modeling di engine combustion using dynamic adaptive chemistry (dac) schemes. *Combustion Theory and Modelling*, 14(1):69–89, 2010.
- [109] D. A. Schwer, P. Lu, and W. H. Green Jr. An adaptive chemistry approach to modeling complex kinetics in reacting flows. *Combust. Flame*, 133(4):451–465, 2003.
- [110] M. A. Singer and W. H. Green. Using adaptive proper orthogonal decomposition to solve the reaction–diffusion equation. *Applied Numerical Mathematics*, 59(2):272–279, 2009.
- [111] C. Xu, M. M. Ameen, S. Som, J. H. Chen, Z. Ren, and T. Lu. Dynamic

- adaptive combustion modeling of spray flames based on chemical explosive mode analysis. *Combustion and Flame*, 195:30–39, 2018.
- [112] O. O. Oluwole, B. Bhattacharjee, J. E. Tolsma, P. I. Barton, and W. H. Green. Rigorous valid ranges for optimally reduced kinetic models. *Combustion and Flame*, 146(1-2):348–365, 2006.
- [113] I. Banerjee and M. G. Ierapetritou. An adaptive reduction scheme to model reactive flow. *Combust. Flame*, 144(3):619–633, 2006.
- [114] G. D’Alessio, A. Parente, A. Stagni, and A. Cuoci. Adaptive chemistry via pre-partitioning of composition space and mechanism reduction. *Combust. Flame*, 211:68–82, 2020.
- [115] A. Parente, J. C. Sutherland, L. Tognotti, and P. J. Smith. Identification of low-dimensional manifolds in turbulent flames. *Proceedings of the Combustion Institute*, 32(1):1579–1586, 2009.
- [116] G. D’Alessio, A. Cuoci, G. Aversano, M. Bracconi, A. Stagni, and A. Parente. Impact of the partitioning method on multidimensional adaptive-chemistry simulations. *Energies*, 13(10):2567, 2020.
- [117] G. D’Alessio, A. Cuoci, and A. Parente. Feature extraction and artificial neural networks for the on-the-fly classification of high-dimensional thermochemical spaces in adaptive-chemistry simulations. *Data-Centric Engineering*, 2, 2021.

- [118] L. Tosatto, B. Bennett, and M. Smooke. A transport-flux-based directed relation graph method for the spatially inhomogeneous instantaneous reduction of chemical kinetic mechanisms. *Combustion and Flame*, 158(5):820–835, 2011.
- [119] Y. Liang, S. B. Pope, and P. Pepiot. A pre-partitioned adaptive chemistry methodology for the efficient implementation of combustion chemistry in particle pdf methods. *Combustion and Flame*, 162(9):3236–3253, 2015.
- [120] A. S. Newale, Y. Liang, S. B. Pope, and P. Pepiot. A combined ppac-rcce-isat methodology for efficient implementation of combustion chemistry. *Combustion Theory and Modelling*, pages 1–33, 2019.
- [121] Q. Cazères, P. Pepiot, E. Riber, and B. Cuenot. A fully automatic procedure for the analytical reduction of chemical kinetics mechanisms for computational fluid dynamics applications. *Fuel*, 303:121247, 2021.
- [122] S. Pope and U. Maas. Simplifying chemical kinetics: Trajectory-generated low-dimensional manifolds. *Mechanical and Aerospace Engineering Report: FDA*, 11, 1993.
- [123] T. Lu and C. K. Law. Diffusion coefficient reduction through species bundling. *Combustion and flame*, 148(3):117–126, 2007.
- [124] S. Lam and D. Goussis. Understanding complex chemical kinetics with computational singular perturbation. In *Symposium (International) on Combustion*, volume 22, pages 931–941. Elsevier, 1989.

- [125] C. S. Yoo, T. Lu, J. H. Chen, and C. K. Law. Direct numerical simulations of ignition of a lean n-heptane/air mixture with temperature inhomogeneities at constant volume: Parametric study. *Combustion and Flame*, 158(9):1727–1741, 2011.
- [126] D. G. Goodwin, H. K. Moffat, and R. L. Speth. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. <http://www.cantera.org>, 2017. Version 2.3.0.
- [127] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).
- [128] H. Wu, P. C. Ma, and M. Ihme. Efficient time stepping for reactive turbulent simulations with stiff chemistry. In *2018 AIAA Aerospace Sciences Meeting*, page 1672, 2018.
- [129] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics. Springer-Verlag, 2010. ISBN 9783642052217.
- [130] A. Imren and D. Haworth. On the merits of extrapolation-based stiff ode solvers for combustion cfd. *Combustion and Flame*, 174:1–15, 2016.
- [131] O. Herbinet, W. J. Pitz, and C. K. Westbrook. Detailed chemical kinetic

- mechanism for the oxidation of biodiesel fuels blend surrogate. *Combustion and Flame*, 157(5):893–908, 2010.
- [132] C. K. Westbrook, C. V. Naik, O. Herbinet, W. J. Pitz, M. Mehl, S. M. Sarathy, and H. J. Curran. Detailed chemical kinetic reaction mechanisms for soy and rapeseed biodiesel fuels. *Combustion and Flame*, 158(4):742–755, 2011.
- [133] M. Mehl, W. J. Pitz, C. K. Westbrook, and H. J. Curran. Kinetic modeling of gasoline surrogate components and mixtures under engine conditions. *Proceedings of the Combustion Institute*, 33(1):193–200, 2011.
- [134] C. K. Westbrook, W. J. Pitz, P. R. Westmoreland, F. L. Dryer, M. Chaos, P. Oswald, K. Kohse-Höinghaus, T. A. Cool, J. Wang, B. Yang, N. Hansen, and T. Kasper. A detailed chemical kinetic reaction mechanism for oxidation of four small alkyl esters in laminar premixed flames. *Proceedings of the combustion institute*, 32(1):221–228, 2009.
- [135] C. K. Westbrook, W. J. Pitz, O. Herbinet, H. J. Curran, and E. J. Silke. A comprehensive detailed chemical kinetic reaction mechanism for combustion of n-alkane hydrocarbons from n-octane to n-hexadecane. *Combustion and flame*, 156(1):181–199, 2009.
- [136] S. M. Sarathy, C. K. Westbrook, M. Mehl, W. J. Pitz, C. Togbe, P. Dagaut, H. Wang, M. A. Oehlschlaeger, U. Niemann, K. Seshadri, P. S. Veloo, C. Ji, F. N. Egolfopoulos, and T. Lu. Comprehensive chemical kinetic modeling of the oxidation of 2-methylalkanes from C_7 to C_{20} . *Combustion and flame*, 158(12):2338–2357, 2011.

- [137] K. Seshadri, T. Lu, O. Herbinet, S. Humer, U. Niemann, W. J. Pitz, R. Seiser, and C. K. Law. Experimental and kinetic modeling study of extinction and ignition of methyl decanoate in laminar non-premixed flows. *Proceedings of the Combustion Institute*, 32(1):1067–1074, 2009.
- [138] C. K. Law. Combustion at a crossroads: Status and prospects. *Proceedings of the Combustion Institute*, 31(1):1–29, 2007.
- [139] V. Hiremath, Z. Ren, and S. B. Pope. A greedy algorithm for species selection in dimension reduction of combustion chemistry. *Combustion Theory and Modelling*, 14(5):619–652, 2010.
- [140] H. Goyal and P. Pepiot. A compact kinetic model for biomass pyrolysis at gasification conditions. *Energy & Fuels*, 31(11):12120–12132, 2017.
- [141] N. Peters and R. Kee. The computation of stretched laminar methane-air diffusion flames using a reduced four-step mechanism. *Combustion and Flame*, 68(1):17–29, 1987.
- [142] Y. Ju and T. Niioka. Reduced kinetic mechanism of ignition for nonpremixed hydrogen/air in a supersonic mixing layer. *Combustion and flame*, 99(2):240–246, 1994.
- [143] C. Sung, C. Law, and J.-Y. Chen. An augmented reduced mechanism for methane oxidation with comprehensive global parametric validation. *Symposium (International) on Combustion*, 27(1):295–304, 1998.

- [144] T. Løvs, D. Nilsson, and F. Mauss. Automatic reduction procedure for chemical mechanisms applied to premixed methane/air flames. *Proceedings of the Combustion Institute*, 28(2):1809–1815, 2000.
- [145] T. Lu and C. K. Law. A criterion based on computational singular perturbation for the identification of quasi steady state species: A reduced mechanism for methane oxidation with no chemistry. *Combustion and Flame*, 154(4):761–774, 2008.
- [146] A. Felden, L. Esclapez, E. Riber, B. Cuenot, and H. Wang. Including real fuel chemistry in les of turbulent spray combustion. *Combustion and Flame*, 193:397–416, 2018.
- [147] B. J. McBride. *Coefficients for calculating thermodynamic and transport properties of individual species*, volume 4513. National Aeronautics and Space Administration, Office of Management . . . , 1993.
- [148] H. Pitsch and M. Bollig. Flamemaster, a computer code for homogeneous and one-dimensional laminar flame calculations. *Institut für Technische Mechanik, RWTH Aachen*, 1994.
- [149] F. Lindemann, S. Arrhenius, I. Langmuir, N. Dhar, J. Perrin, and W. M. Lewis. Discussion on “the radiation theory of chemical action”. *Transactions of the Faraday Society*, 17:598–606, 1922.
- [150] R. Gilbert, K. Luther, and J. Troe. Theory of thermal unimolecular reactions

- in the fall-off range. ii. weak collision rate constants. *Berichte der Bunsengesellschaft für physikalische Chemie*, 87(2):169–177, 1983.
- [151] T. Løvs, P. Amnéus, F. Mauss, and E. Mastorakos. Comparison of automatic reduction procedures for ignition chemistry. *Proceedings of the Combustion Institute*, 29(1):1387–1393, 2002.
- [152] P. Pepiot. *Automatic strategies to model transportation fuel surrogates*. PhD thesis, Stanford University, 2008.
- [153] T. Yao, Y. Pei, B.-J. Zhong, S. Som, T. Lu, and K. H. Luo. A compact skeletal mechanism for n-dodecane with optimized semi-global low-temperature chemistry for diesel engine simulations. *Fuel*, 191:339–349, 2017.
- [154] F. Tagliante, T. Poinso, L. M. Pickett, P. Pepiot, L.-M. Malbec, G. Bruneaux, and C. Angelberger. A conceptual model of the flame stabilization mechanisms for a lifted diesel-type flame based on direct numerical simulation and experiments. *Combustion and Flame*, 201:65–77, 2019.
- [155] *MATLAB version 9.10.0.1613233 (R2021a)*. The Mathworks, Inc., Natick, Massachusetts, 2021.
- [156] Z. Ren, Y. Liu, T. Lu, L. Lu, O. O. Oluwole, and G. M. Goldin. The use of dynamic adaptive chemistry and tabulation in reactive flow simulations. *Combust. Flame*, 161(1):127–137, 2014.
- [157] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch. High order con-

- servative finite difference scheme for variable density low mach number turbulent flows. *Journal of Computational Physics*, 227(15):7125–7159, 2008.
- [158] C. Meneveau, T. S. Lund, and W. H. Cabot. A lagrangian dynamic subgrid-scale model of turbulence. *Journal of fluid mechanics*, 319:353–385, 1996.
- [159] C. Gruselle. *Etude du développement d'une flamme soumise à un gradient de concentration: Rôle de la stratification et des EGR*. PhD thesis, INSA de Rouen, 2014.
- [160] E. L. Petersen, D. M. Kalitan, S. Simmons, G. Bourque, H. J. Curran, and J. M. Simmie. Methane/propane oxidation at high pressures: Experimental and detailed chemical kinetic modeling. *Proc. Combust. Inst.*, 31(1):447–454, 2007.
- [161] Y. Zhang, O. Mathieu, E. L. Petersen, G. Bourque, and H. J. Curran. Assessing the predictions of a NO_x kinetic mechanism on recent hydrogen and syngas experimental data. *Combust. Flame*, 182:122–141, 2017.
- [162] R. Barlow and J. Frank. Effects of turbulence on species mass fractions in methane/air jet flames. *Symp. Int. Combust.*, 27:1087–1095, 1998.
- [163] C. Bowman, R. Hanson, D. Davidson, J. W.C. Gardiner, G. S. V. Lissianski, D. Golden, M. Frenklach, and M. Goldenberg. Gri-mech 2.11. http://www.me.berkeley.edu/gri_mech/, 1995.
- [164] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-

scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[165] F. Tagliante, L.-M. Malbec, G. Bruneaux, L. M. Pickett, and C. Angelberger. Experimental study of the stabilization mechanism of a lifted diesel-type flame using combined optical diagnostics and laser-induced plasma ignition. *Combustion and Flame*, 197:215–226, 2018.

[166] M. Hassanaly, B. A. Perry, M. E. Mueller, and S. Yellapantula. Uniform-in-phase-space data selection with iterative normalizing flows. *arXiv preprint arXiv:2112.15446*, 2021.