COMPUTATIONALLY-EFFICIENT AND SCALABLE IMPLEMENTATION OF CHEMISTRY IN LARGE-SCALE SIMULATIONS OF 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

Varun Hiremath

January 2013

COMPUTATIONALLY-EFFICIENT AND SCALABLE IMPLEMENTATION OF CHEMISTRY IN LARGE-SCALE SIMULATIONS OF TURBULENT COMBUSTION

Varun Hiremath, Ph.D.

Cornell University 2013

A major challenge in the numerical simulations of turbulent reacting flows involving large numbers of chemical species and reactions is the accurate and computationally-efficient representation of combustion chemistry. Recent advances on the experimental and theoretical fronts of the study of real fuel chemistry have led to more accurate chemical mechanisms of real fuels involving hundreds to thousands of species and thousands of reactions. However, the direct use of such detailed chemistry in large-scale calculations of turbulent reacting flows still remains computationally prohibitive. In our work, we focus on the combined Large-Eddy Simulation (LES)/Probability Density Function (PDF) computations of turbulent reacting flows, in which the thermochemical composition of the fluid is represented by a large number of particles. To reduce the cost of chemistry calculations in large-scale LES/PDF computations, we have developed a combined dimension reduction and tabulation approach in which the chemistry is represented accurately and efficiently in terms of a reduced number of "represented species". In this combined approach, the dimension reduction of combustion chemistry is performed using the Rate-Controlled Constrained-Equilibrium (RCCE) method, followed by tabulation using the *In Situ* Adaptive Tabulation (ISAT) algorithm. An automated Greedy Algorithm with Local Improvement (GALI) has been developed for selecting good represented species for use in this approach. In addition, we have developed a Partitioned Uniform Random (P-URAN) parallel strategy for the efficient parallel implementation of chemistry in large-scale LES/PDF simulations on multiple cores. This strategy has been tested by performing full-scale LES/PDF simulations of the Sandia Flame D turbulent jet flame on up to 9,216 cores and it is found to achieve good scaling. In this work (1) we describe in detail the implementation of ISAT/RCCE/GALI and the P-URAN parallel strategy; (2) we show that the combined ISAT/RCCE/GALI yields orders of magnitude speed-up with very good error control; (3) we demonstrate that our implementation of RCCE is an accurate, efficient and robust implementation; (4) we show that the P-URAN parallel strategy achieves over 85% relative weak scaling efficiency and around 60% relative strong scaling efficiency on up to 9,216 cores; (5) we show that the combined ISAT/RCCE methodology with P-URAN significantly reduces the simulation time; and (6) a combination of ISAT/RCCE and P-URAN algorithms enables us to perform accurate and computationally-efficient large-scale LES/PDF simulations with real fuel chemistry involving hundreds of chemical species.

BIOGRAPHICAL SKETCH

Varun Hiremath was born on November 15, 1985 in Gulbarga, Karnataka in India. Varun was brought up in New Delhi, India where he did his schooling. After finishing his school education in 2003, Varun joined the Indian Institute of Technology (IIT) Madras, Chennai to study Aerospace Engineering. He graduated from IIT Madras in 2008 with a B.Tech. and M.Tech. in Aerospace Engineering. He then went to the USA to pursue his graduate studies at the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY. At Cornell, he obtained an M.S. degree in 2011 and a Ph.D. in 2012.

Varun is also an open source software enthusiast, and an active contributor to many open source projects. Varun is a Debian Developer and has been contributing to the Debian GNU/Linux project, a worldwide volunteer organization dedicated to producing a high-quality, free, Linux-based operating system.

This work is dedicated to my parents, father M.G. Hiremath and mother Danamma Hiremath, and my sisters Shobha and Kaveri, for their enduring love, support and encouragement.

ACKNOWLEDGEMENTS

A Ph.D. is a long journey – one that requires a lot of inspiration, dedication, encouragement and patience to complete. With the right guidance and support, this journey can be made very enjoyable. I sure can say that I have thoroughly enjoyed working on my Ph.D., thanks to my advisor, faculty at Cornell, my friends and family.

I am very fortunate to have Professor Stephen B. Pope as my Ph.D. thesis advisor. In him, I found the perfect advisor for my Ph.D. study. I would like to thank him immensely for his constant guidance, support and encouragement through this study. His strong dedication and commitment to the field of turbulent combustion research and his outstanding contributions to this field have been very inspirational and encouraging to me and many other young researchers. I have learned through him a systematic and methodical approach to performing and presenting research work, which is going to greatly benefit me in my future work.

I am also very glad to have Professor David A. Caughey and Professor Derek H. Warner in my thesis committee. I am extremely thankful to them for their time, support and comments on this work. Outside of my thesis work, I have also had the pleasure of working with them on course projects and other assignments.

I would also like to thank our research group members - Dr. Zhuyin Ren, Dr. Haifeng Wang and Dr. Steven R. Lantz - who have contributed to this work, and I have greatly benefited from their help during this study. I also thank all the other members of our research group and my friends at Cornell for their support and appreciation.

I would like to thank the Sibley School of Mechanical and Aerospace Engineering for hosting me as a Ph.D. student and for offering an enriching graduate student program experience involving course work, research, teaching and various other activities.

Finally, I am very grateful to my parents and sisters for their love, support and encouragement through my entire life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The world's energy needs are ever increasing. According to IEA [5] and other sources, from 1990 to 2008, the world population increased by 27% and the average energy consumption per person increased by 10%, resulting in an overall 39% increase in the world energy consumption. In 2008, 81.2% of the overall energy needs was met by combustion of fossil fuels (oil 33.5%, coal 26.8%, gas 20.8%). Non-fossil fuel sources - hydro, nuclear and renewables (wind, solar, biofuels, etc.) - provided the remaining share of energy. According to the projections made by several agencies [2, 5, 4], the world's energy demand is expected to increase by another 40% by 2030 and the share of energy supply from renewables (e.g., biofuels) will continue to increase in the coming decades. However, fossil fuels will continue to remain the major source of energy, providing around 80% of the world energy needs until 2030 [2, 5, 4]. A recent study [83] states that "if the world continues to consume fossil fuels at 2006 rates, the reserves of oil, coal and gas will last a further 40, 200 and 70 years, respectively". So we can expect the fossil fuels to continue to provide a major share of world's energy for many decades to come.

The combustion of fossil fuels results in the emission of greenhouse gases and other air pollutants, which is of increasing environmental concern. The emission of greenhouse gases (primarily carbon dioxide, CO_2) results in global climate change, which in a long term is expected to have adverse effects on the environment and human society. According to IPCC [3], "the observed increase in global average temperatures since the mid-20th century is very likely ($> 90\%$) due to the observed increase in anthropogenic greenhouse gas concentrations".

The IPCC highlights the need for mitigation of greenhouse gas emissions over the coming decades through energy policies, imposition of emission standards and regulations [3]. To control the emissions, various countries have been imposing emission standards on the engine manufacturers, and these emission standards are becoming more and more stringent with time. In the near future, it will become very challenging for the engine manufacturers to meet these emission standards if advancements are not made in computational and experimental tools for engine design.

With the impetus for cleaner energy sources and design of more efficient combustion devices which meet the stricter emission standards in the future, the combustion research has a significant role to play in the coming decades. Furthermore, most practical combustion devises (for example the gas turbine engines and the reciprocating internal combustion engines) used in the transportation and industrial sectors operate in the turbulent regime. Consequently, the study of *turbulent combustion* poses an even more challenging task. Over the years, significant advances have been made in the study of turbulent flows [67] and turbulent combustion [61, 22]. Progress on all the three fronts - theoretical, experimental and computational - has lead to better understanding of the processes involved in turbulent combustion problems.

This work focuses primarily on the development of computational algorithms for the study of turbulent combustion and reacting flow problems. The numerical study of turbulent reacting flows poses three-fold challenges: (1) resolving the wide range of length and time scales present in the turbulent flow [67, 70]; (2) tracking the composition of a large number of chemical species when dealing with real fuel chemistry [100, 82, 71]; and (3) capturing accurately the turbulence-chemistry interactions [74, 6].

Simulation of turbulent non-reacting flows in itself is very challenging, and various approaches have been developed over the years for studying turbulent flows [67]. One of the earliest approaches used for simulating turbulent flows, which is still prevalent, is the Reynolds-Averaged Navier-Stokes (RANS) approach [21, 39]. This approach involves solving for the time-averaged mean flow quantities with a model for all the unclosed terms in the governing equations. The RANS approach, due to its relatively simple implementation and great computational-efficiency, has been used for studying a wide range of turbulent flow problems. However, due to its over-simplification of the physics and modeling issues, the RANS approach is not accurate enough to simulate complex flows and capture all the interesting flow characteristics.

In the last two decades, starting in the 1990s, the availability of faster computational resources and the introduction of advanced programming languages and tools for performing large-scale distributed parallel computing, has helped two alternative approaches for the simulation of turbulent flows to emerge. These are Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES). The DNS approach involves solving numerically the Navier-Stokes equations on very fine grids to resolve all the required flow and combustion length and time scales without any modeling [34, 16]. The DNS approach provides a great tool for studying low to moderate Reynolds number turbulent flows. However, it is still prohibitively expensive to use DNS for simulating high Reynolds number turbulent flows and practical combustion devices like gas turbine engines, and it is likely to remain so for several decades.

In contrast, in the LES approach, only the large scale motions are resolved on the grid, while the small scale sub-grid motions are modeled [25, 47, 45]. This approach is thus computationally much less expensive than DNS, and yet provides enough resolution to capture the important flow characteristics. Over the last decade, LES has been widely accepted as the more suitable approach for simulating high Reynolds number turbulent flows. In this study as well, we employ the LES methodology to solve for the turbulent flow field.

For simulating turbulent reacting flows, in addition to solving for the turbulent flow characteristics, we also need to solve for the chemical species composition and to handle the turbulence-chemistry interactions. Most modern day numerical solvers for turbulent reacting flows have two components: one to solve for the (non-reacting) turbulent flow, typically referred to as the flow solver; and another to solve for the combustion chemistry, typically referred to as a combustion model. These two components are coupled and work in tandem to incorporate the turbulence-chemistry interactions.

A major challenge in the simulation of reacting flow with real fuel chemistry is the presence of a large number of chemical species and reaction time scales, which makes reaction computations prohibitively expensive. A very good discussion of the various approaches developed over the years for combustion modeling is provided in [71]. As described in [71], most combustion modeling approaches are either Flamelet based or Probability Density Function (PDF) based. In the Flamelet based approaches, instead of keeping track of all the chemical species composition, the chemistry is represented in terms of two to three reduced scalars. So these approaches are based on a very strong assumption that the chemical species compositions lie around a very-low-dimensional manifold in the high-dimensional composition space and thus can be parameterized in terms of two to three reduced scalars. The Flamelet based approaches are thus very easy to implement and computationally very efficient. These have been widely used for studying turbulent combustion problems [61, 63]. However, due to their over-simplification of the turbulence chemistry interactions, these approaches have limited applicability and are in general not very accurate for more complex flows involving extinction re-ignition and strong turbulence-chemistry interactions [71]. In contrast, the PDF based methods enable the direct use of detailed chemistry (without any modeling) in reacting flow computations [65]. Over the years, the PDF methods have proven to be more accurate at representing chemistry and the turbulence-chemistry interactions in the more complex and challenging flow regimes [14, 96, 102]. Here we focus on the PDF approach to solve for the combustion chemistry.

In this study we focus on turbulent reacting flows at low Mach numbers and neglect acoustic and compressibility effects. To solve for the turbulent flow field we are using an LES solver [62, 63] obtained from the Stanford university and further developed in our group at Cornell. This LES solver has been coupled with a PDF solver, named HPDF [97], which has been entirely developed at Cornell. The HPDF solver solves for the composition PDF using Lagrangian particle based methods. In this approach, the thermochemical composition of the fluid is represented by a large number of notional particles in the computational domain [97, 29, 30].

The HPDF solver has three main components:

1. *transport*: to account for the change in position of the particles due to advection (including a random-walk model for molecular transport in cer-

tain implementations [98, 95]);

2. *mixing*: to account for the change in chemical composition of particles due to mixing (which models the molecular diffusion);
3. *reaction*: to account for the change in chemical composition of particles due to chemical reaction.

These components are implemented in fractional steps [98] and thus can be studied individually.

In coupled LES/PDF simulations of turbulent reacting flows with detailed chemistry (without any simplification), the reaction computations are extremely expensive and may consume over 99% of the overall computation time [29]. The main focus of this work is to develop algorithms to reduce the cost of reaction computations without losing much of the accuracy offered by the detailed mechanisms and PDF methods.

Over the years various approaches have been developed to reduce the cost of reaction computations, which can be broadly classified into the following three categories:

1. *mechanism reduction*: methods in this category are aimed at developing skeletal mechanisms involving fewer species by systematically removing unimportant species and reactions (within a specified error tolerance) from the detailed mechanisms [53, 60, 58]. The use of a skeletal mechanism (in place of the corresponding detailed mechanism) significantly reduces the number of species that need be tracked in the computations, thereby greatly reducing the computational cost.

2. *dimension reduction:* these approaches focus on representing the detailed chemistry in terms of a reduced number of variables [41, 44, 55, 79]. Only the reduced variables are tracked in the computations, so the computational cost is significantly reduced.

3. *storage and retrieval:* these methods are based on the idea of storing the reaction solutions in a table in an attempt to retrieve a linear approximation to the reaction solution in the subsequent time-steps using the stored solutions in the table [66, 49, 92, 93]. Since retrieving an approximate solution from the table is relatively inexpensive, these methods are found to significantly reduce the overall computational cost.

In addition, several combined methodologies [91, 89, 77] have also been developed which employ a combination of the aforementioned methods; and more recently *adaptive chemistry* methodologies [46, 54, 19] have also been developed which use the aforementioned methods on-the-fly during a simulation to reduce the computational cost.

In this work, we develop a combined dimension reduction and tabulation approach using the Rate-Controlled Constrained-Equilibrium (RCCE) [41, 40, 38] dimension reduction method and the *In Situ* Adaptive Tabulation (ISAT) [66, 49] tabulation algorithm. This combined ISAT/RCCE [32] methodology enables representing chemistry accurately and efficiently in terms of a reduced number of specified *represented species*. An automated Greedy Algorithm with Local Improvement (GALI) [31, 32] has been developed for selecting good represented species for use in this combined approach. This combined ISAT/RCCE/GALI methodology can be used either directly with a detailed mechanism or with a skeletal mechanism (for very large detailed mechanisms). As a result, we are using a combination of all the three aforementioned approaches to represent chemistry, and as is demonstrated in this work, this combined approach helps reduce the computational cost associated with reaction by orders of magnitude with very good error control.

We first implement and test this combined ISAT/RCCE/GALI [31, 32] methodology in a Partially-Stirred Reactor (PaSR), which is a computationally cheaper test case representative of real LES/PDF simulations. We then extend our LES/PDF solver with the capability of representing chemistry using this combined ISAT/RCCE methodology. In performing large-scale parallel LES/PDF simulations, significant load imbalance among the participating cores is created due to reaction since the chemical reactivity is in general not uniform over the entire computational domain [48, 29]. As a result, additional parallel strategies are required to redistribute the reaction work load in order to minimize the overall simulation time. We implemented various parallel strategies using the *x2f_mpi* Fortran library to manage the reaction load distribution. In particular a Partitioned Uniform Random (P-URAN) [29] strategy is found to perform the best among all the strategies and scales well to a large number of cores.

In this work, we describe in detail the implementation of the combined ISAT/RCCE/GALI algorithm and the P-URAN parallel strategy. We first quantify the accuracy and efficiency of the combined ISAT/RCCE methodology using the PaSR, and then demonstrate the accuracy, efficiency and scalability of the combined ISAT/RCCE and P-URAN strategy for performing large-scale LES/PDF simulations of the Sandia Flame D [9] – a methane/air turbulent piloted jet flame.

The outline of the remainder of this thesis is as follows:

- In Chapter 2, we describe the Greedy algorithm used for selecting good represented species for use in the RCCE dimension reduction method.
- In Chapter 3, we describe in detail the implementation of the combined ISAT/RCCE algorithm and an extended Greedy Algorithm with Local Improvement (GALI) for selecting good represented species.
- In Chapter 4, we explore different parallel strategies for managing reaction load distribution in large-scale LES/PDF simulations. We describe the Partitioned Uniform Random (P-URAN) strategy which is found to perform the best among all the strategies tested and scales well to large number of cores. We implement and test these strategies for performing large-scale LES/PDF simulations of Sandia Flame D on up to 9,216 cores.
- In Chapter 5, we extend our LES/PDF solver with the option of representing chemistry using our combined ISAT/RCCE methodology with parallel implementation using the P-URAN strategy. We present results for the Sandia Flame D.
- In Chapter 6, we compare the relative accuracy and efficiency of our implementation of RCCE using the Trajectory In Full Space (TIFS) approach with some of the previous implementations of RCCE. We show that our implementation of RCCE is the most accurate, efficient and robust implementation.
- Finally, in Chapter 7, we conclude by listing the major contributions of this work and future challenges.

CHAPTER 2

A GREEDY ALGORITHM FOR SPECIES SELECTION IN DIMENSION REDUCTION OF COMBUSTION CHEMISTRY [†]

2.1 Abstract

Computational calculations of combustion problems involving large numbers of species and reactions with a detailed description of the chemistry can be very expensive. Numerous dimension reduction techniques have been developed in the past to reduce the computational cost. In this paper, we consider the rate controlled constrained-equilibrium (RCCE) dimension reduction method, in which a set of constrained species is specified. For a given number of constrained species, the “optimal” set of constrained species is that which minimizes the dimension reduction error. The direct determination of the optimal set is computationally infeasible, and instead we present a greedy algorithm which aims at determining a “good” set of constrained species; that is, one leading to near-minimal dimension reduction error. The partially-stirred reactor (PaSR) involving methane premixed combustion with chemistry described by the GRI-Mech 1.2 mechanism containing 31 species is used to test the algorithm. Results on dimension reduction errors for different sets of constrained species are presented to assess the effectiveness of the greedy algorithm. It is shown that the first four constrained species selected using the proposed greedy algorithm produce lower dimension reduction error than constraints on the major species: CH_4, O_2, CO_2 and H_2O. It is also shown that the first ten constrained species selected using the proposed greedy algorithm produce a non-increasing dimen-

[†] 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.

sion reduction error with every additional constrained species; and produce the lowest dimension reduction error in many cases tested over a wide range of equivalence ratios, pressures and initial temperatures.

2.2 Introduction

Modern chemical mechanisms for real fuels typically involve hundreds of species and thousands of reactions [64, 100]. Computational calculations of reactive flows involving such fuels with detailed chemistry are prohibitive even on a distributed computing platform.

Numerous techniques have been developed in the past to reduce the computational cost of implementing combustion chemistry. These include:

1. Skeletal Mechanisms: A skeletal mechanism consists of a subset of the species and reactions from the detailed mechanism. Many methods have been developed to systematically generate skeletal mechanisms from detailed mechanisms, such as the Directed Relation Graph (DRG) [53], DRG with error propagation (DRGEP) [60] and Simulation Error Minimization Connectivity Method (SEM-CM) [58].
2. Reduced Chemical Mechanisms (based on QSSA): The quasi-steady-state approximation (QSSA) [11, 87] has been widely applied to develop reduced chemical mechanisms. The QSSA method involves the identification of QSS species in the system, whose net rate of production is assumed to be zero, thereby reducing the governing differential equation for the QSS species into an algebraic relation. These algebraic relations are used

to eliminate the QSS species from the system.

3. Dimension Reduction Methods: Another class of dimension reduction techniques is based on the observation that chemical systems involve reactions with a wide range of time scales. As a consequence, reaction trajectories are attracted to lower dimensional attracting manifolds in the composition space. Computations can be performed in a reduced space by identifying such low-dimensional manifolds, thereby reducing the overall computational cost. Methods based on this idea include the rate-controlled constrained equilibrium (RCCE) [41, 40], computational singular perturbation (CSP) [44], intrinsic low-dimensional manifolds (ILDM) [55], trajectory-generated low-dimensional manifolds (TGLDM) [72], invariant constrained equilibrium-edge pre-image curve (ICE-PIC) [79] and one-dimensional slow invariant manifold (1D SIM) [7].

4. Storage Retrieval Methods: In these approaches, combustion chemistry computations are stored in a table, and are used to build inexpensive approximate solutions at a later stage of computation to reduce the overall cost. Methods based on this idea include the structured look-up tabulation [17], repro-modelling [93], artificial neural network (ANN) [18], in *situ* adaptive tabulation (ISAT) [66, 49] and piecewise reusable implementation of solution mapping (PRISM) [92].

In recent times, combined methodologies have also been developed, wherein reduced reaction mechanisms or dimension reduction methods are used in conjunction with storage/retrieval methods, such as ISAT-QSSA [91], ISAT-RCCE [89], and recently ICE-PIC with ISAT [77].

In reactive flow calculations, the species concentration are governed by two processes: chemical reaction and transport. We consider the important class of solution methods in which splitting scheme is used, where the chemical reaction and transport processes are accounted for in two separate steps. In the computational modelling of the turbulent combustion using PDF methods [65], the fluid composition within the solution domain is represented by a large number of particles. In a full-scale PDF calculation, more than 2 million particles may be used, and the solution can advance for more than 5000 time steps, leading to approximately 10^9 particle-reaction sub-steps. If such a calculation involves 100 species with the chemistry represented by a detailed mechanism, then at each reaction sub-step, 100 coupled ODEs need to be solved to determine the species concentrations, which can be very expensive and computationally prohibitive. Instead a dimension reduction method (such as RCCE or ICE-PIC) integrated with ISAT can be used to perform the reactive flow calculations in terms of say 10 represented species; where the combined reduction-storage methodology determines and tabulates (*in situ*) the reduced space in terms of the 10 represented species based on the detailed mechanism.

In a reactive system, the reaction trajectories rapidly approach a hierarchy of attracting manifolds of decreasing dimensions and the reactive system's slow dynamics is well approximated by these low-dimensional attracting slow invariant manifolds (SIMs) in the reactive space. Numerous dimension reduction methods have been developed which exploit this property to represent the chemistry using a reduced set of variables. Here we focus on the RCCE and ICE-PIC dimension reduction methods, which have been successfully implemented and used in many reactive flow computations [80, 35, 36, 77]. The slow invariant manifold is approximated in the RCCE method by the constrained-equilibrium manifold (CEM) constructed using thermodynamic concepts, and in the ICE-PIC method by the invariant constrained-equilibrium edge (ICE) manifold which is a collection of all the reaction trajectories emanating from the CEM edge. Recent studies [7] have shown that the consideration of the topology of classical thermodynamic functions may not provide a good approximation to the 1D SIM. However, in RCCE and ICE-PIC we typically use more than one constrained species (i.e., work in higher dimensions), and the errors in the approximation generally decrease in higher dimensions. We also note that the method described by [7] has yet to be applied in many dimensions.

The RCCE dimension reduction method (originally proposed in [41]) is based on the assumption that in a reactive system, the reaction trajectories relax to the complete equilibrium with a rate determined solely by the slow reactions, while the fast reactions tend to locally equilibrate the system subject to the constraints imposed by the slow reactions. Thus, the system reaches the complete equilibrium by evolving through a sequence of constrained-equilibrium states on the CEM. As a result, only the rate equations of the slowly changing constraints need to be solved, though a different approach is used in our implementation of RCCE as described in Section 2.3.3. The constrained-equilibrium state can be determined (locally) by computing the state corresponding to the maximum entropy subject to the given set of constraints.

The very first step involved in the application of RCCE (and ICE-PIC) method is specification of the constraints for dimension reduction. The conservation of elements form the most basic time-independent constraints. Among the time-dependent constraints, a literature review [27, 35, 36] shows that most commonly used constraints in RCCE include general linear constraints on the total number of moles; moles of active valence (AV); moles of free oxygen (FO) etc.

In our implementations of RCCE and ICE-PIC, which are integrated with ISAT [89, 79, 77, 78], the concentration of a specified set of constrained species form the constraints. The specification of good constraints is crucial for the accuracy of dimension reduction, but there are no systematic methods available to select good constraints in an automated way.

Ideally one wants to find the smallest set of constrained species that yields the dimension reduction errors below a specified tolerance; or one wants to find the "optimal" set of constrained species that minimizes the dimension reduction errors for a fixed number of constrained species. This is a very hard problem, so we aim to devise an algorithm to select a "good" set of constrained species, i.e., a near-optimal set of species which produces low dimension reduction errors.

The proposed method works by considering a computationally inexpensive representative test problem (the partially-stirred reactor (PaSR)), and directly measuring the dimension reduction error. A "good" set of constrained species are selected by employing a greedy algorithm; which selectively adds "good" species to the set (initially empty) one at a time to minimize the dimension reduction error at each stage. Here we consider the application of this method for RCCE, but the methodology developed is also applicable to ICE-PIC. The SEM-CM [58] method used for developing skeletal mechanisms employs a similar idea for identifying the species to be retained in the skeletal mechanism. In the SEM-CM method, the mechanism building procedure is started from a set of specified important species, and then species are added (based on ranking) one at a time until the simulation error using the skeletal mechanism becomes smaller than a required threshold.

The outline of the remainder of the paper is as follows: In Section 2.3, we develop a mathematical representation for a gas-phase reacting system. We then describe the RCCE dimension reduction procedure and define the dimension reduction errors involved in this procedure. In Section 2.4, we propose a definition for the "optimal" set of species and later present the greedy algorithm for selecting a "good" (near-optimal) set of species. In Section 2.5 we present results obtained using the greedy algorithm for the test case of a partially-stirred reactor (PaSR) with methane premixed combustion. Finally, in Section 2.8 we draw conclusions based on the presented results.

2.3 Representation of Chemistry

We consider a reacting gas-phase mixture consisting of n_s chemical species, composed of n_e elements. The set of all species is denoted by $\mathbf{\Phi}$. The thermochemical state of the mixture (at a given position and time) is completely characterized by the pressure p, the mixture enthalpy h, and the n_s-vector $\mathbf{z}$ of specific moles of the species. To simplify the exposition, we consider an adiabatic and isobaric system with h and p taken to be given constants, and so the thermochemical state is given by $\mathbf{z}$.

It is useful to consider the species composition $\mathbf{z}$ to be an n_s-vector or a point in the n_s-dimensional *full composition space*.

2.3.1 Reaction Trajectories

Due to chemical reactions, the chemical composition $\mathbf{z}$ evolves in time according to the following set of ordinary differential equations (ODEs)

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{S}(\mathbf{z}(t)), \tag{2.1}$$

where $\mathbf{S}$ is the n_s-vector of chemical production rates determined by the chemical mechanism used to represent the chemistry.

The *reaction mapping*, $\mathbf{R}(\mathbf{z}, t)$ is defined to be the solution to Eqn.2.1 after time t starting from the initial composition $\mathbf{z}$. In this work, the *reaction mapping* is computed by numerically integrating the set of ODEs (2.1) using DDASAC [15].

2.3.2 Dimension Reduction

The dimension reduction methods that we are interested in are methods based on low-dimensional manifolds, and in particular the RCCE and ICE-PIC dimension reduction methods. In this section we briefly describe the notation used in these dimension reduction methods; detailed descriptions are provided in [79, 77].

In RCCE and ICE-PIC, the set of species $\mathbf{\Phi}$ is decomposed as $\mathbf{\Phi} = \{\mathbf{\Phi}^r, \mathbf{\Phi}^u\}$, where $\mathbf{\Phi}^r$ is the set of *represented* species with cardinality n_{rs}, and $\mathbf{\Phi}^u$ is the set of *unrepresented* species with cardinality n_{us}, where $n_{rs}+n_{us} = n_s$ and $n_{rs} < n_s - n_e$.

The *reduced representation* of the species composition is denoted by $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$, where $\mathbf{z}^r$ is n_{rs}-vector of specific moles of represented species, $\mathbf{\Phi}^r$; and

$\mathbf{z}^{u,e}$ is an n_e-vector giving the specific moles of the elements in the unrepresented species, $\mathbf{\Phi}^u$ (for atom conservation). Thus, $\mathbf{r}$ is a vector of length $n_r = n_{rs} + n_e$ in the *reduced composition space*, and the dimension of the system is reduced from n_s to $n_r < n_s$. At any time t, the reduced representation, $\mathbf{r}(t)$ is related to the full composition, $\mathbf{z}(t)$ as

$$\mathbf{r}(t) = \mathbf{B}^T \mathbf{z}(t), \tag{2.2}$$

where $\mathbf{B}$ is constant $n_s \times n_r$ matrix which can be determined for a specified set of represented species.

2.3.3 Steps Involved in Dimension Reduction

In this section we briefly describe the four main steps involved in our implementation of RCCE. Since our implementation of RCCE is integrated with ISAT, some of the steps in our implementation of RCCE differ from other works found in the literature, those steps are highlighted and justified.

1. The first important step in the application of the RCCE dimension reduction method is the *selection* of the set of represented (constrained) species, $\mathbf{\Phi}^r$. For a given set of represented species, $\mathbf{\Phi}^r$, the reduced representation is given as $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$.

   Alternatively, in many of the RCCE implementations [35, 36] general linear constraints on species are specified. In our implementation of RCCE, to simplify the user interface and specification of constraints, we use the species specific moles of the represented species as the constraints.

2. The next step is the *species reconstruction*, i.e., given a reduced representation $\mathbf{r}(0)$ at time $t = 0$, reconstruct an estimate of the full composition

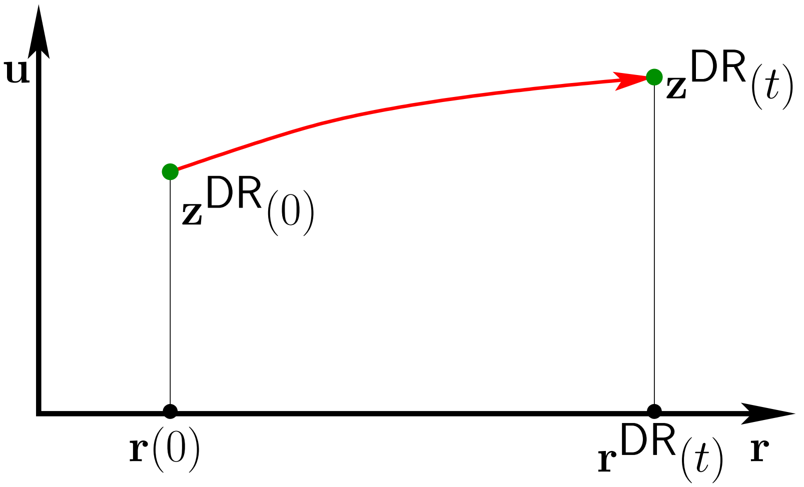

u
$\mathbf{z}^{\mathsf{DR}}(t)$
$\mathbf{z}^{\mathsf{DR}}(0)$
$\mathbf{r}(0)$
$\mathbf{r}^{\mathsf{DR}}(t)$
r

Figure 2.1: Sketch of the composition space (indicated by represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ axes) showing the four steps involved in the dimension reduction procedure using RCCE and ICE-PIC: (i) the initial reduced composition is represented by $\mathbf{r}(0)$; (ii) the reconstructed composition at $\mathbf{r}(0)$ is represented by $\mathbf{z}^{DR}(0)$; (iii) the reaction mapping starting from $\mathbf{z}^{DR}(0)$ after time t is represented by $\mathbf{z}^{DR}(t)$; (iv) the reduced composition after time t is represented by $\mathbf{r}^{DR}(t)$.

denoted by $\mathbf{z}^{DR}(0)$.

In RCCE the *species reconstruction* is performed by computing the constrained-equilibrium composition for the given constraints. In our implementation of RCCE, the constrained-equilibrium composition is computed using the CEQ [68] code, with the constraints given by the reduced representation $\mathbf{r}$. The constrained-equilibrium composition at $\mathbf{r}$ is denoted by $\mathbf{z}^{CE}(\mathbf{r})$. So the reconstructed composition in RCCE is given as

$$\mathbf{z}^{DR}(0) = \mathbf{z}^{CE}(\mathbf{r}(0)). \tag{2.3}$$

3. The next step is to obtain the *reaction mapping*. Starting from the reconstructed composition, $\mathbf{z}^{DR}(0)$, the set of ODEs (2.1) are integrated numerically in the full space using DDASAC to obtain the reaction mapping after

time t denoted by $\mathbf{z}^{DR}(t)$ as shown in Fig.2.1.

An alternative approach for RCCE method as suggested in [40] and also used in [35, 36] is to integrate the rate equations for the constraint potentials which is more economical than integrating the ODEs (2.1) directly. In our implementation of RCCE, we chose the latter approach for the ease of integrating RCCE dimension reduction method with ISAT, which is discussed in more detail in [89].

4. The final step involved in the dimension reduction method is *reduction*, i.e., from the obtained *reaction mapping* after time t, $\mathbf{z}^{DR}(t)$, compute the *reduced representation* denoted by $\mathbf{r}^{DR}(t)$ (shown in Fig.2.1), given as:

$$\mathbf{r}^{DR}(t) = \mathbf{B}^T\mathbf{z}^{DR}(t). \tag{2.4}$$

To summarize, the key steps involved in the RCCE dimension reduction method are

1. **Selection**: Identifying good constraints or the set of represented species, $\mathbf{\Phi}^r$ for dimension reduction.
2. **Species Reconstruction**: Given the constraints, $\mathbf{r}(0)$, reconstructing the full composition, $\mathbf{z}^{DR}(0)$.
3. **Reaction Mapping**: Starting from the *reconstructed composition* $\mathbf{z}^{DR}(0)$, computing the *reaction mapping* after time t in the full composition space $\mathbf{z}^{DR}(t)$.
4. **Reduction**: From the *reaction mapping* $\mathbf{z}^{DR}(t)$, obtaining the *reduced representation* $\mathbf{r}^{DR}(t)$ after time t.

The ICE-PIC dimension method also involves the same four aforementioned steps, with the only difference in the *species reconstruction* step, wherein the reconstructed composition is defined based on the invariant constrained-equilibrium (ICE) manifold [79, 77]. The remaining three steps: *selection*, *reaction mapping* and *reduction* are identical to the steps in RCCE.

Among these steps, the *selection* of the represented (constrained) species is an important step as the errors involved in the remaining three steps implicitly depend on the choice of the represented species, $\mathbf{\Phi}^r$. As mentioned in [35, 36] also, identification of appropriate constraints is essential for the accuracy of the RCCE dimension reduction method. In the following sections, we develop an automated algorithm to select a "good" set of represented species, $\mathbf{\Phi}^r$ for the accurate implementation of RCCE and ICE-PIC dimension reduction methods.

2.3.4 Partially-Stirred Reactor

In methods to develop QSSA based reduced mechanisms, it is useful to consider a range of test cases both to identify QSS species and to validate the resulting reduced mechanisms.

Here, we are interested in applying RCCE and ICE-PIC methods to LES/PDF calculations, for which the partially-stirred reactor (PaSR) is a (computationally cheaper) representative test case. We can vary pressure p, temperature T and the time step Δt to be representative of conditions in an LES/PDF calculation.

In this study, we consider the test case of a partially-stirred reactor (PaSR) involving premixed combustion of a methane/air mixture. A description of the

PaSR is given in [66]; here we list only the important parameters involved.

There are two inflowing streams: a premixed stream of stoichiometric methane/air mixture at 600 K; and a pilot stream consisting of the adiabatic equilibrium products of a stoichiometric methane/air mixture at 2375 K (corresponding to unburnt gas temperature of 600 K). The mass flow rates of these streams are in the ratio 0.95:0.05. Initially (t = 0), all particle compositions are set to be the pilot-stream composition. The pressure is atmospheric throughout.

Other important parameters involved are: number of particles, N_P = 100; residence time, τ_{res} = 10 ms; mixing time scale, τ_{mix} = 1 ms; pairing time scale, τ_{pair} = 1 ms; time step = 0.1 ms (involving 3 fractional sub-steps of mixing, reaction and mixing); and reaction time step Δt = 0.033 ms. The PaSR is run for 3,400 time steps, each involving 3 sub-steps over 100 particles, leading to more than 10^6 particle-sub-steps.

In this study, the GRI-Mech 1.2 mechanism involving 31 species is used to describe the methane combustion. The species involved are

$$\begin{aligned}\mathbf{\Phi} = \{&H_2, H, O_2, OH, H_2O, CH_3, CH_4, CO, CO_2, CH_2O, C_2H_4, O, HO_2, H_2O_2, C, CH, CH_2, CH_2(S), \\ &HCO, CH_2OH, CH_3O, CH_3OH, C_2H, C_2H_2, C_2H_3, C_2H_5, C_2H_6, HCCO, CH_2CO, HCCOH, N_2\}\end{aligned} \tag{2.5}$$

To give some idea about the state of the PaSR, scatter plots of species specific moles of CH_4 and CO (retrieved from 10 selected particles from the PaSR) versus the temperature are shown in Fig.2.2 and Fig.2.3, respectively. We see that the CH_4 concentration drops with temperature, as more and more CH_4 reacts to form products. The concentration of CO on the other hand, increases with temperature and reaches a maximum at around $2000K$.

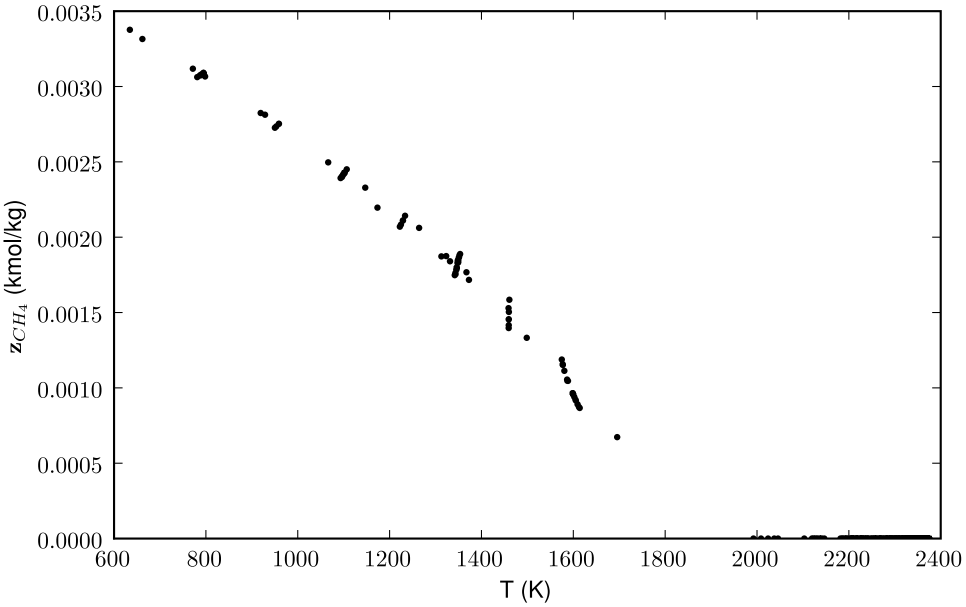


Figure 2.2: Scatter plot of the specific moles of CH_4, $\mathbf{z}_{CH_4}$ (retrieved from 10 selected particles from the PaSR) versus temperature, T, obtained from 3,400 time steps of $0.1\,ms$ each in the PaSR for a stoichiometric methane premixed combustion at atmospheric pressure and an initial temperature of $600\,K$.

In our implementation of PaSR, computations can be performed using the full set of species, $\mathbf{\Phi}$, in the full composition space (without any dimension reduction) or using a smaller set of represented species, $\mathbf{\Phi}^r$, with one of the dimension reduction methods – RCCE or ICE-PIC. For a given test case, PaSR calculations are performed with and without dimension reduction, and the compositions obtained with the two approaches are compared to estimate errors involved in dimension reduction.

In the next section we define the various errors involved in the dimension reduction steps. Subsequently, based on these definitions of error, we propose a definition for “optimal” set of represented species and present an algorithm to

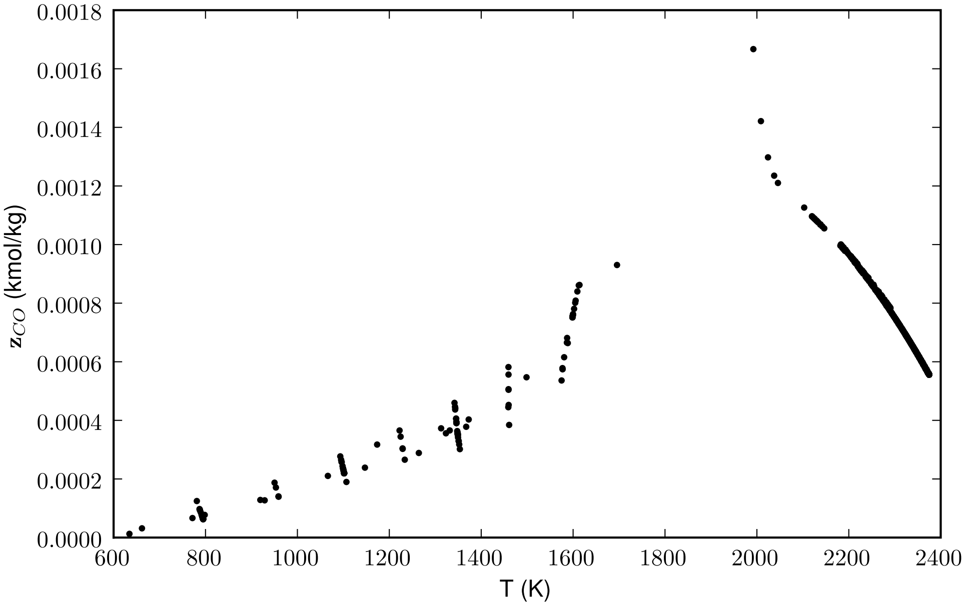


Figure 2.3: Scatter plot of the specific moles of CO, $\mathbf{z}_{CO}$ (retrieved from 10 selected particles from the PaSR) versus temperature, T, obtained from 3,400 time steps of $0.1\,ms$ each in the PaSR for a stoichiometric methane premixed combustion at atmospheric pressure and an initial temperature of $600\,K$.

select "good", near-optimal set of represented species.

2.3.5 Dimension Reduction Errors

In this section we define the various errors involved in the dimension reduction process and describe the method employed to measure these errors using the PaSR.

Given a composition, $\mathbf{z}(0)$ in the full composition space, the *reaction mapping*, $\mathbf{R}(\mathbf{z}(0), t)$ (for $t \geq 0$) is more concisely denoted by $\mathbf{z}(t)$ (see Fig.2.4).

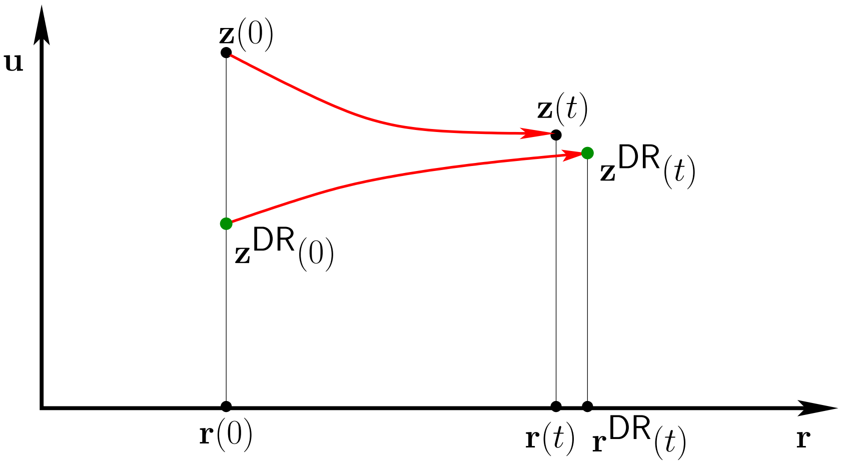


Figure 2.4: Sketch of the steps involved in the computation of dimension reduction errors. $\mathbf{z}(0)$ and its reaction mapping after time t, $z(t)$, are given compositions in the full space. For a specified set of represented species, $\mathbf{\Phi}^r$, the reduced representation at $\mathbf{z}(0)$ in denoted by $\mathbf{r}(0)$. The reconstructed composition at $\mathbf{r}(0)$ is denoted by $\mathbf{z}^{DR}(0)$. The reaction mapping from $\mathbf{z}^{DR}(0)$ after time t is denoted by $\mathbf{z}^{DR}(t)$, and the reduced composition at $\mathbf{z}^{DR}(t)$ is denoted by $\mathbf{r}^{DR}(t)$.

For a given set of *represented* species, $\mathbf{\Phi}^r$, the *reduced* representation of the full composition, $\mathbf{z}(0)$ is denoted by $\mathbf{r}(0)$ and is obtained by performing the *reduction* using (2.2) as

$$\mathbf{r}(0) = \mathbf{B}^T\mathbf{z}(0). \tag{2.6}$$

At $\mathbf{r}(0)$, the reconstructed composition using a dimension reduction method is denoted by $\mathbf{z}^{DR}(0)$. Starting from the reconstructed composition, the *reaction mapping*, $\mathbf{R}(\mathbf{z}^{DR}(0), t)$ in the full composition space is more concisely denoted by $\mathbf{z}^{DR}(t)$ (see Fig.2.4).

Now for a representative test problem, to estimate the errors incurred using a dimension reduction method, a number of test compositions are selected in the full space. Let the number of test compositions used be denoted by, N. We perform a PaSR computation in the *full composition space* (without dimension reduction) and then pick N distinct test compositions in the full space denoted by $\mathbf{z}^{(n)}(0)$, for $n = 1$ to N; and their corresponding *reaction mappings* after a fixed constant time t, are denoted by $\mathbf{z}^{(n)}(t)$.

At the N chosen compositions, $\mathbf{z}^{(n)}(0)$, in the full space, and for a given set of *represented* species, $\mathbf{\Phi}^r$, we denote the corresponding reduced representations by $\mathbf{r}^{(n)}(0)$, the reconstructed compositions by $\mathbf{z}^{DR(n)}(0)$, and the reaction mappings by $\mathbf{z}^{DR(n)}(t)$.

Note that, given $\mathbf{z}(0)$ and t, $\mathbf{z}^{DR}(t)$ depends on the specification of the represented species, $\mathbf{\Phi}^r$. As needed, we show this dependence explicitly by the notation $\mathbf{z}^{DR}(t, \mathbf{\Phi}^r)$.

At this stage, we define the error in the *reaction mapping* obtained after time t starting from the reconstructed composition to be

$$\epsilon(t, \mathbf{\Phi}^r) = \frac{[\mathbf{z}^{DR(n)}(t, \mathbf{\Phi}^r) - \mathbf{z}^{(n)}(t)]_{rms}}{[\mathbf{z}^{(n)}(t)]_{rms}}, \tag{2.7}$$

where the operator $[\,]_{rms}$ is defined by, for example,

$$[\mathbf{z}^{(n)}(t)]_{rms} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}||\mathbf{z}^{(n)}(t)||^2}, \tag{2.8}$$

where $||\mathbf{z}||$ denotes the 2-norm.

In particular we have two important errors in the dimension reduction method corresponding to $t = 0$ and $t = \Delta t$:

1. **Species Reconstruction Error**: This is the error in reconstructing the full composition given a reduced composition $\mathbf{r}(0)$ at $t = 0$ and is given by Eqn.2.7 as $\epsilon(0, \mathbf{\Phi}^r)$.

2. **Reaction Mapping Error**: This is the error in the reaction mapping obtained after time step Δt (reaction time step) starting from the reconstructed composition and is equal to $\epsilon(\Delta t, \mathbf{\Phi}^r)$.

Both the species reconstruction and reaction mapping errors depend on the choice of represented species, $\mathbf{\Phi}^r$, and the goal of this work is to identify a "good" set of represented species which reduces these errors.

2.4 Selection of Optimal Species

2.4.1 Optimal Species

In the previous section we looked at the various errors involved in the dimension reduction method, and with the goal of reducing these errors for accurate implementation of dimension reduction in combustion chemistry, here we propose a definition for optimal species based on the RCCE dimension reduction method.

The definition of the optimal set of species is based on the dimension reduction error, $\epsilon(t, \mathbf{\Phi}^r)$. We consider either the species reconstruction error, $\epsilon(0, \mathbf{\Phi}^r)$ or the reaction mapping error, $\epsilon(\Delta t, \mathbf{\Phi}^r)$; and the error used for defining the optimal set is concisely denoted by $\epsilon(\mathbf{\Phi}^r)$.

For a given definition of the error $\epsilon(\mathbf{\Phi}^r)$, and a given value of n_{rs}, the set of n_{rs} represented species, $\mathbf{\Phi}^r$ which minimizes the error, $\epsilon(\mathbf{\Phi}^r)$, is defined to be the *optimal* set of species and is denoted by $\mathbf{\Phi}^{opt}$.

The optimal set of species, $\mathbf{\Phi}^{opt}$, (by definition) produces the minimum error:

$$\epsilon(\mathbf{\Phi}^{opt}) = \min_{\mathbf{\Phi}^r} \epsilon(\mathbf{\Phi}^r). \tag{2.9}$$

2.4.2 Objectives of Species Selection

Ideally one wants to find the *optimal* set of represented species for implementing dimension reduction. For a given value of n_{rs}, a simple brute-force method for selecting the *optimal* set of species is to form all possible $\binom{n_s}{n_{rs}}$ number of sets of represented species, and then compute the error, $\epsilon(\mathbf{\Phi}^r)$, for each of the sets. The set of species producing the minimum error is the required optimal set. Such a brute-force method involves $O\binom{n_s}{n_{rs}}$ number of computations, which can become very expensive even at small values of n_{rs} and n_s (for example, for $n_s = 30$ and $n_{rs} = 10$, over 30 million sets of represented species can be formed) and hence this brute-force method is in general impracticable to use.

An alternative approach is to use a *greedy algorithm*. A greedy algorithm proceeds in stages, making a locally optimal choice at each stage to find a near-optimal solution [20]. Greedy algorithms are shortsighted in their approach, making one greedy choice at a time without worrying about the consequences of such a choice in the future. In other words, a greedy algorithm never reconsiders its choices. Greedy algorithms are not guaranteed to give the optimal solutions, but provide good solutions for many mathematical problems.

A "good" set (initially empty) of represented species denoted by $\mathbf{\Phi}^g$ is formed in stages using a greedy algorithm by selecting at each stage the species whose addition to the set produces the minimum dimension reduction error.

The idea is, for a given value of n_{rs}, to select a set of n_{rs} "good" species, proceed in n_{rs} stages, from $1, 2 \ldots n_{rs}$, selecting the best species at each stage, i.e., the species which minimizes the error. So, at Stage 1, pick the first best species from $\mathbf{\Phi}$ corresponding to the minimum error. Next, at Stage 2 pick the next best species from the remaining set of species which minimizes the error, and continue until n_{rs} species are selected.

At each stage S (for $S = 1 \ldots n_{rs}$) of this algorithm, $(n_s - S + 1)$ number of computations are performed. Overall in n_{rs} stages only $O(n_s \, n_{rs})$ number of computations are performed and hence this method is economical.

Moreover, in the implementations of RCCE and ICE-PIC it is often desirable to start working with a given set of represented species, and if required to add more species to the existing set. For such a purpose, the greedy algorithm is ideal, as it selects the best *available* represented species from the remaining set of species.

In the next section we formally describe this automated greedy algorithm, and then present results.

2.4.3 Greedy Algorithm for Species Selection

Notation

Here we define certain terms and quantities used to describe the algorithm:

- **Determined species**: In a given chemical system with a specified number of moles of elements, the species whose concentration can be determined by atom conservation alone are called the determined species. Obviously such species are not good choices for represented species as they are already determined.

- **Unrepresented determined species ($\mathbf{\Phi}^{ud}$)**: For a given set of represented species, $\mathbf{\Phi}^r$, the set of other species whose concentration can be determined by element conservation alone are called the unrepresented determined species and are denoted by $\mathbf{\Phi}^{ud}$. (There may be no such species, in which case $\mathbf{\Phi}^{ud}$ is the empty set.)
- **Unrepresented undetermined species ($\mathbf{\Phi}^{uu}$)**: The set of unrepresented species which are not determined are called the unrepresented undetermined species and are denoted by $\mathbf{\Phi}^{uu}$. (If there are no such species, i.e., all the species are either represented or determined, then $\mathbf{\Phi}^{uu}$ is the empty set.)

The greedy algorithm presented in the next section, selects at each stage a good species (producing minimum dimension reduction error) from the set of unrepresented undetermined species, $\mathbf{\Phi}^{uu}$ to form a good set of represented species, $\mathbf{\Phi}^g$.

Greedy Algorithm

The greedy algorithm is described below for finding the entire species ordering i.e., until the set $\mathbf{\Phi}^{uu}$ is empty, based on the defined error $\epsilon(\mathbf{\Phi}^r)$.

1. The n_s species in set $\mathbf{\Phi}$ are assigned indices $1, 2, \ldots n_s$ in an arbitrary order. We use the notation *species* k to denote the species with index k.

2. The algorithm proceeds in S stages, numbered $1, 2, \ldots S$ where S is at most $n_s - n_e$.

3. At the end of the j^{th} stage, there are j "good" represented species selected by the algorithm, which form the represented set $\mathbf{\Phi}_j^g$.

4. Initially, before the beginning of stage 1, the set $\mathbf{\Phi}_0^g$ is initialized to an empty set.

5. At the beginning of the j^{th} stage, based on the set of represented species from the previous stage, $\mathbf{\Phi}_{j-1}^g$, the set $\mathbf{\Phi}_j^{uu}$ of unrepresented undetermined species is identified. If this set is empty, then the algorithm terminates. Let the set of indices of species in $\mathbf{\Phi}_j^{uu}$ be denoted by $\mathbf{I}_j^{uu}$.

6. In the j^{th} stage, another *species* m_j for $m_j \in \mathbf{I}_j^{uu}$ is identified to be added to $\mathbf{\Phi}_{j-1}^g$ to form $\mathbf{\Phi}_j^g$.

   - For each *species* k ($k \in \mathbf{I}_j^{uu}$), $\mathbf{\Phi}_{j,k}^g$ denotes the union of $\mathbf{\Phi}_{j-1}^g$ and *species* k from $\mathbf{\Phi}_j^{uu}$.

   - For each set $\mathbf{\Phi}_{j,k}^g$ the defined error $\epsilon(\mathbf{\Phi}^r)$ is computed as

$$\epsilon_{jk} = \epsilon(\mathbf{\Phi}_{j,k}^g). \tag{2.10}$$

7. The selected species $m_j \in \mathbf{I}_j^{uu}$ is that which minimizes the error, i.e.,

$$\epsilon_{jm_j} \leq \epsilon_{jk} \,\text{for all}\, k \in \mathbf{I}_j^{uu}. \tag{2.11}$$

8. The species m_j is added to the set of good represented species, $\mathbf{\Phi}_{j-1}^g$ to yield

$$\mathbf{\Phi}_j^g = \mathbf{\Phi}_{j-1}^g \cup \{\text{species}\, m_j\}. \tag{2.12}$$

9. The value of j is incremented, and the next stage is started at (5).

At the end of the algorithm, the ordered set $\mathbf{\Phi}^g$ presents a good choice of represented species for dimension reduction methods. For implementing dimension reduction at any given value of n_{rs}, the first n_{rs} number of species from the ordered set $\mathbf{\Phi}^g$ are used as the represented species.

Note that for any given reduced dimension n_{rs}, the above algorithm does not give the optimal set of species, $\mathbf{\Phi}^{opt}$, which minimizes the global error at that value of n_{rs}, but incrementally adds the best *available* species at each stage to the set of represented species computed from the previous stage.

2.5 Results

2.5.1 Greedy Algorithm Results

The greedy algorithm presented in the previous section is applied on the set of species (2.5) to obtain the species ordering based on the species reconstruction and reaction mapping errors. A total of $N = 2500$ test compositions are used in the full space to compute errors. The justification for choosing this value of N and the sensitivity of results to changes in N are discussed later in Section 2.5.3.

The first three stages of the algorithm for species selection based on the species reconstruction error are illustrated in Fig.2.5. At each stage j (for $1 \leq j \leq 3$), the error ϵ_{jk} (2.10) resulting from the addition of species k (for each $k \in \mathbf{I}_j^{uu}$) to the set of represented species from the previous stage, $\mathbf{\Phi}_{j-1}^g$ is plotted. The species producing the minimum error is selected at each stage. As N_2 is the only species in (2.5) containing nitrogen, it is a determined species and hence is

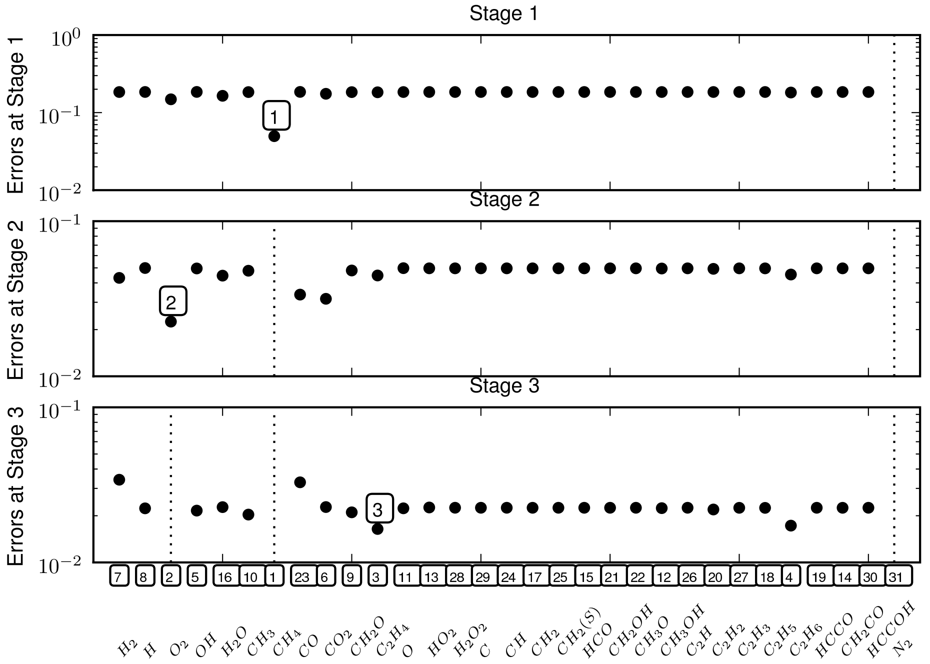


Figure 2.5: Illustration of the first three stages of species selection based on the species reconstruction error using the greedy algorithm. At each stage the species producing the minimum error is selected. At each stage, the determined and already selected species are not included in the selection, and are marked by a dashed line. The numbering on the x-axis shows the final species ordering obtained at the end of the algorithm.

not considered for selection. At each stage the determined and already-selected species, which are not part of the unrepresented undetermined species set, are marked with a dotted line. The x-axis labels show the entire species ordering obtained at the end of the algorithm. At stage 1, we see that the species CH_4 produces 25% less species reconstruction error than other species, when used as the represented species for dimension reduction, and hence CH_4 is selected by the greedy algorithm as the first "best" represented species, $\mathbf{\Phi}_1^g = \{CH_4\}$. At stage 2, we pick the second species which when used along with the previously selected good species, CH_4, produces the minimum species reconstruction error, and as we see the species O_2 with CH_4 produces the minimum error, and

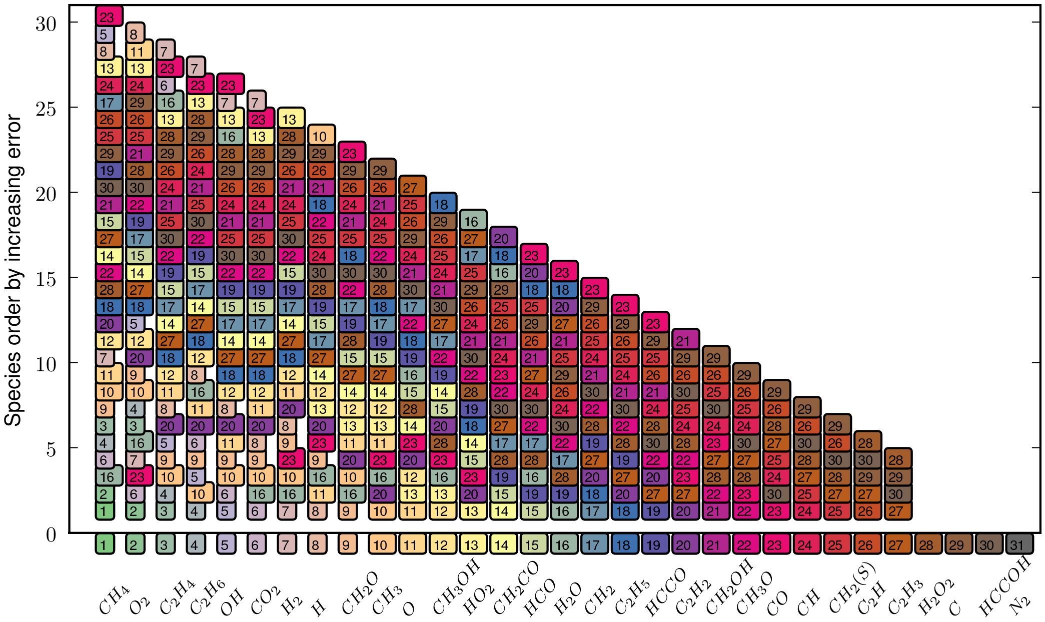


Figure 2.6: Illustration of the working of the greedy algorithm for species selection based on the species reconstruction error. The x-axis shows the stages (1 to $n_s = 31$) for the 31 species present in the GRI-Mech 1.2 methane mechanism. At each stage, the unrepresented undetermined species are plotted with increasing error (from bottom to top) and the species producing the minimum error is selected, which is marked on the x-axis. The algorithm stops at stage 27, when $n_{rs} = n_s - n_e = 27$ because thereafter the species concentration are determined using element conservation. The species ordering of the last four species is inconsequential.

so O_2 is the second species selected by the greedy algorithm, $\mathbf{\Phi}_2^g = \{CH_4, O_2\}$. At stage 3, we pick the third species which when used with the two previously selected species, i.e., $\{CH_4,\ O_2\}$ produces the minimum species reconstruction error, which is found to be the species C_2H_4, and hence is selected in the third stage, $\mathbf{\Phi}_3^g = \{CH_4, O_2, C_2H_4\}$. The algorithm continues in this fashion until the required number of represented species are selected.

The full species ordering based on the species reconstruction error is shown

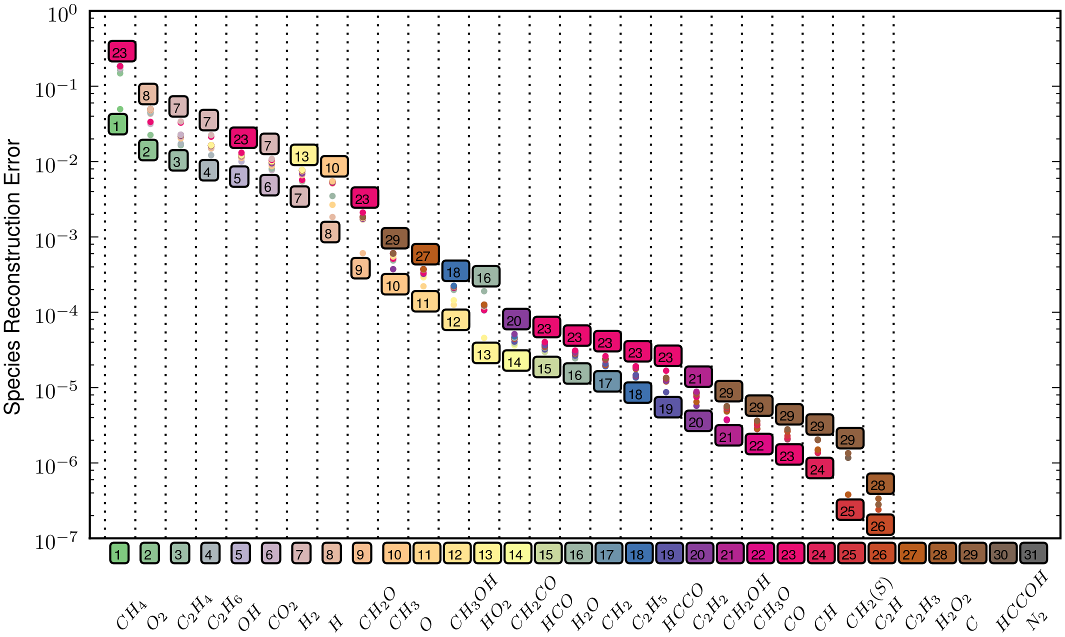


Figure 2.7: Plot showing the range of species reconstruction errors produced by the unrepresented undetermined species at each stage of the species selection using the greedy algorithm. The x-axis shows the stages (1 to $n_s = 31$) for the 31 species present in the GRI-Mech 1.2 methane mechanism. At each stage, the species reconstruction error produced by adding each unrepresented undetermined species, is marked with a dot, and the species that produce the minimum and maximum errors are numbered. The species which minimizes the error is selected at each stage.

in Fig.2.6. The figure illustrates the stage by stage selection of the best available species using the greedy algorithm. At each stage j, the ordering of the unrepresented undetermined species, $\mathbf{\Phi}_j^{uu}$ is shown, based on increasing species reconstruction error ϵ_{jk} (2.10) from bottom to top. The x-axis labels list the best species selected at the end of each stage, which corresponds to the bottom most species (which minimizes the error) appearing in the list at that stage. One important observation we make is that at each stage the unrepresented undetermined species are reordered significantly from the previous stage especially in the initial stages. For example, the species selected in stages 5 to 8 appear at the top of the ordering (produce high error) in the initial stages. The primary reason for this reordering is that the error at every stage depends on all the species selected in the previous stages, and as a consequence the species selected at each stage controls the error in the subsequent stages. As a result, the species selected in the first few stages of the algorithm have a significant effect on the rest of the species ordering.

Fig.2.7 shows the corresponding species reconstruction error values, ϵ_{jk} (2.10) at each stage j. The error values, ϵ_{jk} are marked with a dot, and for clarity, only the species which produce the minimum and maximum error are numbered. We see that the range of errors at each stage is very narrow, except at stages 8, 9 and 13 and the minimum error decreases monotonically with every stage. At every stage, we also observe that certain species, if selected, result in an increase in the species reconstruction error. This behavior is analyzed further in the next section.

Fig.2.8 and Fig.2.9 show the same plots for species selection based on the reaction mapping error. While we make similar observations in Fig.2.8 as for the previous case, Fig.2.9 shows some interesting behavior after stage 16, where we see that the minimum reaction mapping error stays constant and then increases slightly at around stage 23 and 24 and then drops down again at stage 26. The species 28 also consistently shows high errors after stage 16. It appears as if the solution is “trapped” in a local minimum, where addition of any more species does not result in any further decrease in error. This may be a shortcoming of using the greedy algorithm, which picks the locally optimal species at each stage without reconsidering the previous choices. As a consequence, the greedy

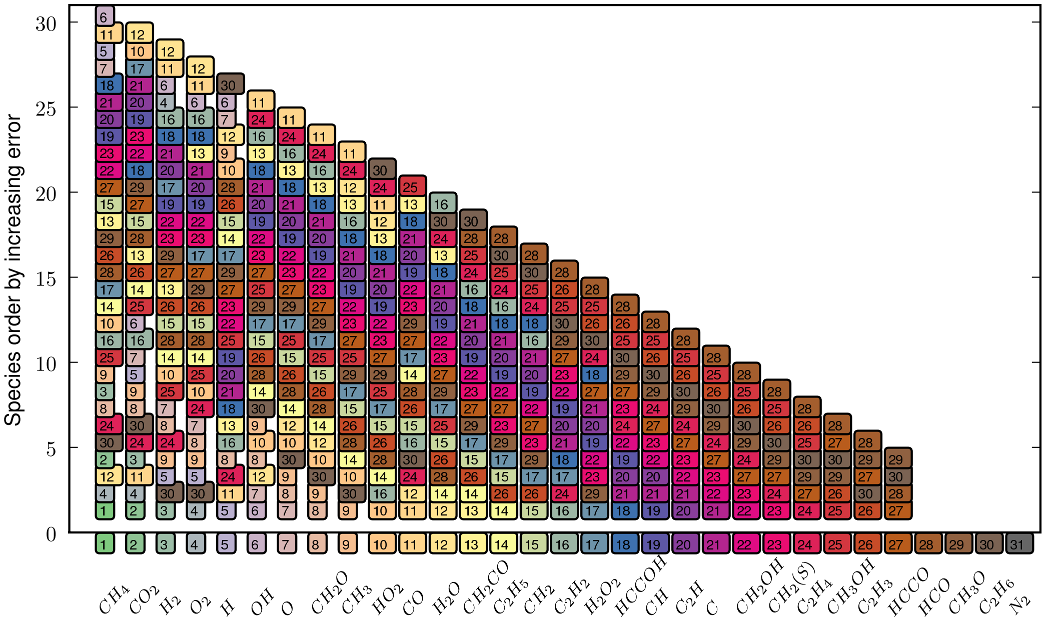


Figure 2.8: Illustration of the working of the greedy algorithm for species selection based on the reaction mapping error. The x-axis shows the stages (1 to $n_s = 31$) for the 31 species present in the GRI-Mech 1.2 methane mechanism. At each stage, the unrepresented undetermined species are plotted with increasing error (from bottom to top) and the species producing the minimum error is selected, which is marked on the x-axis. The algorithm stops at stage 27, when $n_{rs} = n_s - n_e = 27$ because thereafter the species concentration are determined using element conservation. The species ordering of the last four species is inconsequential.

algorithm can get trapped in a local minimum and may give a sub-optimal solution.

Nevertheless, from these results we can draw some important conclusions:

1. The greedy algorithm has been successfully implemented, and is able to pick the best available species at each stage.
2. At least up to $n_{rs} = 16$ (for the 31 species methane GRI-Mech 1.2 mecha-

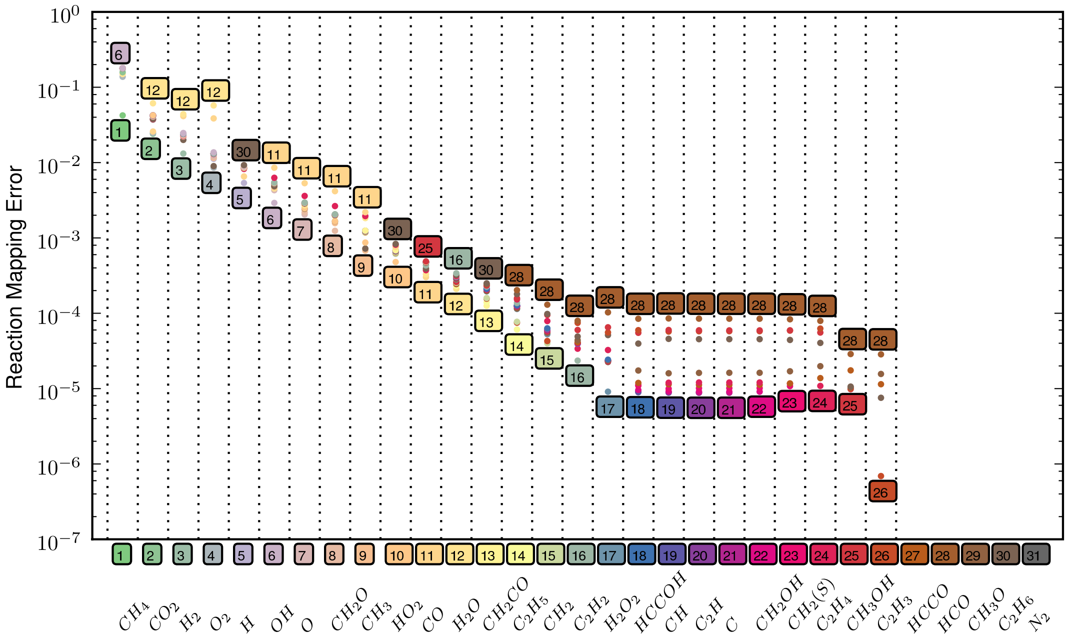


Figure 2.9: Plot showing the range of reaction mapping errors produced by the unrepresented undetermined species at each stage of the species selection using the greedy algorithm. The x-axis shows the stages (1 to $n_s = 31$) for the 31 species present in the GRI-Mech 1.2 methane mechanism. At each stage, the species reconstruction error produced by adding each unrepresented undetermined species, is marked with a dot, and the species that produce the minimum and maximum errors are numbered. The species which minimizes the error is selected at each stage.

nism), the error decreases with each added represented species using the greedy algorithm.

3. The error decreases by more than 30% in the first two stages, and on average, the error decreases by 8% with each added represented species in the first 16 stages.

4. In order to achieve 1% and 0.1% levels of error, approximately 5 and 10 represented species are required, respectively.

2.5.2 Worst Case Scenario

In this section, to stress more on the importance of carefully selecting the represented species for dimension reduction, we perform a series of worst case analysis by computing errors incurred when using a bad set of species compared to using the species obtained from our algorithm.

To pick the "worst" species, we again use the same greedy algorithm described in Section 2.4.3 with the exception that at every stage instead of picking the unrepresented species which minimizes the error, we pick the species which maximizes the error. From here on, we refer to this "worst" species selection algorithm as the *greedy-worst* algorithm.

Fig.2.10 shows the worst species ordering obtained using the *greedy-worst* algorithm based on the reaction mapping error. It can be clearly seen that the error for this worst ordering of species remains almost constant with increase in the dimension, n_{rs}. Also the errors are orders of magnitude more than the errors obtained with the species ordering using our species selection algorithm.

We see that the major species – CH_4, O_2, CO_2 and H_2O – are the last species selected by the previous method. To check if manually including the major species first improves the errors, we performed another worst-case test in which we first manually selected the major species in the system – CH_4, O_2, CO_2 and H_2O – and then picked the rest of the worst species using the *greedy-worst* algorithm. The species ordering for this case based on the reaction mapping error is shown in Fig.2.11. In this cases also, we see that including the major species in the represented set does not solve the problem fully; the errors still stay very high with the worst species ordering.

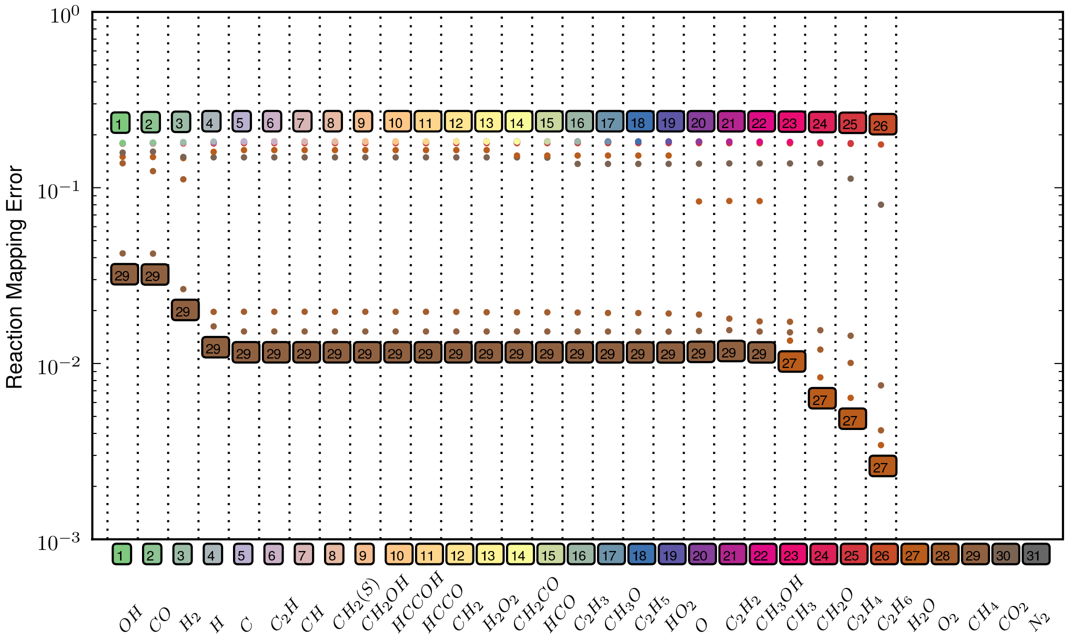


Figure 2.10: Plot showing the reaction mapping errors produced by selecting the worst species, i.e. species with maximum error at each stage of the greedy algorithm. At each stage, reaction mapping error in all the unrepresented undetermined species are marked with a dot, and the species which produce the minimum and maximum errors are numbered. The species which maximizes the error is selected at each stage, which is marked on the x-axis.

We can draw some important conclusions from these worst case scenarios:

1. Increase in the number of represented species (or, equivalently, the dimension of the reduced space) does not necessarily result in a reduction of dimension reduction error.
2. A *bad* set of species can result in an error which is orders of magnitude greater than the error resulting from a well chosen set of species.
3. Including major species does not always help; the errors can still remain very high if the rest of the species are not well selected.

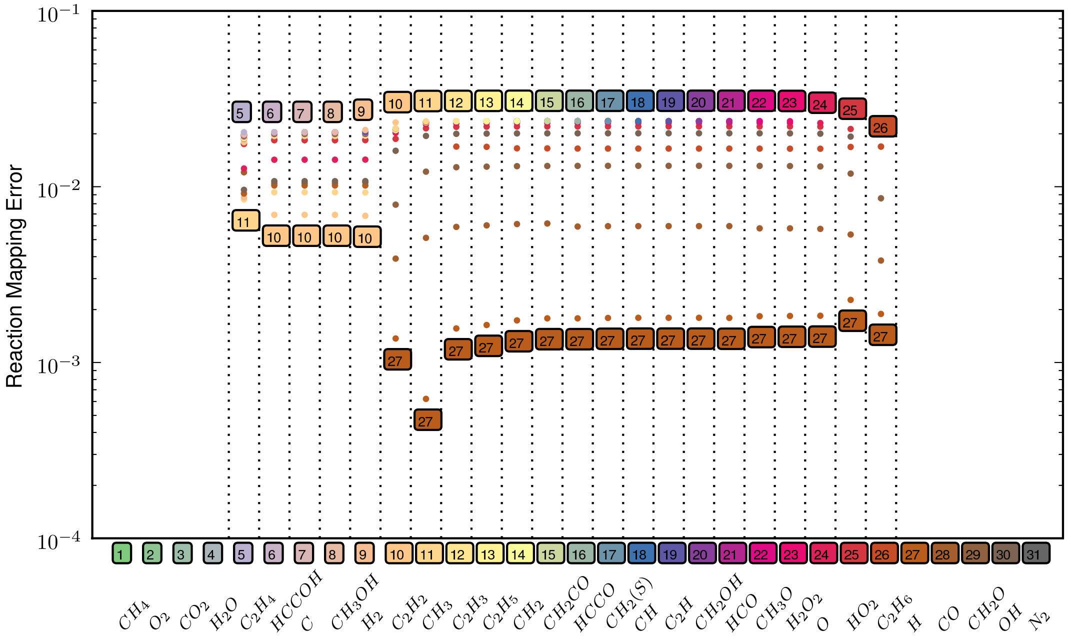


Figure 2.11: Plot shows the reaction mapping errors produced by selecting the worst species, i.e. species with maximum error at each stage of the greedy algorithm, with the major species: CH_4, O_2, CO_2 and H_2O manually fixed first in the ordering. At each stage, the reaction mapping error in all the unrepresented undetermined species are marked with a dot, and the species which produce the minimum and maximum errors are numbered. The species which maximizes the error is selected at each stage, which is marked on the x-axis.

2.5.3 Sensitivity Tests

In this section we perform a series of tests to investigate the sensitivity of the species ordering to changes in the number of test compositions, N, used to compute errors; changes in various testing conditions like the pressure, initial temperature and the equivalence ratio; changes in the definition of the error, ϵ; and also to analyze how strongly the dimension reduction results depend on the choice of represented species.

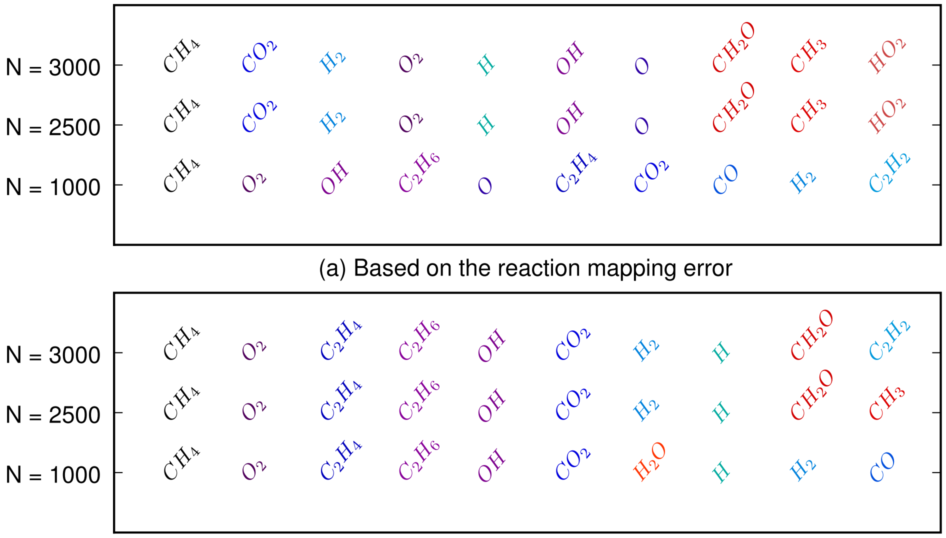
Choice of test compositions

To compute accurate species orderings using the greedy algorithm, the errors involved need to be computed accurately. Since the errors are computed on the N chosen test compositions, the choice of test compositions is crucial.

At each stage of the greedy algorithm, for each candidate species in the unrepresented undetermined species set, the errors are computed at all the chosen N testing compositions to find the species producing the minimum error. Selecting a large number of testing compositions makes the algorithm expensive and too small a value of N may not give accurate species orderings.

The species selected in the initial stages of the greedy algorithm are crucial as, in all the results presented so far, we see that the species reconstruction and the reaction mapping errors drop rapidly to below 10^{-2} in the first 8 to 10 species. Hence, N is chosen high enough such that the first 8 to 10 species in the species ordering remain unchanged with any further increase in N.

Species orderings obtained with increasing N at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ is shown in Fig.2.12. From Fig.2.12(a) it is seen that the first 10 species are identical with $N = 2500$ and $N = 3000$; whereas with $N = 1000$ only the first species is the same. From Fig.2.12(b) we see that the tenth species differs between $N = 2500$ and $N = 3000$; whereas with $N = 1000$ there are three species which differ. At other testing conditions also, the species ordering results (not presented here for brevity) show that the first 8 to 10 species remain unchanged with $N \geq 2500$ test compositions. Hence all the results in this paper are presented with $N = 2500$ test compositions.



(a) Based on the reaction mapping error

(b) Based on the species reconstruction error

Figure 2.12: Displayed are the first ten species selected using the greedy algorithm based on (a) the reaction mapping error and (b) the species reconstruction error using the PaSR at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ with increasing number of test compositions, N, used to compute the errors.

Table 2.1: The following set of testing conditions (overall 12 combinations) are considered for the PaSR tests.

Equivalence Ratio (ϕ)	1	0.8	1.2
Pressure (atm)	1	10	
Initial Temperature (K)	600	1200	

Table 2.2: Displayed are the first ten species selected by the greedy algorithm based on the reaction mapping error at all the 12 testing conditions listed in Table 2.1.

	$\phi = 1.0$	$\phi = 1.0$	$\phi = 1.0$	$\phi = 1.0$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 1.2$	$\phi = 1.2$	$\phi = 1.2$	$\phi = 1.2$
	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$
	$p = 1$	$p = 1$	$p = 10$	$p = 10$	$p = 1$	$p = 1$	$p = 10$	$p = 10$	$p = 1$	$p = 1$	$p = 10$	$p = 10$
1	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4
2	CO_2	CO_2	CO	CO_2	CO_2	CO_2	C_2H_6	C_2H_6	O_2	CO_2	O_2	CO_2
3	H_2	H_2	C_2H_6	CH_2O	H_2	H_2	OH	CO	CH_3	H_2	C_2H_6	H_2
4	O_2	H	H_2	C_2H_6	O_2	O_2	C_2H_2	CH_2O	C_2H_6	C_2H_6	CO_2	C_2H_6
5	H	C_2H_6	C_2H_4	H	C_2H_4	C_2H_4	CH_2CO	H_2	CO_2	O	CH_2O	CH_2O
6	OH	O	CH_2O	H_2	CH_2O	CH_2O	$HCCOH$	O	H_2O	CH_2O	H_2	OH
7	O	CH_2O	CH_3	O	CH_3OH	CO	CH	CH_3	H	CH_3OH	CH_3	O_2
8	CH_2O	OH	HO_2	CH_3	HO_2	O	C	CH_3OH	CH_2O	CH_2	C_2H_2	HO_2
9	CH_3	O_2	OH	C_2H_2	C_2H_2	OH	C_2H	C_2H_2	OH	C_2H_3	C_2H_4	CO
10	HO_2	HO_2	H	CH_2CO	CH_2CO	C_2H_2	$HCCO$	CH_2CO	C_2H_2	CH	C_2H_3	O

Table 2.3: Displayed are the first ten species selected by the greedy algorithm based on the species reconstruction error at all the 12 testing conditions listed in Table 2.1.

	$\phi = 1.0$	$\phi = 1.0$	$\phi = 1.0$	$\phi = 1.0$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 0.8$	$\phi = 1.2$	$\phi = 1.2$	$\phi = 1.2$	$\phi = 1.2$
	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$
	$p = 1$	$p = 1$	$p = 10$	$p = 10$	$p = 1$	$p = 1$	$p = 10$	$p = 10$	$p = 1$	$p = 1$	$p = 10$	$p = 10$
1	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4	CH_4
2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2	O_2
3	C_2H_4	C_2H_4	C_2H_4	C_2H_6	C_2H_4	C_2H_4	C_2H_6	C_2H_6	C_2H_4	C_2H_4	H_2O	C_2H_6
4	C_2H_6	OH	C_2H_6	C_2H_4	C_2H_6	OH	C_2H_4	C_2H_4	CO_2	C_2H_6	CO	C_2H_4
5	OH	C_2H_6	CO_2	CH_2O	OH	C_2H_6	CO_2	CH_2O	H_2	CH_3	H_2	CO_2
6	CO_2	CH_3	H_2	H_2O	CO_2	O	H_2	OH	H	CO_2	CO_2	H_2
7	H_2	H	CH_2O	CO	H_2	H	CH_2O	CO_2	CH_2O	H_2O	OH	H
8	H	O	H	CH_3	H	CH_3	H	H_2	OH	H	C_2H_6	CH_2O
9	CH_2O	CO_2	OH	OH	CH_2O	CO_2	OH	O	C_2H_2	H_2	H	OH
10	CH_3	CH_2O	H_2O	H	O	CH_2O	HO_2	HO_2	CH_3	CH_3OH	C_2H_4	HO_2

Sensitivity to changes in testing conditions

To investigate the species ordering sensitivity to changes in PaSR testing conditions, the species ordering are obtained at all of the 12 conditions listed in Table 2.1. The species orderings (first 10 species) obtained for these cases based on the reaction mapping error are listed in Table 2.2 and based on species reconstruction error are shown in Table 2.3.

Next using these species ordering we performed two sets of tests:

1. Different PaSR tests at fixed species ordering:
   We pick a species ordering obtained at a particular testing condition, and then using this species ordering we perform PaSR tests at all the conditions listed in Table 2.1 with dimension reduction and analyze the reaction mapping and species reconstruction errors at various values of n_{rs}. We are interested in determining whether the error in all these cases decreases monotonically or if it shows some irregular trends.
2. Fixed PaSR test with different species orderings:
   In this we perform a fixed PaSR test with dimension reduction using a selected number of species orderings i.e, different sets of represented species (obtained at different testing conditions listed in Table 2.1) and analyze how the reaction mapping and species reconstruction errors vary with different choices of represented species.

Since we are more interested in the reaction mapping error which determines the error in the represented species concentration at the end of the reaction time step Δt, we perform more tests based on the reaction mapping error than on the species reconstruction error.

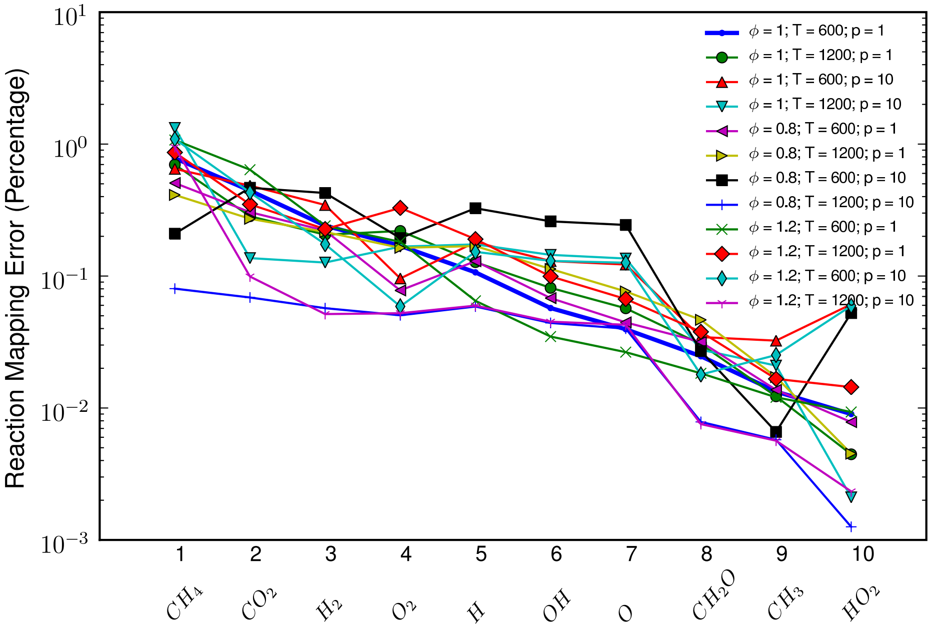


Figure 2.13: Plot of the reaction mapping error in PaSR tests performed at all the 12 conditions listed in Table 2.1 with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from the species ordering obtained based on the reaction mapping error at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$.

PaSR tests at fixed species ordering

We pick the species ordering based on the reaction mapping error obtained using the PaSR test at an equivalence ratio, $\phi = 1$, initial temperature, $T = 600K$ and pressure $p = 1\,atm$. Using this species ordering we perform PaSR tests with dimension reduction for $1 \leq n_{rs} \leq 10$ at all the conditions listed in Table 2.1. For each value of n_{rs}, the first n_{rs} number of species from the chosen species ordering are used as represented species.

The reaction mapping error obtained at each value of n_{rs} is plotted in Fig.2.13. We make the following observations:

Table 2.4: Displayed are the species orderings (first 10 species) obtained using the greedy algorithm based on the reaction mapping error at five selected testing conditions (from Table 2.1) with the four major species: CH_4, O_2, CO_2 and H_2O manually fixed first in the species ordering.

	$\phi = 1.0$	$\phi = 1.0$	$\phi = 1.0$	$\phi = 0.8$	$\phi = 1.2$
	$T = 600$	$T = 1200$	$T = 600$	$T = 1200$	$T = 1200$
	$p = 1$	$p = 1$	$p = 10$	$p = 1$	$p = 1$
1	CH_4	CH_4	CH_4	CH_4	CH_4
2	O_2	O_2	O_2	O_2	O_2
3	CO_2	CO_2	CO_2	CO_2	CO_2
4	H_2O	H_2O	H_2O	H_2O	H_2O
5	CH_3	CH_3	H_2	CH_3	CH_3
6	H	H	CH_2O	OH	H
7	C_2H_6	OH	HO_2	H	OH
8	OH	H_2	CH_3	CH_2O	H_2
9	CH_2O	O	CH_3OH	C_2H_6	HO_2
10	C_2H_2	CO	H_2O_2	HO_2	CH_3OH

1. For a given value of n_{rs}, the error typically varies by a factory of 10 depending on the conditions.

2. While the general behavior is for the error to decrease with increasing value of n_{rs}, there are many instances of the error increasing significantly: for example, in one case ($\phi = 0.8$, $T = 600\,K$ and $p = 10\,atm$), as the value of n_{rs} increases from 9 to 10, the error increases by a factor of 10.

Fixed PaSR test with various species ordering

In the greedy algorithm presented, all the species are treated equally and there are no predefined "major" or "minor" species. But, in practice, it is often desirable to include the major species in the calculations. So at a few selected testing conditions, we obtained species ordering using the greedy algorithm with the major species CH_4, O_2, CO_2 and H_2O fixed first in the ordering. The species ordering obtained (first 10 species) are listed in Table 2.4.

We performed PaSR tests with dimension reduction at $1 \leq n_{rs} \leq 10$ using various species orderings:

- Fig.2.14 shows the reaction mapping error obtained using the PaSR test performed at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ with various species orderings based on the reaction mapping error. We see that the species ordering obtained at the same testing conditions ($\phi = 1$, $T = 600\,K$ and $p = 1\,atm$) produces the lowest error at all values of n_{rs} except at $n_{rs} = 10$. The error with this species ordering decreases by 10% on average with every stage, while at other species orderings the error values are greater and at many instances stay constant with increase in n_{rs}. This case clearly demonstrates that the greedy algorithm is successfully able to select "good" represented species which produce low dimension reduction error.
- Fig.2.15 shows results for the PaSR test performed at a slightly higher temperature of $T = 1200\,K$, $\phi = 1$ and $p = 1\,atm$. Here also we see that the species ordering obtained at the same testing conditions ($T = 1200\,K$, $\phi = 1$ and $p = 1\,atm$) produces the lowest error for $n_{rs} \leq 7$, thereafter the error values are still low but slightly higher than species ordering obtained

with major species fixed first and species ordering obtained at $\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$. This again shows that the greedy algorithm successfully captured the best species in the initial stages, but since the greedy algorithm does not reconsider its choices, it fails to capture the best possible set of represented species at higher values of n_{rs}.

- Fig.2.16 shows results for the PaSR test performed at higher pressure of $p = 10$, $\phi = 1$ and $T = 600\,K$. Here we see that for $n_{rs} \leq 3$ the species ordering obtained at the same testing conditions ($p = 10$, $\phi = 1$ and $T = 600\,K$) produces the lowest error, but soon is out-performed by some of the other species orderings. In the range $5 \leq n_{rs} \leq 9$ the species ordering with major species fixed first produces the lowest error. For this case, even though the greedy algorithm does not give the best possible set of species at many values of n_{rs}, the error with the species ordering obtained at the same testing conditions decreases monotonically and remains very close to the lowest error values achieved at all the values of n_{rs}. The other species orderings which produce lower errors, show a highly irregular trend, with the error increasing-decreasing by more than 10% at many values of n_{rs}.

- Fig.2.17 shows the results for lean premixed combustion at $\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$. In this case also we see behavior similar to the previous case: the species ordering obtained at the same conditions ($\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$) produces lowest error for $n_{rs} \leq 3$; remains close to the lowest error achieved at higher values of n_{rs}, and decreases monotonically.

- Fig.2.18 shows the results for a rich premixed combustion at $\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$. In this case, interestingly we see that the species

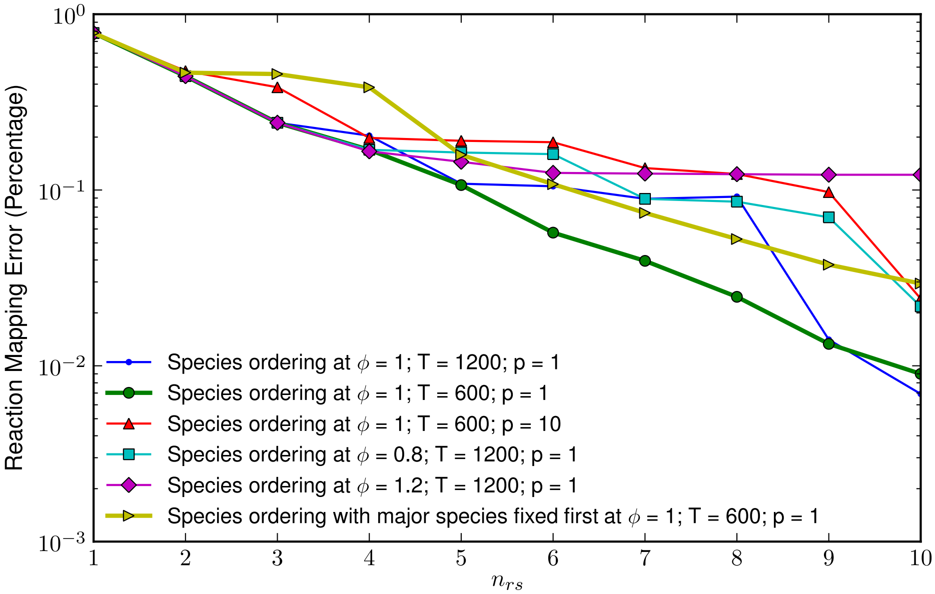


Figure 2.14: Plot of the reaction mapping error in the PaSR test performed at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from various species orderings based on the reaction mapping error, given in Table 2.2 and Table 2.4.

ordering obtained using the greedy algorithm at the same conditions ($\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$) produces the lowest error for $n_{rs} \leq 5$ but thereafter the error decreases at a very slow rate. This again could be a case where the greedy algorithm got "trapped" in a local minimum and as a consequence the error value almost stays constant with any further addition of represented species. The other species orderings including the species ordering with the major species fixed, start with very high errors at low values of n_{rs}, but show some improvement at higher values of n_{rs}, with the error values decreasing but not very significantly.

Apart from the individual observations made in each of the cases, the results

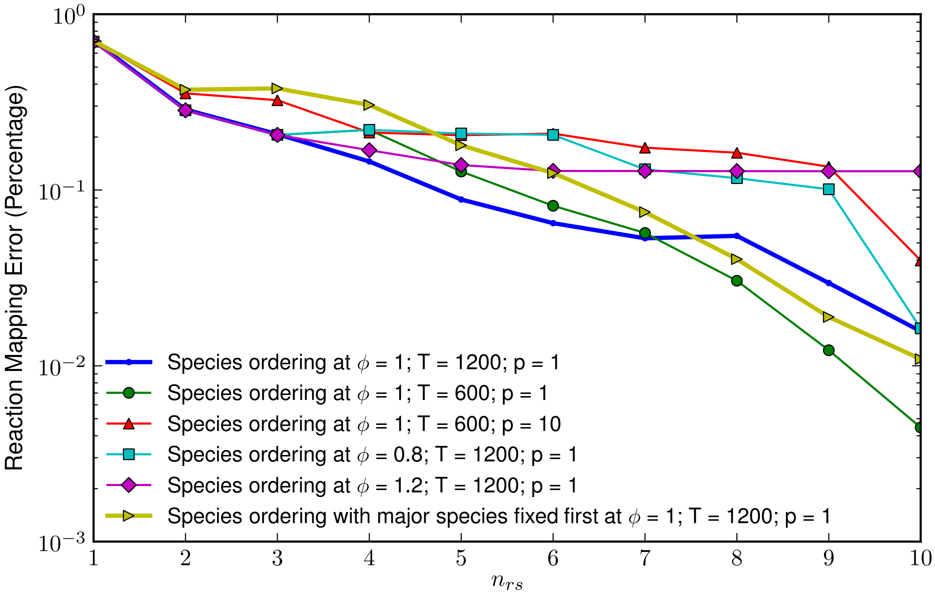


Figure 2.15: Plot of the reaction mapping error in the PaSR test performed at $\phi = 1$, $T = 1200\,K$ and $p = 1\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from various species orderings based on the reaction mapping error, given in Table 2.2 and Table 2.4.

for all the five cases also show the following:

1. The species ordering obtained (without any constrained major species) at the same testing conditions at which the PaSR test is performed, always shows the lowest error at low values of n_{rs}, suggesting that the greedy algorithm successfully picks the best species in the initial stages (as it certainly does on the first stage).
2. The species ordering obtained at the same testing conditions at which the PaSR test is performed, is found to produce a non-increasing error with each addition of represented species for $n_{rs} \leq 10$ (even when no such restrictions are enforced by the algorithm itself), which is not always true

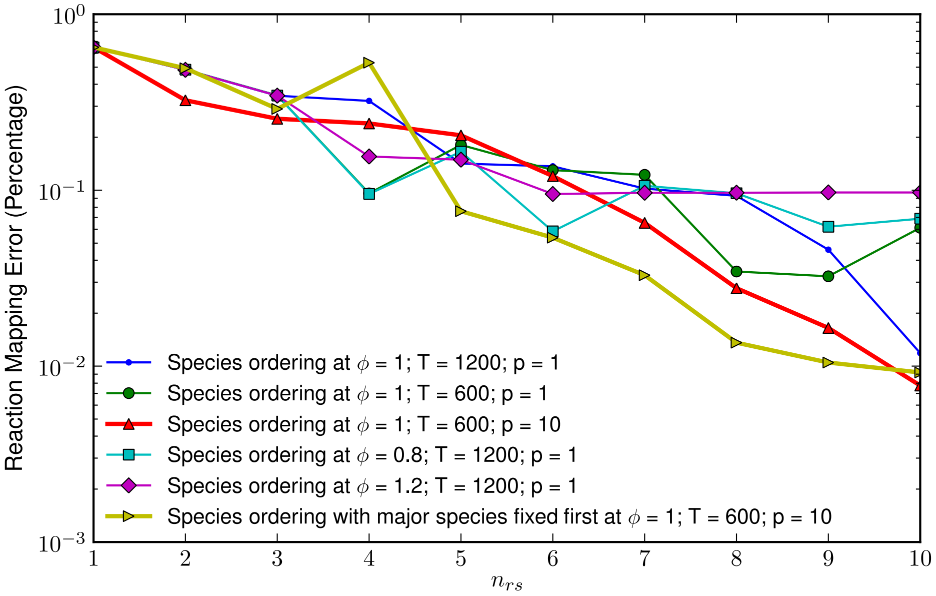


Figure 2.16: Plot of the reaction mapping error in the PaSR test performed at $\phi = 1$, $T = 600\,K$ and $p = 10\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from various species orderings based on the reaction mapping error, given in Table 2.2 and Table 2.4.

with other species orderings.

3. At low values of n_{rs}, the error generated by the species ordering with major species fixed is higher than that generated by other species orderings in all the cases.

4. At high values of n_{rs}, the greedy algorithm does not always give the best possible set of represented species producing the lowest error. But, this is an expected outcome of using a greedy algorithm as it picks the locally optimum species at every stage and is not guaranteed to give the global optimum set of species at all values of n_{rs}.

In short, the results demonstrate that the greedy algorithm (within its own lim-

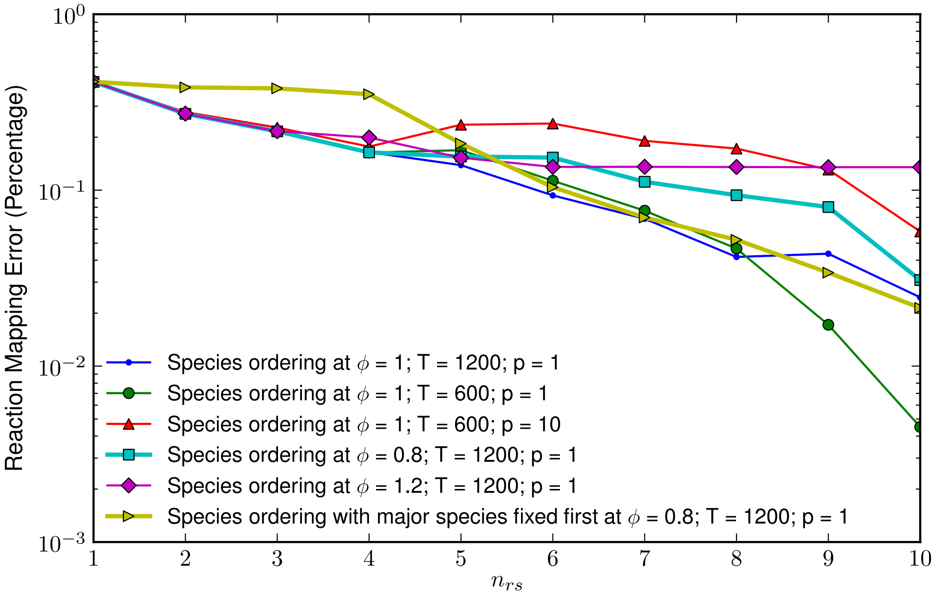


Figure 2.17: Plot of the reaction mapping error in the PaSR test performed at $\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from various species orderings based on the reaction mapping error, given in Table 2.2 and Table 2.4.

itations) is able to produce a "good" set of represented species for a majority of cases tested over a wide range of testing conditions.

Sensitivity to changes in the definition of error

The species ordering obtained using the greedy algorithm depends on the given definition of error, ϵ, i.e., the dimension reduction error that we want to minimize.

The dimension reduction error in the reaction mapping after time t starting from the reconstructed composition, $\mathbf{z}^{DR}(0)$ is given by $\epsilon(t, \mathbf{\Phi}^r)$.

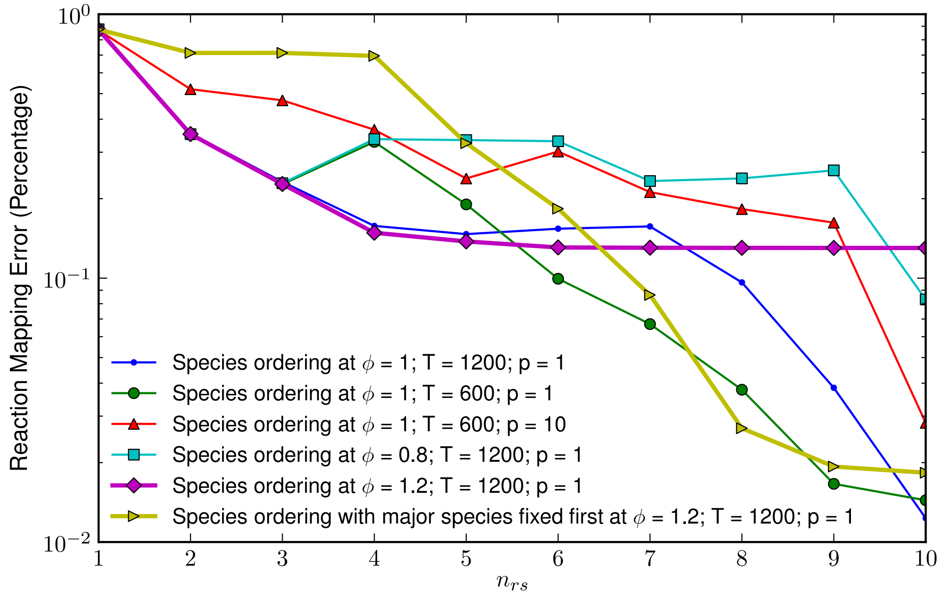


Figure 2.18: Plot of the reaction mapping error in the PaSR test performed at $\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from various species orderings based on the reaction mapping error, given in Table 2.2 and Table 2.4.

We have so far presented results based on two errors:

1. Species reconstruction error, $\epsilon(0, \mathbf{\Phi}^r)$
2. Reaction mapping error, $\epsilon(\Delta t, \mathbf{\Phi}^r)$

We have already seen that the species ordering obtained with these two definitions of error are significantly different, as seen in Fig.2.6 and Fig.2.8, and also in the various species orderings obtained at different testing conditions in Table 2.2 and Table 2.3.

Among the two definitions of the error used, the species reconstruction error is cheaper and easier to compute because it does not involve any ODE integrations. So, to examine how much reaction mapping error is incurred if we use represented species selected based on the species reconstruction error instead of the reaction mapping error, we perform a PaSR test at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ with dimension reduction using represented species from (i) species ordering based on the reaction mapping; and (ii) species ordering based on the species reconstruction error, and then compare the errors for the two cases. The results are shown in Fig.2.19, and we see that the species ordering based on the species reconstruction error results in higher reaction mapping errors than species ordering based on the reaction mapping error, except at $n_{rs} = 1, 2$ and 8 where they produce approximately the same error. Thus, the species ordering based on the species reconstruction error may not work well for problems involving the computation of reaction mappings, but it still provides a quick and systematic way of obtaining a decent set of represented species to start working with.

From the results presented so far, we see that there is no one good definition of the error which will work all the time to select good species using the greedy algorithm. A good definition of the error is more problem-specific, and depends on the specifications of the problem we are working on and what we want to achieve from using the dimension reduction method. If one is investigating a steady state problem involving only species reconstructions, then an error based at $t = 0$ will work best. In reactive flow problems, depending on the reaction-time-step one can choose a specific value of time, t to define the error, or one may also define an error averaged over the time from $t = 0$ to some specific time t. However, note that the greedy algorithm presented in this paper is independent of the definition of the error, and works unchanged with any given definition of the error.

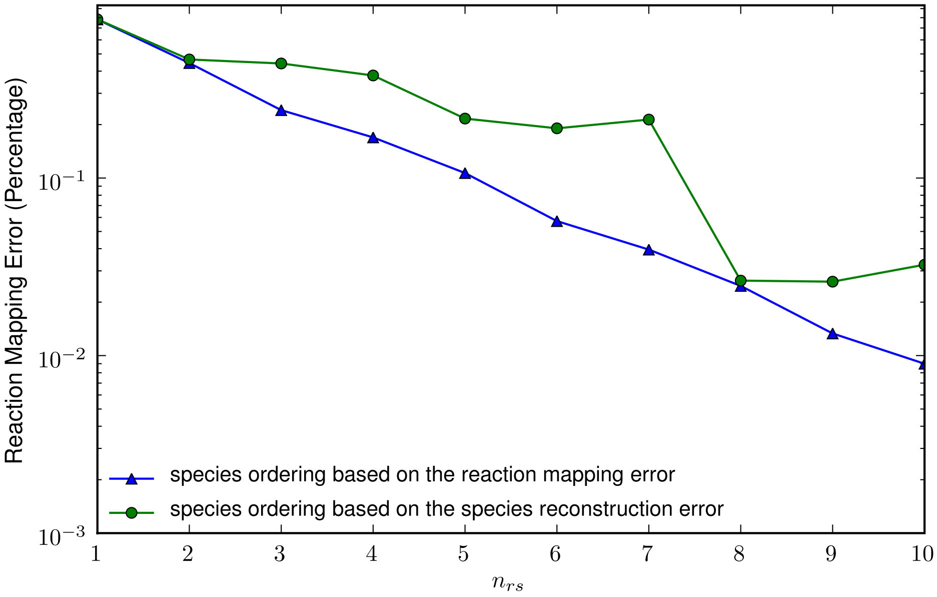


Figure 2.19: Plot of the reaction mapping error in the PaSR test performed at $\phi = 1$, $T = 600\,K$ and $p = 1\,atm$ with dimension reduction (for $1 \leq n_{rs} \leq 10$) using represented species from species orderings obtained at the same conditions ($\phi = 1$, $T = 600\,K$ and $p = 1\,atm$) based on the reaction mapping and species reconstruction errors, given in Table 2.2 and Table 2.3.

2.5.4 Computational Cost

In order to compare the computational cost involved in the dimension reduction steps (species reconstruction and reaction mapping) and to assess the overall cost of the greedy algorithm, the CPU times were measured on a 2.2 GHz Quad-Core AMD Opteron(tm) Processor and are reported in this section.

The average CPU time taken per species reconstruction to compute $\mathbf{z}^{DR}(0) = \mathbf{z}^{CE}(\mathbf{r}(0))$ using CEQ, and the time taken to compute the reaction mapping, $\mathbf{z}^{DR}(\Delta t)$ starting from $\mathbf{z}^{DR}(0)$ using DDASAC are shown in Fig.2.20. We see that solving the full set of ODEs (2.1) using DDASAC to compute the reaction

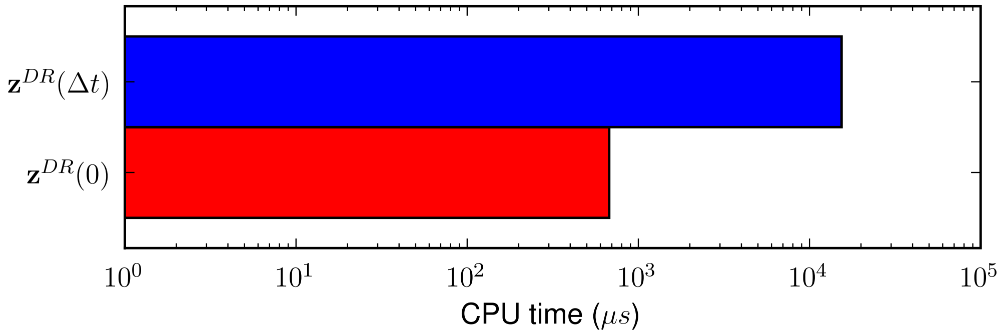


Figure 2.20: Plot showing the average CPU time taken (in micro-seconds) per species reconstruction, i.e., to compute, $\mathbf{z}^{DR}(0)$ given $\mathbf{r}(0)$ using CEQ; and to compute the reaction mapping, $\mathbf{z}^{DR}(\Delta t)$, starting from $\mathbf{z}^{DR}(0)$ using DDASAC. The CPU times are measured on a 2.2 GHz Quad-Core AMD Opteron(tm) Processor.

mapping takes approximately 15 times more CPU time than that required for performing a species reconstruction using CEQ.

The computation of full species ordering for a chemical mechanism involving n_s species using the greedy algorithm on N chosen test compositions based on:

1. the species reconstruction error involves $O(N\, n_s^2/2)$ species reconstructions using CEQ.
2. the reaction mapping error involves computation of $O(N\, n_s^2/2)$ species reconstructions using CEQ and their reaction mappings using DDASAC.

The total CPU time taken to compute the full species ordering based on the species reconstruction error, $\epsilon(0, \mathbf{\Phi}^r)$ and based on the reaction mapping error, $\epsilon(\Delta t, \mathbf{\Phi}^r)$ is shown in Fig.2.21. Since the computation of the reaction mapping alone takes 15 times more CPU time than that taken for the species reconstruction, we see that computing the full species ordering based on the reaction map-

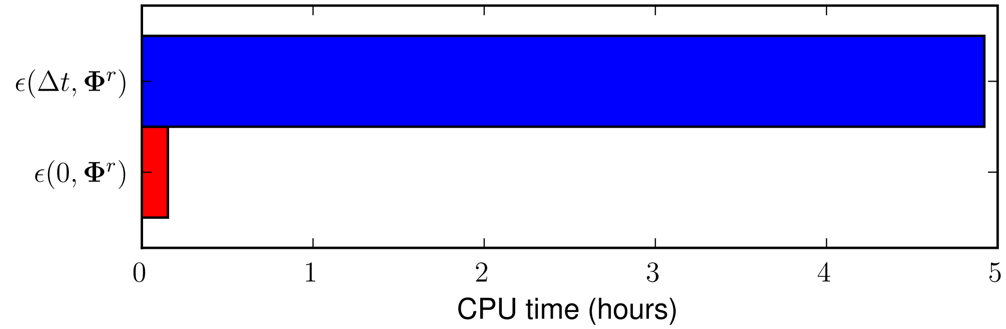


Figure 2.21: Plot showing the total CPU time taken (in hours) by the greedy algorithm to compute the full species ordering for the GRI-Mech 1.2 mechanism involving 31 species based on the species reconstruction error, $\epsilon(0, \mathbf{\Phi}^r)$ and based on the reaction mapping error, $\epsilon(\Delta t, \mathbf{\Phi}^r)$ using $N = 2500$ test compositions. The CPU time is measured by running a serial implementation of the greedy algorithm on a 2.2 GHz Quad-Core AMD Opteron(tm) Processor.

ping error takes approximately 25 times more CPU time than that required for species ordering based on the species reconstruction error.

In Fig.2.21, the CPU timings are reported for computing the full species ordering. In practice, however, we are mostly interested in only the n_{rs} most important species for a dimension reduction with n_{rs} represented species. So, we can stop once n_{rs} number of species are selected using the algorithm, which involves $O(N\, n_s\, n_{rs})$ number of computations. So, for selecting n_{rs} species, with a fixed number of test compositions, N, the algorithm cost increases linearly with the number of species, n_s in the system.

2.6 Comparison with Time Scale Based Methods

As mentioned earlier, the RCCE and ICE-PIC methods are based on the observation that chemical systems involve wide range of time scales, and as a result the reaction trajectories are attracted to a hierarchy of low dimensional slow invariant manifolds (SIMs). The key issue in the RCCE and ICE-PIC methods is to identify the slow time scales or the slowly evolving species (know as represented species in our work) to best approximate the SIMs.

The greedy algorithm described in this paper provides one way of selecting the representing species. One could also select represented species based on a time scale analysis. Though we are unaware of any implementation of RCCE in which species or constraints are selected based on a time scale analysis, but there exist many methods for generating reduced reaction mechanisms based on QSS [103], CSP [52, 94] and ILDM [12], wherein a time scale analysis is used to identify the unimportant species and reactions in the detailed mechanism which are eliminated to obtain reduced mechanisms.

It may be insightful to compare the species ordering obtained by the greedy algorithm with the order in which species are removed in these reduced mechanism generating methods. In [52] an automatic reduction of chemistry with CSP (ARC-CSP) method is developed to generate reduced mechanisms. In this method, CSP is used to analyze perfectly stirred reactor (PSR) data (obtained using a detailed mechanism) to identify and eliminate QSS species from the detailed mechanism to obtain reduced mechanisms. The ARC-CSP method when applied to develop a 10-step reduced mechanism from the GRI Mech 1.2 mechanism (using PSR data from a wide range of operating conditions) identifies and eliminates 17 QSS species in the following order (see Fig. 5 in [52]): $CH_2(S)$, C_2H, HCO, CH_3O, CH, C_2H_5, CH_2OH, CH_2, C_2H_3, H_2O_2, HO_2, C, C_2H_6, CH_3, CH_2CO, $HCCO$, CH_3OH. The 10-step reduced mechanism retains the following 13 species (in no particular order): CH_4, O_2, H_2O, CO_2, O, CO, H_2, OH, H, C_2H_2, C_2H_4, CH_2CO, $HCCOH$ (the inert species N_2 is not considered).

It is interesting to note that only 3 QSS species (CH_3 (9^{th}), HO_2 (10^{th}) and CH_2CO (13^{th})) appear in the first 13 species selected using the greedy algorithm based on the reaction mapping error (which involves reaction kinetics) as seen in Fig.2.9. Also, in the first 10 species selected by the greedy algorithm based on the reaction mapping error at various testing conditions listed in Table 2.2, at most 4 QSS species are seen in each case, and overall 8 out of the 17 QSS species appear in Table 2.2. We note that 5 out of the 8 QSS species appearing in Table 2.2 are at the edge of the 10-step cut-off limit (Fig. 5 in [52]), and as pointed out in [52], these 5 species: CH_3OH, $HCCO$, CH_2CO, CH_3 and C_2H_6 have approximately the same normalized time scales and are hard to distinguish, and hence are treated as a group of QSS species. However, in the greedy algorithm the species are treated individually and are selected based on an error criterion, and hence some of these species could possibly be selected in the early stages of the algorithm if they produce the lowest error.

Finally, we note that the species CH_4 appears 10^{th} in Fig. 5, [52] based on the time scale analysis, and any safety factor of $\alpha < 30$ will generate a reduced mechanism without the CH_4 species. However, as we have seen, the greedy algorithm selects CH_4 as the optimum species (producing the minimum error) at stage 1 in all the cases reported here, and so this clearly shows that a time scale analysis based method for selecting represented species may not work at low values of n_{rs}.

In summary, we do see some similarities between the species ordering obtained using the greedy algorithm with the ordering in which species are removed in time scale based reduced mechanism generating methods. However, we also note that a time scale analysis based method alone may not give "good" represented species for RCCE dimension reduction method, especially at low values of n_{rs}.

2.7 Limitations and Variants of the Greedy Algorithm

Based on the results reported here, we see that the current implementation of the greedy algorithm does not always give a near-optimal set of species, especially at high values of n_{rs}. This may be due to one or more of the following limitations:

1. The greedy algorithm is shortsighted in nature and never reconsiders its choices.
2. The species in the current implementation are treated individually without considering any connections (imposed by the reaction mechanism) between species and treating (strongly connected) species in groups as is done in many reduced mechanism building methods [53, 58].

To address some of these limitations, implementation of the following variants of the greedy algorithm are being considered for future work:

1. Selection-Rejection Strategy:

The inherent flaw in the greedy algorithm is that choices made in the previous stages are never reconsidered. To compensate for this drawback, we can consider a selection-rejection algorithm, in which every 3^{rd} or 4^{th} stage of the algorithm, we reject the worst species from the current set of selected species and resume the algorithm. The cost of rejection goes linearly with number of selected species $O(n_{rs})$, and so is fairly inexpensive and may help in improving the results.

2. Selecting Species in Groups:

   The greedy algorithm selects the best species at every stage, instead a variant of this algorithm would involve selecting species in groups of 2 to 3 species. So considering the case in which we want to select two good species at every stage, then at stage 1 we perform tests with all possible $\binom{n_s}{2}$ number of sets of species and pick the set which produces the minimum error; next at stage 2 we test the remaining $\binom{n_s-2}{2}$ sets of species to pick the next two good species; and so on until the required number of species are selected. Obviously this is an expensive algorithm since the number of tests performed scales as $O(N\, n_s^3)$, but by using a fewer number of test points, N, the computational cost can be reduced. This approach is expected to produce a more nearly optimal set of species.

2.8 Conclusions

An automated algorithm for selecting a *good* set of species for the accurate implementation of dimension reduction methods has been presented. The following conclusions can be drawn from the results presented in this paper

1. The dimension reduction errors in RCCE are highly sensitive to the choice of constrained species; errors can differ by orders of magnitude at the same dimension with two different sets of represented species as seen in Fig.2.9 and Fig.2.11.
2. Constraints based on the major species concentrations are not always the best constraints for the RCCE method as is evident from the results presented in Fig.2.14 to Fig.2.18.
3. At low values of $n_{rs} \leq 5$, the greedy algorithm is successfully able to select a near-optimal set of represented species.
4. At high values of n_{rs}, the greedy algorithm is not guaranteed to produce near-optimal sets of represented species (see Fig.2.18) but nevertheless is found to produce a good set of species for the majority of cases tested (as demonstrated in Fig.2.14 to Fig.2.17) over a wide range of testing conditions.
5. In all the cases tested, the species ordering generated by the greedy algorithm for $n_{rs} \leq 10$, produces a *non-increasing* error with every addition of represented species.

2.9 Acknowledgements

This work is supported by Department of Energy under Grant DE-FG02-90ER and the National Science Foundation under Grant CBET-0426787. SBP has a financial interest in Ithaca Combustion Enterprise, LLC., which has licensed the software ISAT-CK and CEQ used in this work.

CHAPTER 3

COMBINED DIMENSION REDUCTION AND TABULATION STRATEGY USING ISAT-RCCE-GALI FOR THE EFFICIENT IMPLEMENTATION OF COMBUSTION CHEMISTRY[†]

3.1 Abstract

Computations of turbulent combustion flows using detailed chemistry involving a large number of species and reactions are computationally prohibitive, even on a distributed computing system. Here, we present a new combined dimension reduction and tabulation methodology for the efficient implementation of combustion chemistry. In this study, the dimension reduction is performed using the *rate controlled constrained-equilibrium* (RCCE) method, and tabulation of the reduced space is performed using the *in situ* adaptive tabulation (ISAT) algorithm. The dimension reduction using RCCE is performed by specifying a set of represented (constrained) species, which in this study is selected using a new *greedy algorithm with local improvement* (GALI) (based on the greedy algorithm). This combined approach is found to be particularly fruitful in the *probability density function* (PDF) approach, wherein the chemical composition is represented by a large number of particles in the solution domain. In this work, the combined approach has been tested and compared to reduced and skeletal mechanisms using a partially-stirred reactor (PaSR) for premixed combustion of (i) methane/air (using the 31-species GRI-Mech 1.2 detailed mechanism and the 16-species ARM1 reduced mechanism) and (ii) ethylene/air (using the 111-

[†] V. Hiremath, Z. Ren, and S. B. Pope. Combined dimension reduction and tabulation strategy using ISAT-RCCE-GALI for the efficient implementation of combustion chemistry. Combustion and Flame, 158(11):2113-2127, 2011.

species USC-Mech II detailed mechanism, a 38-species skeletal mechanism and a 24-species reduced mechanism). Results are presented to quantify the relative accuracy and efficiency of three different ways of representing the chemistry: (i) ISAT alone (with a detailed mechanism); (ii) ISAT (with a reduced or skeletal mechanism); and (iii) ISAT-RCCE with represented species selected using GALI. We show for methane/air: ISAT (with ARM1 reduced mechanism) incurs 6% error, while ISAT-RCCE incurs the same error using just 8 or more represented species, and less than 1% error using 11 or more represented species, with a two-fold speedup relative to using ISAT alone with the GRI-Mech 1.2 detailed mechanism. And we show for ethylene/air: ISAT incurs 7% and 3% errors with the reduced and skeletal mechanisms, respectively, while ISAT-RCCE achieves the same levels of error 7% with just 18 and 3% with just 25 represented species, and also provides fifteen-fold speedup relative to using ISAT alone with the USC-Mech II detailed mechanism. With fewer species to track in the CFD code, this combined ISAT-RCCE-GALI reduction-tabulation algorithm provides an accurate and efficient way to represent combustion chemistry.

3.2 Introduction

Modern chemical mechanisms of real fuels involve hundreds or thousands of species and thousands of reactions [64, 100]. Incorporating such detailed chemistry in the combustion flow calculations is computationally prohibitive, and thus some form of modeling to reduce the computational cost is inevitable.

In the last two decades or so, numerous dimension reduction techniques have been developed to reduce the computational cost of combustion chemistry.

These include:

1. Skeletal Mechanisms: A skeletal mechanism consists of a selected subset of the species and reactions from the detailed mechanism, and is applicable to within certain accuracy, over the entire range of conditions (e.g., pressures, temperatures, equivalence ratios) of the detailed kinetic mechanism. Many methods have been developed to systematically generate skeletal mechanisms from detailed mechanisms, such as the Directed Relations Graph (DRG) [53], DRG with error propagation (DRGEP) [60] and Simulation Error Minimization Connectivity Method (SEM-CM) [58].

2. Reduced Chemical Mechanisms (based on QSSA): The quasi-steady-state approximation (QSSA) [11, 87] has been widely applied to develop reduced chemical mechanisms. The QSSA method involves the identification of QSS species in the system, whose net rate of production is assumed to be zero, thereby reducing the governing differential equation for the QSS species into an algebraic relation. These algebraic relations are used to eliminate the QSS species from the system. Reduced mechanisms are generally valid only over a limited range of conditions (i.e., pressures, temperatures, equivalence ratios) compared to skeletal mechanisms.

3. Dimension Reduction Methods: Another class of dimension reduction techniques is based on the observation that chemical systems involve reactions with a wide range of time scales. As a consequence, reaction trajectories are attracted to lower-dimensional attracting manifolds in the composition space. Computations can be performed in a reduced space by identifying such low-dimensional manifolds, thereby reducing the overall computational cost. Methods based on this idea include rate-

controlled constrained equilibrium (RCCE) [41, 40], computational singular perturbation (CSP) [44], intrinsic low-dimensional manifolds (ILDM) [55], trajectory-generated low-dimensional manifolds (TGLDM) [72], invariant constrained equilibrium-edge pre-image curve (ICE-PIC) [79] and one-dimensional slow invariant manifold (1D SIM) [7].

4. Storage Retrieval Methods: In these approaches, combustion chemistry computations are stored in a table, and are used to build inexpensive approximate solutions at a later stage of computation to reduce the overall cost. Methods based on this idea include the structured look-up tabulation [17], repro-modelling [93], artificial neural network (ANN) [18], *in situ* adaptive tabulation (ISAT) [66, 49] and piecewise reusable implementation of solution mapping (PRISM) [92].

In recent times, combined methodologies have also been developed, wherein reduced reaction mechanisms or dimension reduction methods are used in conjunction with storage/retrieval methods, such as ISAT-QSSA [91], ISAT-RCCE [89], and recently ICE-PIC with ISAT [77].

In reactive flow calculations, the species concentrations are governed by two processes: chemical reaction and transport. We consider the important class of solution methods in which a splitting scheme is used, where the chemical reaction and transport processes are accounted for in two separate sub-steps.

In the computational modelling of turbulent combustion using PDF methods [65], the fluid composition within the solution domain is represented by a large number of particles. In a full-scale PDF calculation, more than 2 million particles may be used, and the solution can advance for more than 5000 time steps, leading to approximately 10^{10} particle-reaction sub-steps. If such a calculation involves 100 species with the chemistry represented by a detailed mechanism (without reduction or tabulation), then at each reaction sub-step, 100 coupled ODEs need to be solved to determine the species concentrations, which can be very expensive $\mathcal{O}(0.1)$ seconds and computationally prohibitive (i.e., 30 years for 10^{10} reaction sub-steps). Instead a dimension reduction method (such as RCCE or ICE-PIC) integrated with ISAT can be used to perform the reactive flow calculations in terms of say 20 represented species; where the combined reduction-storage methodology determines and tabulates (*in situ*) the reduced space in terms of the 20 represented species based on the detailed mechanism. This combined approach can reduce the reaction sub-step time to $\mathcal{O}(10)$ μs giving more than 10^4-fold speedup compared to a direct evaluation using ODE integration (resulting in 30 hours for 10^{10} reaction sub-steps).

In our implementation of RCCE, which is integrated with ISAT [79, 89, 77, 78], a specified set of represented (constrained) species is used as constraints to perform dimension reduction. The specification of good constraints is crucial for the accuracy of dimension reduction, and recently we proposed a *greedy algorithm* [31] for selecting good represented species. In this paper we introduce a new *Greedy Algorithm with Local Improvement* (GALI) which can be used to generate an even better set of represented species by further reducing the dimension reduction error obtained by the greedy algorithm at any given dimension. In this study, we use GALI to select the represented species for performing dimension reduction with RCCE.

The combined reduction-tabulation methodologies primarily involve two errors: (i) the *tabulation* error due to the use of ISAT; and (ii) the *reduction* error due to the use of either a reduced or skeletal mechanism or due to the use of fewer represented species in the dimension-reduction methods like RCCE. The tabulation error is controlled by the specified ISAT error tolerance (ϵ_{tol}), and the reduction error is controlled by the number of species retained in the skeletal or reduced mechanism, and by the number of represented species used in the dimension-reduction method. The overall computational cost of course depends on the desired level of accuracy, and the lower the specified error tolerance or the larger the number of represented species, the more is the computational cost. There is a trade-off between accuracy and efficiency, and in this study, we consider a net reduction-tabulation error (defined more precisely in Section 3.8.3) of about 2% to be an acceptable level of accuracy. We use a low ISAT error tolerance ($\epsilon_{tol} = 10^{-5}$) and select a sufficient number of represented species with GALI to reduce the reduction-tabulation error below 2%.

The outline of the remainder of the paper is as follows: In Section 3.3 we develop a mathematical representation for a gas-phase reacting system. Next, in Section 3.4 we describe the partially-stirred reactor (PaSR) which is used for testing the combined dimension reduction and tabulation approach in this study. In the next two Sections 3.5 and 3.6 we briefly review the *in situ* adaptive tabulation (ISAT) algorithm and the rate controlled constrained-equilibrium (RCCE) methods. A reader familiar with these concepts may skip these two sections. Following this in Section 3.7, we present a brief overview of the greedy algorithm and the details of the new GALI. Next, in Section 3.8 we describe the combined dimension reduction and tabulation methodology. Finally, in Sections 3.9 and 3.10 we present the results and draw conclusions.

3.3 Representation of Chemistry

We consider a reacting gas-phase mixture consisting of n_s chemical species, composed of n_e elements. The set of all species is denoted by $\mathbf{\Phi}$. The thermochemical state of the mixture (at a given position and time) is completely characterized by the pressure p, the mixture enthalpy h, and the n_s-vector $\mathbf{z}$ of specific moles of the species. (The specific moles of species i is given as $z_i = Y_i/w_i = X_i/w_{mix}$, where Y_i, X_i and w_i are the mass fraction, mole fraction and molecular weight of the species i, respectively, and $w_{mix} \equiv \sum_{i=1}^{n_s} X_i w_i$ is the mixture molecular weight.) To simplify the exposition, we consider an adiabatic and isobaric system with a fixed specified pressure, p, and so the thermochemical state is fully characterized by $\mathbf{z}, h$.

It is useful to consider the species composition $\mathbf{z}$ to be an n_s-vector or a point in the n_s-dimensional *full composition space*. With $\mathbf{w}$ denoting the n_s-vector of molecular weights of all the species, then, for realizability, $\mathbf{z}$ must satisfy the normalization condition, $\mathbf{w}^T\mathbf{z} = 1$. (This corresponds to the species mass fractions summing to unity.)

3.3.1 Reaction Trajectories

Due to chemical reactions, the chemical composition $\mathbf{z}$ evolves in time according to the following set of ordinary differential equations (ODEs)

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{S}(\mathbf{z}(t)), \tag{3.1}$$

where S is the n_s-vector of chemical production rates determined by the chemical mechanism used to represent the chemistry.

The *reaction mapping*, $\mathbf{R}(\mathbf{z}, t)$ is defined to be the solution to Eqn.3.1 after time t starting from the initial composition $\mathbf{z}$. The reaction mapping obtained by directly integrating the set of ODEs given by Eqn.3.1 is referred to as a *direct evaluation* (DE). In this study, we use DDASAC [15] for direct evaluation.

3.4 Partially-Stirred Reactor (PaSR)

We are interested in studying turbulent combustion flow problems using the LES/PDF approach. Computations of turbulent flames using the LES/PDF approach are computationally expensive. To demonstrate the working and efficiency of our combined dimension reduction and tabulation approach, we instead consider the computationally cheaper partially-stirred reactor (PaSR), which is a representative test case for the combustion problems of our interest. A PaSR is similar to a particle PDF method applied to a statistically homogeneous flow. We can vary pressure, inflowing stream temperatures and the reaction time step Δt in the PaSR to be representative of conditions in an LES/PDF calculation.

In this study, we consider the test case of a PaSR involving premixed combustion of two different fuel/air mixtures: (i) methane/air and (ii) ethylene/air.

Unless specified otherwise, the default configuration studied in the PaSR involves two inflowing streams: a stoichiometric premixed stream of given fuel/air mixture at 600 K; and a pilot stream consisting of the adiabatic equilibrium products of the stoichiometric fuel/air mixture (corresponding to unburnt gas temperature of 600 K). The mass flow rates of these streams are in the ratio 0.95:0.05. Initially (t = 0), all particle compositions are set to be the pilot-stream

Table 3.1: The default values of various PaSR and ISAT parameters used for the methane/air and ethylene/air premixed combustion test cases.

Parameter	methane/air	ethylene/air
ϕ	1	1
T	600 K	600 K
p	1 atm	1 atm
τ_{res}	10 ms	100 μs
τ_{mix}	1 ms	10 μs
τ_{pair}	1 ms	10 μs
Δt	0.033 ms	0.33 μs
N_p	100	100
ϵ_{tol}	10^{-5}	10^{-5}
S	1 GB	1 GB

composition. The pressure is atmospheric throughout.

Other important parameters involved in the PaSR for the methane/air and ethylene/air premixed combustion cases are listed in the Table 3.1. The parameters include the equivalence ratio, ϕ; (unburnt) temperature of the inflowing streams, T; pressure, p; residence time, τ_{res}; mixing time scale, τ_{mix}; pairing time scale, τ_{pair}; reaction time step, Δt; number of particles, N_p; ISAT error tolerance, ϵ_{tol}; and ISAT table size, S.

The residence time for methane/air and ethylene/air premixed combustion in PaSR is chosen small enough (relative to the chemical time scale) to introduce a good range of non-equilibrium temperature and species compositions in

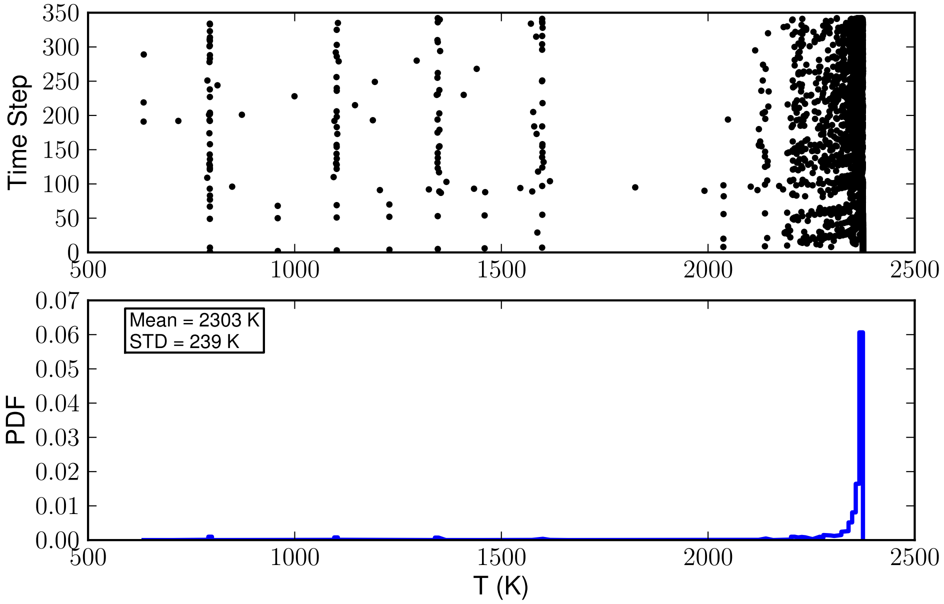


Figure 3.1: Scatter plot of temperature (top) retrieved from 10 selected particles over 350 time steps and PDF of temperature (bottom) in the statistically stationary state (time steps 100 and above) inside PaSR involving methane/air premixed combustion with residence time τ_{res} = 10 ms. The equilibrium temperature is 2374 K.

the PaSR. As the residence time is lowered and approaches the blow-out limit, the PaSR calculations become more computationally intensive. We have tried to maintain a good balance between the computational cost and the range of chemical compositions in the PaSR to perform tests. The temperature distributions inside PaSR for the methane/air and ethylene/air premixed combustion is shown in Fig.3.1 and Fig.3.2, respectively.

The chemical mechanisms used to represent the chemistry for the methane and ethylene fuels are listed in Table 3.2 along with the details about the number of elements, species and reactions involved in these mechanisms.

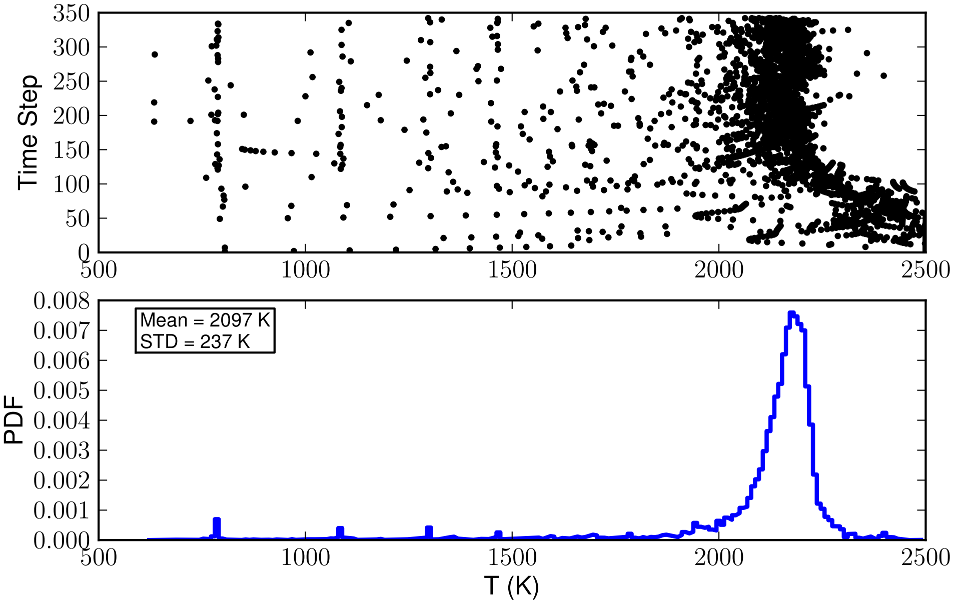


Figure 3.2: Scatter plot of temperature (top) retrieved from 10 selected particles over 350 time steps and PDF of temperature (bottom) in the statistically stationary state (time steps 100 and above) inside PaSR involving ethylene/air premixed combustion with residence time τ_{res} = 100 μs. The equilibrium temperature is 2500 K.

In our implementation of PaSR, computations can be performed using the full set of species, $\mathbf{\Phi}$, in the full composition space (without any dimension reduction) or using a smaller set of represented species, $\mathbf{\Phi}^r$, using RCCE. For a given test case, PaSR calculations are performed with and without dimension reduction, and the compositions obtained with the two approaches are compared to estimate errors involved in dimension reduction.

In the next two sections we review the concepts of ISAT and RCCE algorithms. A reader familiar with these concepts may skip the next two sections.

Table 3.2: The details of the chemical mechanisms used for methane and ethylene. (The mechanisms with 4 elements do not include Ar.)

Fuel	Mechanism	Name	Elements	Species	Reactions
Methane	Detailed	GRI-Mech 1.2 [24]	4	31	175
	Reduced	ARM1 [88]	4	16	12-step
Ethylene	Detailed	USC-Mech II (Optimized in 2009) [99, 84]	5	111	784
	Skeletal	Skeletal [23] (with Ar added)	5	38	212
	Reduced	Reduced (personal communication, based on [103])	4	24	20-step

3.5 *In Situ* Adaptive Tabulation (ISAT)

3.5.1 Introduction

ISAT is a tabulation algorithm which returns accurate approximations (within a specified error tolerance) for computationally expensive multi-dimensional functions using (previously evaluated) tabulated function values. The ISAT algorithm has been applied in various fields [81, 28], but in particular it has proved to be extremely fruitful in reducing the computational cost of evaluating the reaction mapping in combustion flow problems [49].

3.5.2 Overview of the ISAT Algorithm

In this section we briefly describe the key terms involved in the ISAT algorithm which are referred to repeatedly in the following sections.

Consider the application of the ISAT algorithm for the evaluation of reaction mappings in an adiabatic isobaric reactive system with a specified constant pressure, p, such that the thermochemical state is fully characterized by $\{\mathbf{z}, h\}$. In such a system, given the initial composition $\mathbf{z}(0)$, enthalpy h, and the reaction time step Δt, ISAT aims to return the reaction mapping $\mathbf{z}(\Delta t)$ within a specified error tolerance denoted by ϵ_{tol}. The initial input composition vector given to ISAT is generally referred to as a *query*, and is denoted by x. The query vector x, for example, in this case could be of the form

$$\mathbf{x} = \{\mathbf{z}(0), h_s(0), \Delta t\}, \tag{3.2}$$

where $h_s \equiv h_s(h, \mathbf{z})$ is the sensible enthalpy which is defined as

$$h_s = h - \mathbf{h}^{f^T}\mathbf{z}, \tag{3.3}$$

where $\mathbf{h}^f$ denotes the n_s-vector of molar enthalpies of formation of the species.

For this query vector $\mathbf{x}$, ISAT returns a mapping, which is denoted by $\mathbf{f}$, which is given by

$$\mathbf{f} = \{\mathbf{z}(\Delta t), h_s(\Delta t), T(\Delta t)\}, \tag{3.4}$$

where $\mathbf{z}(\Delta t)$, $h_s(\Delta t)$ and $T(\Delta t)$ denote the composition vector, the sensible enthalpy and the temperature, respectively, after time step Δt.

Given a query vector $\mathbf{x}$ to ISAT, a series of steps are followed in the ISAT algorithm to determine the mapping $\mathbf{f}$, which are described in detail in [66, 49]. Here we list only the important definitions, and key events involved:

1. A stored entry in the ISAT table is referred to as a *leaf*.
2. The region around a leaf's value of $\mathbf{x}$ in which a linear approximation to $\mathbf{f}(\mathbf{x})$ can be returned within the specified error tolerance ϵ_{tol} is called the *region of accuracy* of that leaf. These regions are approximated by hyper-ellipsoids which are referred to as *ellipsoids of accuracy* (EOA).
3. Given a query $\mathbf{x}$, a *search* is performed inside the ISAT table to find an EOA which covers the query point $\mathbf{x}$.
4. If such an EOA is found, then a linear approximation to the mapping $\mathbf{f}(\mathbf{x})$ is returned and this event is referred to as a successful *retrieve*.
5. If the retrieve attempt is unsuccessful, then a direct evaluation of the mapping $\mathbf{f}(\mathbf{x})$ is performed by integrating Eqn.3.1.

6. After the direct evaluation, a certain number of leaves close to the query point $\mathbf{x}$ are selected for grow attempts. If the exact mapping obtained by direct evaluation is found to be within the specified error tolerance (ϵ_{tol}) of the linear approximation obtained from one of these leaves, then the EOA of that leaf is grown to include the query point $\mathbf{x}$ and this event is called a *grow*.
7. If all the grow attempts fail, then a new leaf is added to the ISAT table (if not full) and this event is referred to as an *add*.
8. If the table size is full, then $\mathbf{f}(\mathbf{x})$ is returned without performing an add, and this event is referred to as a *direct evaluation* (DE).

In the present study, ISAT is used to tabulate the reaction mapping, $\mathbf{R}(\mathbf{z}, t)$, in the full composition space (without dimension reduction), and also to tabulate the reaction mapping in the reduced space (defined in Section 3.8.2) when used in conjunction with dimension reduction. In the next section we describe the RCCE dimension reduction method and its implementation.

3.6 Dimension Reduction

In this section we briefly describe the notation used in the RCCE dimension reduction method; detailed descriptions are provided in [79, 77, 31].

In RCCE, the set of species $\mathbf{\Phi}$ is decomposed as $\mathbf{\Phi} = \{\mathbf{\Phi}^r, \mathbf{\Phi}^u\}$, where $\mathbf{\Phi}^r$ is the set of *represented* species with cardinality n_{rs}, and $\mathbf{\Phi}^u$ is the set of *unrepresented* species with cardinality n_{us}, where $n_{rs} + n_{us} = n_s$ and $n_{rs} < n_s - n_e$.

The *reduced representation* of the species composition is denoted by $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$, where $\mathbf{z}^r$ is an n_{rs}-vector of specific moles of represented species, $\mathbf{\Phi}^r$; and $\mathbf{z}^{u,e}$ is an n_e-vector of specific moles of the elements in the unrepresented species, $\mathbf{\Phi}^u$ (for atom conservation). Thus, $\mathbf{r}$ is a vector of length $n_r = n_{rs} + n_e$ in the *reduced composition space*, and the dimension of the system is reduced from n_s to $n_r < n_s$. At any time t, the reduced representation, $\mathbf{r}(t)$, is related to the full composition, $\mathbf{z}(t)$, by

$$\mathbf{r}(t) = \mathbf{B}^T \mathbf{z}(t), \tag{3.5}$$

where $\mathbf{B}$ is constant $n_s \times n_r$ matrix which can be determined for a specified set of represented species.

3.6.1 Steps Involved in Dimension Reduction

In this section we briefly describe the four main steps involved in our implementation of RCCE. Since our implementation of RCCE is integrated with ISAT, some of the steps in our implementation of RCCE differ from those of other works found in the literature: those steps are highlighted and justified.

1. The first important step in the application of the RCCE dimension reduction method is the *selection* of the set of represented (constrained) species, $\mathbf{\Phi}^r$. For a given set of represented species, $\mathbf{\Phi}^r$, the reduced representation is given as $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$.

   Alternatively, in many of the RCCE implementations [35, 36] general linear constraints on species are specified. In our implementation of RCCE, to simplify the user interface and specification of constraints, we use the species specific moles of the represented species as the constraints.

2. The next step is the *species reconstruction*, i.e., given a reduced representation $\mathbf{r}(0)$ at time $t = 0$, reconstruct an estimate of the full composition denoted by $\mathbf{z}^{DR}(0)$.

   In RCCE the *species reconstruction* is performed by computing the constrained-equilibrium composition for the given constraints. In our implementation of RCCE, the constrained-equilibrium composition is computed using the CEQ [69] code, with the constraints given by the reduced representation $\mathbf{r}$. The constrained-equilibrium composition at $\mathbf{r}$ is denoted by $\mathbf{z}^{CE}(\mathbf{r})$. So the reconstructed composition in RCCE is given as

$$\mathbf{z}^{DR}(0) = \mathbf{z}^{CE}(\mathbf{r}(0)). \tag{3.6}$$

3. The next step is to obtain the *reaction mapping*. Starting from the reconstructed composition, $\mathbf{z}^{DR}(0)$, the set of ODEs (3.1) is integrated numerically in the full space using DDASAC [15] to obtain the reaction mapping after time t, denoted by $\mathbf{z}^{DR}(t)$, as shown in Fig.1 in [31].

   A different approach for the RCCE method as suggested in [40] and also used in [35, 36] is to integrate the rate equations for the constraint potentials which is more economical than integrating the ODEs (3.1) directly. In our implementation of RCCE, we chose the former approach for the ease of combining RCCE dimension reduction method with ISAT, which is discussed in more detail in [89]. It should be noted that the reaction mappings obtained using these two approaches are not the same, and they actually provide two different approximations to the exact reaction mapping.

4. The final step involved in the dimension reduction method is *reduction*, i.e., from the obtained *reaction mapping* after time t, $\mathbf{z}^{DR}(t)$, compute the

reduced representation denoted by $\mathbf{r}^{DR}(t)$ (shown in Fig.1 in [31]), given as:

$$\mathbf{r}^{DR}(t) = \mathbf{B}^T \mathbf{z}^{DR}(t). \tag{3.7}$$

To summarize, the key steps involved in the RCCE dimension reduction method are

1. **Selection**: Identifying good constraints or the set of represented species, $\mathbf{\Phi}^r$ for dimension reduction.
2. **Species Reconstruction**: Given the constraints, $\mathbf{r}(0)$, reconstructing the full composition, $\mathbf{z}^{DR}(0)$.
3. **Reaction Mapping**: Starting from the *reconstructed composition* $\mathbf{z}^{DR}(0)$, computing the *reaction mapping* after time t in the full composition space $\mathbf{z}^{DR}(t)$.
4. **Reduction**: From the *reaction mapping* $\mathbf{z}^{DR}(t)$, obtaining the *reduced representation* $\mathbf{r}^{DR}(t)$ after time t.

3.6.2 Dimension Reduction Errors

In this section we define the various errors involved in the dimension reduction process and describe the method employed to measure these errors using the PaSR.

Given a composition, $\mathbf{z}(0)$ in the full composition space, the *reaction mapping*, $\mathbf{R}(\mathbf{z}(0), t)$ (for $t \geq 0$) is more concisely denoted by $\mathbf{z}(t)$ (see Fig.4 in [31]).

For a given set of *represented* species, $\mathbf{\Phi}^r$, the *reduced* representation of the full composition, $\mathbf{z}(0)$ is denoted by $\mathbf{r}(0)$ and is obtained by performing the *reduction* using (3.5) as

$$\mathbf{r}(0) = \mathbf{B}^T \mathbf{z}(0). \tag{3.8}$$

At $\mathbf{r}(0)$, the reconstructed composition using a dimension reduction method is denoted by $\mathbf{z}^{DR}(0)$. Starting from the reconstructed composition, the *reaction mapping*, $\mathbf{R}(\mathbf{z}^{DR}(0), t)$ in the full composition space is more concisely denoted by $\mathbf{z}^{DR}(t)$ (see Fig.4 in [31]).

Now for a representative test problem, to estimate the errors incurred using a dimension reduction method, a number of test compositions are selected in the full space. Let the number of test composition used be denoted by N. To generate those test compositions, we perform a PaSR computation in the *full composition space* (without dimension reduction) and then pick N distinct test compositions (corresponding to *adds* in the ISAT table) in the full space and denote them by $\mathbf{z}^{(n)}(0)$, for $n = 1$ to N; and their corresponding *reaction mappings* after a fixed constant time t are denoted by $\mathbf{z}^{(n)}(t)$.

At the N chosen compositions, $\mathbf{z}^{(n)}(0)$, in the full space, and for a given set of *represented* species, $\mathbf{\Phi}^r$, we denote the corresponding reduced representations by $\mathbf{r}^{(n)}(0)$, the reconstructed compositions by $\mathbf{z}^{DR(n)}(0)$, and the reaction mappings by $\mathbf{z}^{DR(n)}(t)$.

Note that, given $\mathbf{z}(0)$ and t, $\mathbf{z}^{DR}(t)$ depends on the specification of the represented species, $\mathbf{\Phi}^r$. As needed, we show this dependence explicitly by the notation $\mathbf{z}^{DR}(t, \mathbf{\Phi}^r)$.

At this stage, we define the error in the *reaction mapping* obtained after time

t starting from the reconstructed composition to be

$$\epsilon(t, \mathbf{\Phi}^r) = \frac{[\mathbf{z}^{DR(n)}(t, \mathbf{\Phi}^r) - \mathbf{z}^{(n)}(t)]_{rms}}{[\mathbf{z}^{(n)}(t)]_{rms}}, \tag{3.9}$$

where the operator $[\,]_{rms}$ is defined by, for example,

$$[\mathbf{z}^{(n)}(t)]_{rms} = \sqrt{\tfrac{1}{N}\sum_{n=1}^{N} ||\mathbf{z}^{(n)}(t)||^2}, \tag{3.10}$$

where $||\mathbf{z}||$ denotes the 2-norm.

In particular we have two important errors in the dimension reduction method corresponding to $t = 0$ and $t = \Delta t$:

1. **Species Reconstruction Error**: This is the error in reconstructing the full composition given a reduced composition $\mathbf{r}(0)$ at $t = 0$ and is given by Eqn.3.9 as $\epsilon(0, \mathbf{\Phi}^r)$.
2. **Reaction Mapping Error**: This is the error in the reaction mapping obtained after time step Δt (reaction time step) starting from the reconstructed composition and is equal to $\epsilon(\Delta t, \mathbf{\Phi}^r)$.

3.7 Selection of Represented Species

In the RCCE dimension reduction method, the selection of represented species for dimension reduction is an important task. In many of the RCCE applications, the constraints are found by a careful examination of the chemical reactions involved in the system [35, 36, 27]. Ideally one would like to find the *optimal* set of represented species that produces the minimum dimension reduction error, but finding such an optimal set of species is very difficult. Recently we proposed a new automated *greedy algorithm* [31] for selecting represented species for the RCCE dimension reduction method, which showed promising results. Based on this greedy algorithm, we have now developed a new *Greedy Algorithm with Local Improvement* (GALI) which performs even better than the original greedy algorithm proposed in [31]. In the following sections, we briefly review the greedy algorithm and then present details about the GALI.

3.7.1 Greedy Algorithm

A greedy algorithm proceeds in stages, making the best (locally optimum) choice at each stage while finding a solution [20]. Greedy algorithms are not guaranteed to give the optimal solutions, but provide good solutions for many mathematical problems.

In our greedy algorithm [31], we split the task of selecting n_{rs} number of represented species into n_{rs} different stages, selecting the best available species (one which minimizes the error) at each stage. In [31], results are reported of RCCE dimension reduction tests using the greedy algorithm for PaSR computations involving methane/air premixed combustion at many equivalence ratios, pressures and temperatures. The main conclusion drawn from the results presented in [31] is that the greedy algorithm works well at small values of n_{rs} (say ≤ 5), but at higher values of n_{rs}, the greedy algorithm is found to generate a poor set of represented species in some cases.

As an attempt to further reduce the error obtained with the greedy algorithm at high values of n_{rs}, we have introduced an additional *local improvement* step to the greedy algorithm which is described in the next section.

3.7.2 Greedy Algorithm with Local Improvement (GALI)

We present a new method for further improving the dimension reduction error of the set of represented species obtained by the greedy algorithm. This is achieved by performing a *local improvement* over the set of species selected by the greedy algorithm at any given dimension.

In the greedy algorithm, at each stage, one new species (producing the minimum error) is added to the set of species selected in the previous stages. The greedy algorithm does not allow for a species selected in the previous stages to be “swapped” with another species producing lower error. For this reason, a new local improvement step has been introduced in the greedy algorithm to enable replacement of previously selected species in the greedy algorithm by species which result in lower error. The idea behind the local improvement step is to sequentially swap a species from the set of represented species obtained from the greedy algorithm by a species from the set of unrepresented species (one at a time) and check for improvement in error. (Only single species swaps are considered to reduce the overall computations involved in the local improvement step.) In the local improvement step, all possible species swaps between the represented and the unrepresented set of species are performed to check for improvement in error. Any species swap that results in improvement in error is saved, and at the end of the local improvement step a new improved set of species (with an error less than or equal to the error given by the greedy algorithm) is generated.

The complete algorithm of first applying the greedy algorithm and later improving the set of represented species by performing local improvement is called the *Greedy Algorithm with Local Improvement* (GALI).

The greedy algorithm is described in detail in [31]. Here we present the algorithm for the local improvement step. In the greedy algorithm a fixed number of test points, $N = N_G$, are used to compute the error. The local improvement step described here can be applied using a fewer number of test points, $N = N_L$, such that $N_L < N_G$, to reduce the computational cost. To explicitly specify the number of test points used for computing the error, here we use the notation $\epsilon\left(N, t, \mathbf{\Phi}^r\right)$ to denote the error (Eqn.3.9) computed using N test points. For a given definition of error – species reconstruction with $t = 0$ or reaction mapping error with $t = \Delta t$ – we denote the error more concisely as $\epsilon\left(N, \mathbf{\Phi}^r\right)$.

Following the notation used in [31], the local improvement step at a given value of n_{rs} can be performed as follows:

1. Apply the greedy algorithm based on the given definition of error, $\epsilon\left(N_G, \mathbf{\Phi}^r\right)$, and obtain the set of n_{rs} represented species denoted by $\mathbf{\Phi}^g$.
2. Let the set of represented species at any stage of the local improvement algorithm be denoted by $\mathbf{\Phi}^r$. Initially, $\mathbf{\Phi}^r = \mathbf{\Phi}^g$. At any stage the set of undetermined unrepresented species is denoted by $\mathbf{\Phi}^{uu}\left(\mathbf{\Phi}^r\right)$.
3. Let the set of species obtained at the end of the local improvement step be denoted by $\mathbf{\Phi}^{gi}$. Initially, we set $\mathbf{\Phi}^{gi} = \mathbf{\Phi}^g$ (assuming no improvement).
4. Let $\mathbf{\Phi}^r_{jk}$ denote the set of species obtained by swapping the j^{th} species from $\mathbf{\Phi}^r$ with the k^{th} species from $\mathbf{\Phi}^{uu}\left(\mathbf{\Phi}^r\right)$.
5. We use N_L test points inside the local improvement step to compute error. The local improvement step involves the following loop:

```
for $j$ from $1$ to cardinality of $\mathbf{\Phi}^r$
    for $k$ from $1$ to cardinality of $\mathbf{\Phi}^{uu}\left(\mathbf{\Phi}^r\right)$
```

```
        ∗ form the set $\mathbf{\Phi}^r_{jk}$ and evaluate $\epsilon\left(N_L, \mathbf{\Phi}^r_{jk}\right)$
        ∗ if $\epsilon\left(N_L, \mathbf{\Phi}^r_{jk}\right) < \epsilon\left(N_L, \mathbf{\Phi}^r\right)$ then reset $\mathbf{\Phi}^r = \mathbf{\Phi}^r_{jk}$
    end for
end for
```

6. If $\mathbf{\Phi}^r \neq \mathbf{\Phi}^{gi}$, then save the set of species, $\mathbf{\Phi}^{gi} = \mathbf{\Phi}^r$, and rerun the previous local improvement loop (5), else continue.
7. Check if $\mathbf{\Phi}^{gi}$ generates lower error than $\mathbf{\Phi}^g$ over the N_G test points used in the greedy algorithm. If $\epsilon\left(N_G, \mathbf{\Phi}^{gi}\right) < \epsilon\left(N_G, \mathbf{\Phi}^g\right)$, then save $\mathbf{\Phi}^{gi}$ else reset $\mathbf{\Phi}^{gi} = \mathbf{\Phi}^g$.

In the above algorithm, the local improvement loop (steps 5 and 6) is guaranteed to terminate because of the strictly non-increasing error check being employed. In all the cases tested, the local improvement loop has never been executed for more than three times. If needed, an upper bound can also be set on the maximum number of times the local improvement loop can be executed.

At the end of the local improvement step, we obtain an improved set of species with an error less than or equal to the error obtained by the greedy algorithm.

3.7.3 Computational Cost

The computational cost of the greedy algorithm and the local improvement step directly depends on the number of computations involved in the evaluation of errors. For a fixed value of n_{rs}, the number of computations involved are

- greedy algorithm: $\mathcal{O}(n_{rs} n_s N_G)$; and
- local improvement: $\mathcal{O}\left(n_{rs}(n_s - n_{rs}) N_L M\right)$,

where M is the average number of local improvement loops executed.

More quantitative results for the computational cost of the greedy algorithm and local improvement are presented in the next section.

3.7.4 Comparison of Results

In this section we present results comparing dimension reduction error obtained by the greedy algorithm and the improvements achieved by the local improvement step.

We consider the PaSR with methane/air premixed combustion and study two cases for which the greedy algorithm does not yield very good results (as presented in [31]):

1. $\phi = 1.2$, $T = 1200$ K and $p = 1$ atm;
2. $\phi = 0.8$, $T = 1200$ K and $p = 1$ atm,

where ϕ is the equivalence ratio, T is the temperature of the inflowing premixed stream and p is the pressure inside the PaSR.

For the above two cases we first apply the greedy algorithm for n_{rs} = 1 to 15 and obtain the represented species ordering. Next, we apply the local improvement step (as described in the previous section) over the species set obtained from the greedy algorithm for n_{rs} = 3 to 15. We skip n_{rs} = 1 which is already the

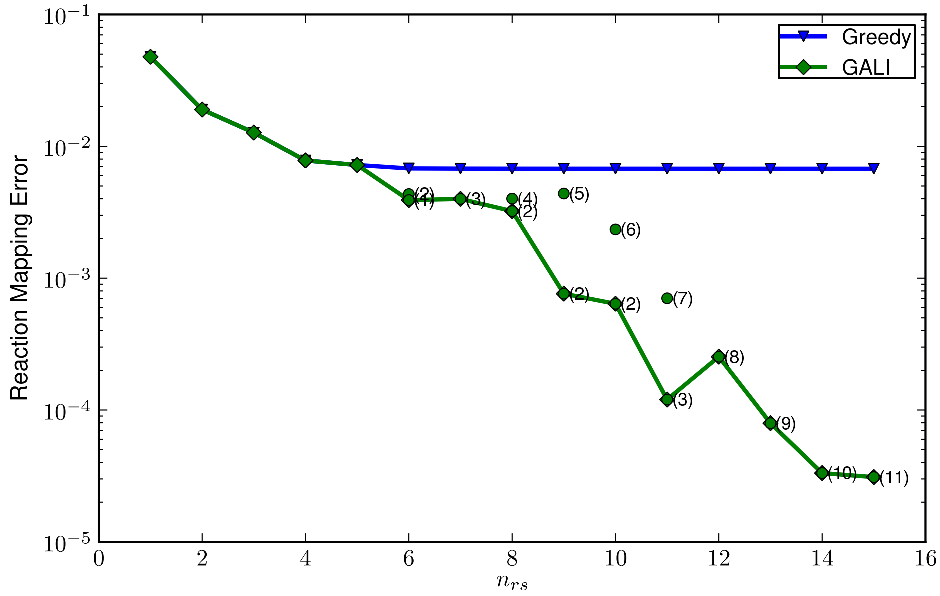


Figure 3.3: Reaction mapping error for methane/air premixed combustion at $\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$ as a function of number of represented species, n_{rs}, obtained with (i) Greedy algorithm (using $N_G = 2500$ test points) and (ii) GALI (using $N_L = 200$ test points for local improvement). At each value of n_{rs}, the error achieved after each local improvement loop in the GALI algorithm is marked with a solid circle and the number of successful swaps (resulting in reduction in error) performed in that loop are indicated in parenthesis. The test points are selected from a PaSR run involving methane/air premixed combustion at $\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$ with chemistry represented by the 31-species GRI-Mech 1.2 detailed mechanism.

optimal, and $n_{rs} = 2$ where the local improvement is not expected to give any further improvement.

The dimension reduction error for case (1) $\phi = 1.2$, $T = 1200$ K and $p = 1$ atm is shown in Fig.3.3 and for case (2) $\phi = 0.8$, $T = 1200$ K and $p = 1$ atm is shown in Fig.3.4. In these figures, we see that the local improvement step reduces the error at many values of n_{rs}. In case (2), the greedy algorithm already performs

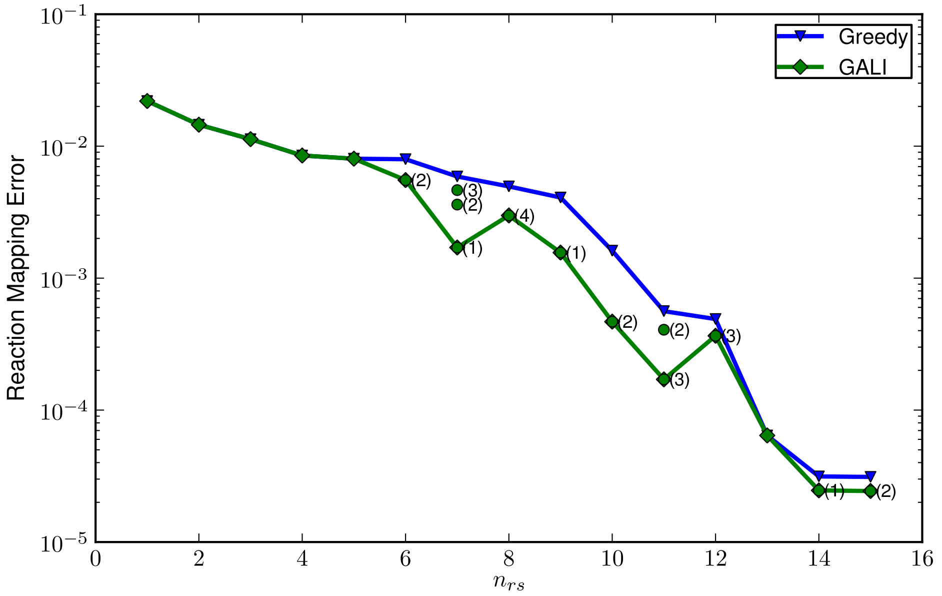


Figure 3.4: Reaction mapping error for methane/air premixed combustion at $\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$ as a function of number of represented species, n_{rs}, obtained with (i) Greedy algorithm (using $N_G = 2500$ test points) and (ii) GALI (using $N_L = 200$ test points for local improvement). At each value of n_{rs}, the error achieved after each local improvement loop in the GALI algorithm is marked with a solid circle and the number of successful swaps (resulting in reduction in error) performed in that loop are indicated in parenthesis. The test points are selected from a PaSR run involving methane/air premixed combustion at $\phi = 0.8$, $T = 1200\,K$ and $p = 1\,atm$ with chemistry represented by the 31-species GRI-Mech 1.2 detailed mechanism.

quiet well (error is less than 10^{-4} for $n_{rs} > 12$), so in Fig.3.4 we do not see significant improvement with the local improvement step, however in case (1), the greedy algorithm performs poorly (error remains constant at about 10^{-2} for $n_{rs} > 4$), and so in Fig.3.3 we see more than an order of magnitude reduction in error for $n_{rs} > 10$. In the figures, at each value of n_{rs}, the reduction in error achieved at the end of each local improvement loop is marked. We see that at any n_{rs} value, no more than three loops of local improvement are executed.

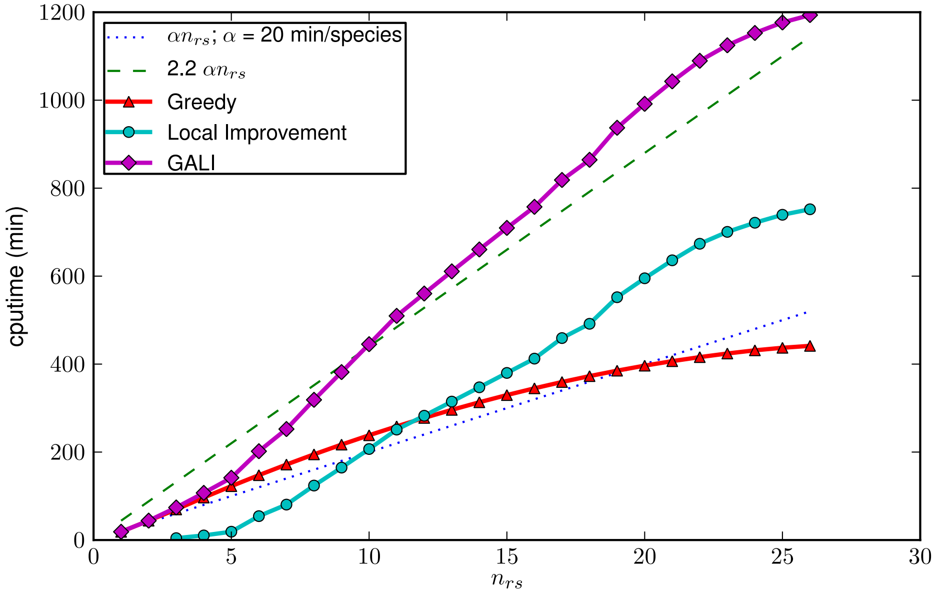


Figure 3.5: The CPU time spent in selecting represented species based on the reaction mapping error (i) using the greedy algorithm (with $N_G = 2500$ test points); (ii) performing local improvement (with $N_L = 200$ test points); and (iii) using complete GALI at different values of n_{rs}. The test points are selected from a PaSR run involving methane/air premixed combustion at $\phi = 1.2$, $T = 1200\,K$ and $p = 1\,atm$ with chemistry represented by the 31-species GRI-Mech 1.2 detailed mechanism. The CPU time is measured by running a serial implementation of the greedy algorithm on a 2.2 GHz Quad-Core AMD Opteron(tm) Processor.

The total CPU time taken for the greedy, local improvement and the overall GALI for case (1) is shown in Fig.3.5. (The CPU times for case (2) are similar and hence are not shown.) We see that the greedy algorithm takes approximately 20 minutes per species selection. The local improvement takes approximately the same order of time as the greedy algorithm at all the values of n_{rs}. The combined cost of the GALI is therefore approximately twice the cost (40 minutes per species selection) of the greedy algorithm, but the cost is still linear in the number of represented species n_{rs}.

Note that here the cost of GALI has been computed for all the values of n_{rs}, and the CPU time taken at $n_{rs} = 26$ is about 20 hours. However,

- for sufficient accuracy, $n_{rs} = 15$ is adequate, and the CPU time for GALI at $n_{rs} = 15$ is around 10 hours;
- GALI needs to be applied only once (for one set of conditions and a specified value of n_{rs}) and the same set of species is likely to be good in other conditions [31]. (In [31] we have already shown that species selected at one condition work well at other conditions, and with the addition of the local improvement step, the combined GALI may work well over a wider range of conditions.); and
- the cost of GALI is small compared to the savings achieved when applied to LES/PDF which require 10^4-10^5 CPU hours.

In short, the local improvement step can be used to reduce errors in cases where the greedy algorithm alone may not yield very good results.

3.8 Combined Dimension Reduction and Tabulation

In the previous two sections we described the tabulation algorithm using ISAT and dimension reduction using RCCE. Both the methodologies are very good in reducing the computational cost of combustion chemistry in their own perspectives, but when combined they can reduce the cost even further.

3.8.1 Introduction

The ISAT algorithm has been successfully applied in combustion chemistry problems involving up to $n_s \leq 50$ species. For problems in higher dimensions, i.e. problems involving more than 50 species, the direct use of ISAT may not be very efficient, due to the large table size and search times. Similarly, dimension reduction can be used to represent the chemistry using a reduced number of species, thereby reducing the cost of tracking species in a CFD code. But, if a tabulation algorithm is not used in conjunction, the dimension reduction alone cannot be very efficient due to the expensive species reconstruction and reaction mapping involved in the algorithm (as described in Section 3.6) for computing the reduced mapping. Our task is to integrate these two successful methodologies - tabulation and dimension reduction - to extract the maximum out of both and make the combined algorithm accurate and efficient for combustion chemistry problems.

In this combined strategy of dimension reduction and tabulation, we first apply the dimension reduction method using RCCE by specifying a set of represented species (selected using the GALI algorithm) and then we tabulate the reduced space using ISAT. This combined methodology can be applied to chemical systems involving a large number of species (100 to 1000) by first applying the dimension reduction to reduce the dimensionality of the system to say around 20 (depending on the level of accuracy needed) and then using the ISAT to tabulate the reaction mapping in the reduced space. (For very large mechanisms involving more than 1000 species, it may be more appropriate to use the ISAT-RCCE-GALI approach on a smaller skeletal mechanism, which of course will be less accurate but may be computationally more feasible.) Since the tabulation is done over the reduced space in a lower dimension, retrieving from the ISAT table is very efficient. However, since the reaction mapping is computed in the full space, constructing the ISAT table may be relatively costly. Nevertheless, for problems such as LES/PDF computation requiring very many queries, the overall efficiency may be significantly improved.

3.8.2 Combined Reduction-Tabulation Algorithm

A simplified schematic of the combined dimension-reduction and tabulation algorithm is shown in Fig.3.6. The first step (pre-processing) in this combined reduction-tabulation approach is the specification of the represented species for dimension reduction. Given the number of constrained species n_{rs}, or a dimension-reduction error tolerance $\epsilon_{DR,tol}$, we apply GALI (using a representative PaSR configuration) to obtain the specified number of represented species or enough number of represented species such that the dimension-reduction error (Eqn.3.9) is less than the specified error tolerance. Given the represented species, the reduced representation is defined as $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$.

In an isobaric adiabatic system with a specified fixed pressure, p, the chemistry is represented using this combined approach by the reduced set of variables, $\{\mathbf{r}, h\}$. So, the only variables that need to be stored and carried (for example by the particles in a PDF simulation) are the reduced variables, $\mathbf{r}$ and h. Given the initial reduced composition $\mathbf{r}(0)$, the mapping after time step Δt is given (as described in Section 3.6.1 and shown in Fig.3.6) by first performing species reconstruction to obtain $\mathbf{z}^{DR}(0)$, followed by computing the reaction mapping $\mathbf{z}^{DR}(\Delta t)$ and then performing the reduction to retrieve

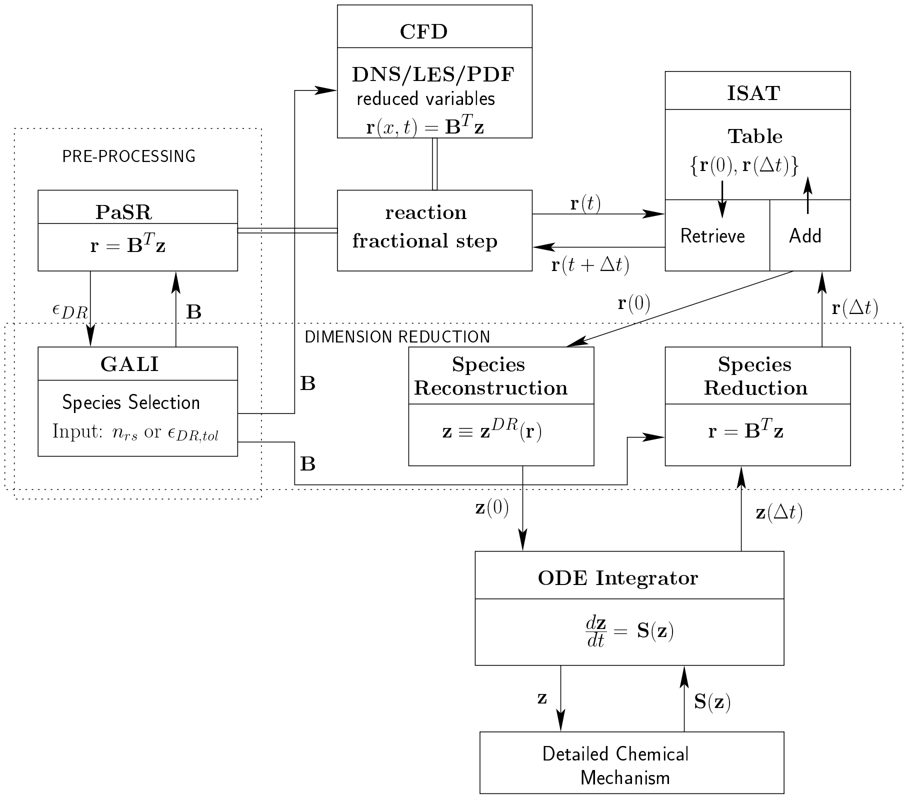


Figure 3.6: Schematic of the combined dimension-reduction and tabulation algorithm. The pre-processing step involves the selection of represented species using GALI with a representative PaSR test problem for a specified number of represented species, n_{rs}, or a specified dimension-reduction error tolerance, $\epsilon_{DR,tol}$. The choice of represented species is encapsulated in the specification of the $n_s \times n_r$ matrix $\mathbf{B}$, which relates the reduced composition $\mathbf{r}$ to the full composition $\mathbf{z}$ by $\mathbf{r} = \mathbf{B}^T\mathbf{z}$. At each reaction fractional step, ISAT is invoked to retrieve the reduced mapping $\mathbf{r}(\Delta t)$. For an unsuccessful retrieve, the reduced mapping is obtained using RCCE by performing species reconstruction (using CEQ [69]), followed by computing the reaction mapping with the detailed mechanism (using DDASAC [15]), followed by reduction to obtain $\mathbf{r}(\Delta t)$.

$\mathbf{r}(\Delta t) \equiv \mathbf{r}^{DR}(\Delta t)$. Computing the exact mapping with dimension reduction is even more expensive than computing the reaction mapping in the full composition space (i.e., getting $\mathbf{z}(\Delta t)$ from $\mathbf{z}(0)$) due to the additional species reconstruction and reduction steps involved. So dimension reduction alone is not very efficient, except for the fact that the total number of variables that need to carried in the CFD code is reduced, and so fewer scalar equations need to be solved in the CFD code.

The dimension reduction when integrated with ISAT becomes much more efficient. In addition to performing dimension reduction, the reduced mappings, $\mathbf{r}(\Delta t)$ are tabulated using ISAT. In this case, the query vector to ISAT is given as

$$\mathbf{x} = \{\mathbf{r}(0), h_s^n(0), \Delta t\}, \tag{3.11}$$

where $h_s^n \equiv h_s^n(h, \mathbf{r})$ is the nominal sensible enthalpy defined only in terms of the reduced variables $\mathbf{r}$, for example, as

$$h_s^n \equiv h - \mathbf{h}^{f,r^T} \mathbf{z}^r, \tag{3.12}$$

where $\mathbf{h}^{f,r}$ denotes the n_{rs}-vector of molar enthalpies of formation of the represented species.

The mapping obtained from ISAT is given as

$$\mathbf{f} = \{\mathbf{r}(\Delta t), h_s^n(\Delta t), T^a(\Delta t)\}, \tag{3.13}$$

where $T^a(\Delta t)$ denotes an approximated temperature after time step Δt. (In Fig.3.6, instead of $\mathbf{x}$ and $\mathbf{f}$ we only show $\mathbf{r}(0)$ and $\mathbf{r}(\Delta t)$ for brevity.) Since with dimension reduction the composition of only the represented species is stored in the ISAT table, the thermodynamic variables like temperature and density need to be approximated using the reduced representation, which is explained in more detail in Appendix A.

In the combined approach shown in Fig.3.6, the exact mapping is computed only when a *grow* or *add* is performed in ISAT. For a successful retrieve, the reduced mapping is returned directly using the stored information in ISAT. Since the size of the query vector $\mathbf{x}$ and the mapping vector $\mathbf{f}$ are smaller, the ISAT table size is also smaller, thus resulting in faster search and retrieve times.

To summarize, the combined dimension reduction and tabulation strategy has the following advantages:

1. Fewer species need be carried and solved for in the CFD code – a significant reduction in computational cost.
2. The ISAT table size is relatively smaller, leading to faster search and query time.

3.8.3 Reduction-Tabulation Error

In this study, we compare the accuracy and performance of the following three methodologies for representing chemistry:

1. ISAT: using ISAT directly to tabulate chemistry with a detailed chemical mechanism (without any dimension reduction)
2. ISAT+SKELETAL/REDUCED: using ISAT to tabulate chemistry with a skeletal or reduced chemical mechanism

3. ISAT+RCCE: using the combined ISAT-RCCE reduction-tabulation algorithm with represented species selected using GALI

The primary quantity of interest is the accuracy of the *reaction mapping* obtained using these three different methodologies. We compute the error involved in these three methods by considering the error in the reaction mapping obtained using these methods relative to a direct evaluation using ODE integration with the detailed mechanism. The reaction mappings obtained using an ODE integrator (we use DDASAC [15] in this work) typically involve very small errors (relative to these methods) and hence can be deemed accurate. We refer to the error involved in these methods as the *reduction-tabulation* error, which for a direct use of ISAT with a detailed mechanism reduces to simply *tabulation* error (as there is no reduction involved).

We measure the reduction-tabulation error using the PaSR test setup. We first perform PaSR calculations in the full composition space (without dimension reduction) using the detailed mechanism and store particle compositions and their reaction mappings computed using direct evaluation (with DDASAC) after each reaction sub-step. Let the stored initial particle composition be denoted by $\mathbf{z}^{(n)}(0)$ and its mapping (after time step Δt) by $\mathbf{z}^{(n)}(\Delta t)$ for $n = 1 \, to \, N$, where N denotes the total number of stored compositions and their reaction mappings. We also store the initial enthalpy $h^{(n)}(0)$, and initial temperature $T^{(n)}(0)$ (corresponding to $\mathbf{z}^{(n)}(0)$), which are needed for computing the reaction mapping in the aforementioned reduction-tabulation methodologies.

Using these stored compositions, the reduction-tabulation error is computed by considering the error in the reaction mapping obtained using one of the three aforementioned methods. At the stored compositions, we first obtain the com-

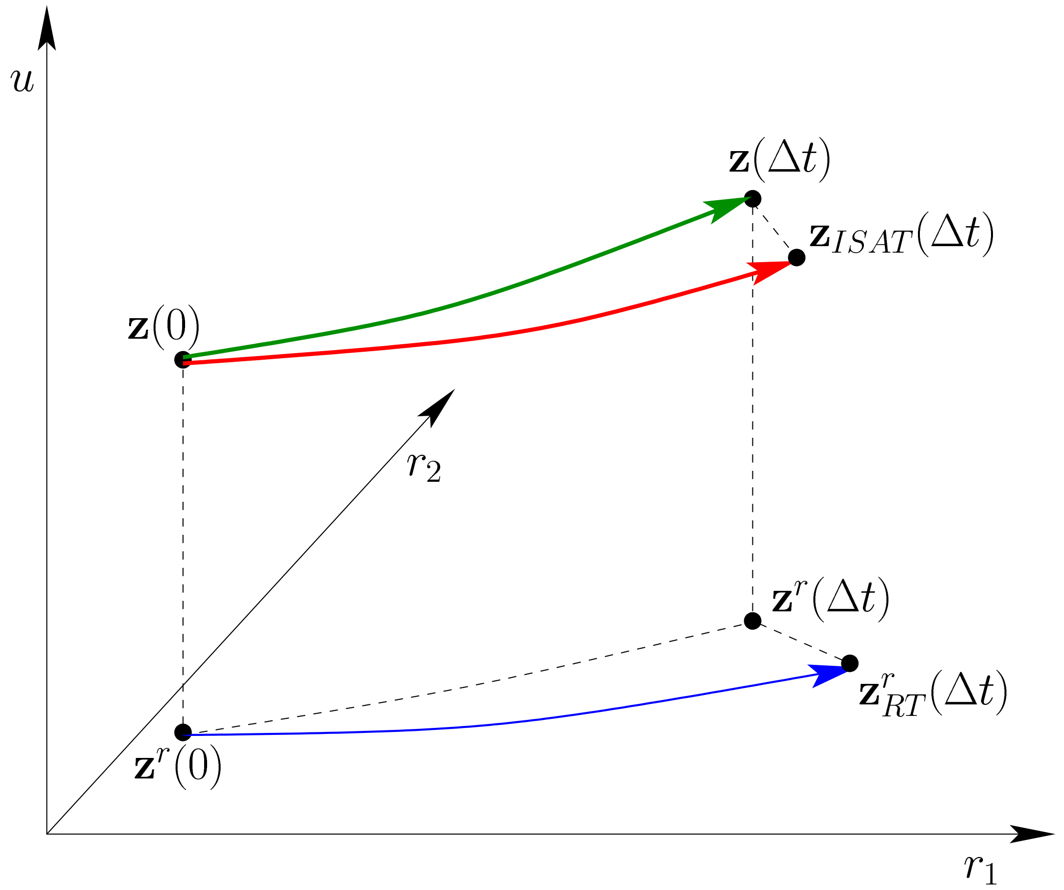


Figure 3.7: The various compositions involved in the computation of reduction-tabulation error. $\mathbf{z}(0)$ denotes an initial composition in the full space. $\mathbf{z}(\Delta t)$ and $\mathbf{z}_{ISAT}(\Delta t)$ denote the reaction mappings obtained with direct evaluation (using DDASAC) and ISAT, respectively in the full composition space using the detailed mechanism. The composition of the represented species at $\mathbf{z}(0)$ is denoted by $\mathbf{z}^r(0)$ and at $\mathbf{z}(\Delta t)$ by $\mathbf{z}^r(\Delta t)$. The reduced mapping obtained using the reduction-tabulation algorithm is denoted by $\mathbf{z}^r_{RT}(\Delta t)$.

position of the represented species used under the considered method. We denote (see Fig.3.7) the reduced representation corresponding to $\mathbf{z}^{(n)}(0)$ by $\mathbf{z}^{r,(n)}(0)$ and that corresponding to $\mathbf{z}^{(n)}(\Delta t)$ by $\mathbf{z}^{r,(n)}(\Delta t)$. The reaction mapping starting from $\mathbf{z}^{r,(n)}(0)$ obtained by the considered reduction-tabulation (RT) method is denoted by $\mathbf{z}^{r,(n)}_{RT}(\Delta t)$.

The reduction-tabulation error, denoted by ϵ_{RT}, is then given as

$$\epsilon_{RT} = \frac{[\mathbf{z}^r(\Delta t) - \mathbf{z}^r_{RT}(\Delta t)]_{rms}}{[\mathbf{z}^r(\Delta t) - \mathbf{z}^r(0)]_{rms}}, \tag{3.14}$$

where the operator $[\,]_{rms}$ is defined previously in Eqn.3.10.

In particular, for the three aforementioned methods, the involved reduced compositions are given as follows:

1. ISAT: since no reduction is involved in this case, we have

   $$\mathbf{z}^r(0) \equiv \mathbf{z}(0), \quad \mathbf{z}^r(\Delta t) \equiv \mathbf{z}(\Delta t) \quad \text{and} \quad \mathbf{z}^r_{RT}(\Delta t) \equiv \mathbf{z}_{ISAT}(\Delta t), \tag{3.15}$$

   where $\mathbf{z}_{ISAT}(\Delta t)$ denotes the reaction mapping obtained from ISAT (see Fig.3.7). The error computed for this case is referred to as the *tabulation* error.

2. ISAT+SKELETAL/REDUCED: in this case, we need to obtain the composition of the species involved in the skeletal or reduced mechanism from the stored full compositions, $\mathbf{z}(t)$. Let $\mathbf{z}^s(t) \equiv \mathbf{z}^s(\mathbf{z}(\mathbf{t}))$ denote the concentration of species in the skeletal mechanism obtained from the full composition, $\mathbf{z}(t)$, then we define

   $$\mathbf{z}^r(t) \equiv \mathbf{z}^s(t)/(\mathbf{w}^{s^T}\mathbf{z}^s(t)), \tag{3.16}$$

   where $\mathbf{w}^s$ denotes molecular weights of the species in the skeletal mechanism, and $\mathbf{z}^s(t)$ is normalized to satisfy the realizability condition, $\mathbf{w}^{s^T}\mathbf{z}^s = 1$. The reaction mapping, $\mathbf{z}^r_{RT}(\Delta t)$, is computed (using the skeletal mechanism) starting from $\mathbf{z}^r(0)$ with the same initial temperature as $\mathbf{z}(0)$.

3. ISAT+RCCE: in this case, for the given set of represented species selected using GALI, the reduced representation is simply the concentration of the represented species denoted by $\mathbf{z}^r$. The reaction mapping $\mathbf{z}^r_{RT}(\Delta t)$ is

obtained using the combined reduction-tabulation algorithm with ISAT-RCCE (see Fig.3.7).

As mentioned earlier, we consider a net reduction-tabulation error of about 2% (1% tabulation and 1% reduction) to be acceptable. In this study, we use an ISAT error tolerance of $\epsilon_{tol} = 10^{-5}$, which gives a tabulation error of less than 1% for both methane/air and ethylene/air premixed combustion in the PaSR. We vary the number of represented species, n_{rs}, in the RCCE method to achieve a combined reduction-tabulation error of less than 2%.

3.9 Results

In this section we present results for the partially-stirred reactor (PaSR) involving premixed combustion of (i) a methane/air mixture and (ii) an ethylene/air mixture with PaSR settings as listed in Table 3.1 and chemistry represented by the chemical mechanisms listed in Table 3.2.

We employ the previously mentioned three methods to represent chemistry - (i) ISAT with a detailed mechanism; (ii) ISAT with a skeletal or reduced mechanism; and (iii) ISAT+RCCE - and compare the reduction-tabulation errors involved in these methods.

We also compare the performance of ISAT under these various methods. To gauge the performance of ISAT under a particular method, we perform a long duration PaSR run involving over 10^9 queries and gather ISAT CPU time statistics. We then compute and analyze the following two quantities:

- *build time*: This is the time taken to add to and to build the table by growing EOAs. It is the total time less the time spent retrieving.
- *query time*: This is the average time taken per query after the ISAT table has been fully built with very few adds or grows being performed in the table.

Out of these two, the primary quantity of interest is the *query time*, which determines the speed of chemistry implementation i.e., the average time per reaction sub-step per particle.

3.9.1 Methane Premixed Combustion

In this section we present results for the methane/air premixed combustion. The PaSR computations involving premixed combustion of stoichiometric methane/air mixture are performed as described in Section 3.4. The chemistry is represented using the GRI Mech 1.2 detailed and the ARM1 reduced mechanisms (details given in Table 3.2).

Represented Species

To select represent species for representing the chemistry with the combined ISAT-RCCE reduction-tabulation algorithm, we apply GALI on the detailed GRI Mech 1.2 mechanism based on the reaction mapping error. The error achieved using the greedy algorithm with $N_G = 2500$ test points and further improvement in error using the local improvement with $N_L = 200$ test points is shown in Fig.3.8. We note that for $n_{rs} \leq 17$, the species reconstruction error obtained by

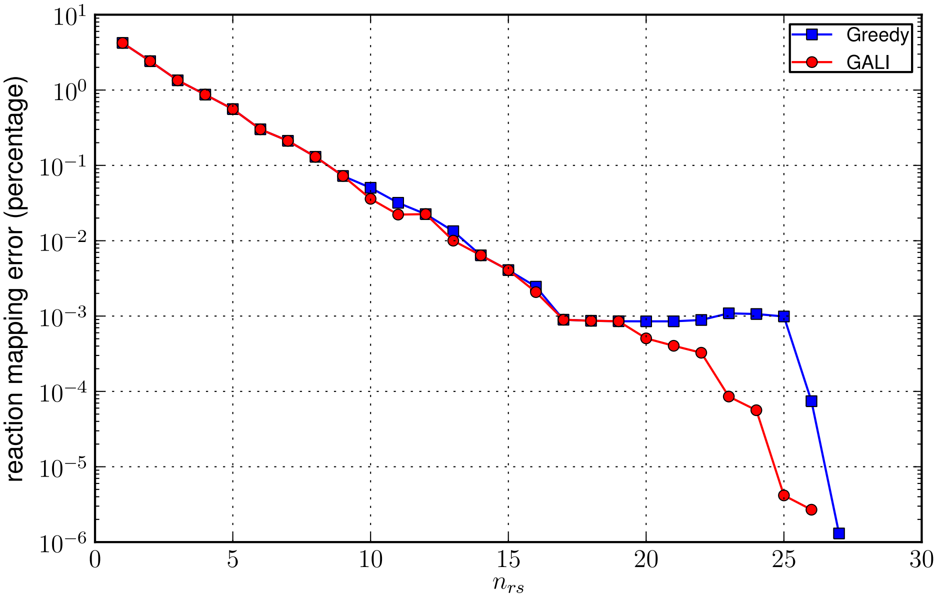


Figure 3.8: Reaction mapping error for methane/air premixed combustion as a function of number of represented species, n_{rs}, obtained with (i) Greedy algorithm (using $N_G = 2500$ test points) and (ii) GALI (using $N_L = 200$ test points for local improvement). The test points are selected from a PaSR run involving methane/air premixed combustion with chemistry represented by the 31-species GRI-Mech 1.2 detailed mechanism.

the greedy algorithm decreases exponentially with n_{rs}, and has a value of about $10^{-3}\%$ at $n_{rs} = 17$. We also note that for $n_{rs} \leq 20$, GALI gives only marginal (less than 1%) improvement in error over the greedy algorithm at some dimensions, but at higher dimensions ($n_{rs} > 20$) we see more than an order of magnitude reduction in error. At any given reduced dimension, n_{rs}, we use the species selected with GALI to perform dimension reduction with RCCE. The sets of represented species obtained using GALI for $n_{rs} = 1$ to 15 are listed in Table 3.3.

Table 3.3: Sets of represented species obtained using GALI (with 31-species GRI-Mech 1.2 mechanism) for dimension reduction of methane/air premixed combustion with RCCE for $n_{rs} = 1$ to 15.

n_{rs}	Represented Species
1	CH_4
2	CH_4, CO_2
3	CH_4, CO_2, H_2
4	CH_4, CO_2, H_2, O_2
5	CH_4, CO_2, H_2, O_2, H
6	CH_4, CO_2, H_2, O_2, H, OH
7	CH_4, CO_2, H_2, O_2, H, OH, O
8	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O
9	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3
10	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, C_2H_6, C_2H_4
11	CH_4, CO_2, H_2, O_2, H, OH, O, H_2O, CH_3, HO_2, CO
12	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O
13	CH_4, CO_2, H_2, O_2, H, OH, O, CH_3OH, CH_3, HO_2, CO, H_2O, CH_2CO
14	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O, CH_2CO, C_2H_5
15	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O, CH_2CO, C_2H_5, CH_2

Reduction-Tabulation Error

The errors incurred for the methane/air premixed combustion with chemistry represented using the three described methods are shown in Fig.3.9. We see that ISAT alone with the detailed mechanism incurs about 0.1% tabulation error and ISAT with the ARM1 reduced mechanism incurs around 6% error. However, with the ISAT-RCCE methodology

- for $n_{rs} < 10$, the error is mainly dominated by the RCCE dimension-reduction error, which decreases exponentially (similar to Fig.3.8) with an increase in the number of represented species, n_{rs}, by about a factor of 10 for every 5 represented species added. (The magnitude of error in Fig.3.8 and Fig.3.9 are different because they are different measures of error.);
- the error is less than ISAT+REDUCED (ARM1 mechanism) with just 8 or more represented species;
- the error is less than 1% with 11 or more represented species; and
- for $n_{rs} \geq 14$, the error (dominantly tabulation) is comparable to ISAT alone, and hence cannot be reduced further by increasing n_{rs}.

ISAT Performance

The ISAT *query time* for PaSR involving methane/air premixed combustion using different methods is shown in Fig.3.10. The ISAT+RCCE method is used with 11 and 15 represented species (which correspond to less than 1% reduction-tabulation errors, see Fig.3.9). We see that the combined reduction-tabulation algorithm provides almost twice the speedup as ISAT alone, with a query time of around 4 to 5 μs, compared to 9 μs using ISAT with the detailed mechanism. The ISAT+REDUCED query time of 5 μs is comparable to ISAT+RCCE. The ISAT build time (estimated by the y-intercepts of dashed-lines) is about the same ($\approx$ 0.6 hours) for all the methods. We also notice that ISAT+RCCE with fewer represented species yields lower query time and overall runtime because the total number of tabulated variables is reduced, and so the ISAT table size is reduced, leading to faster retrieve times.

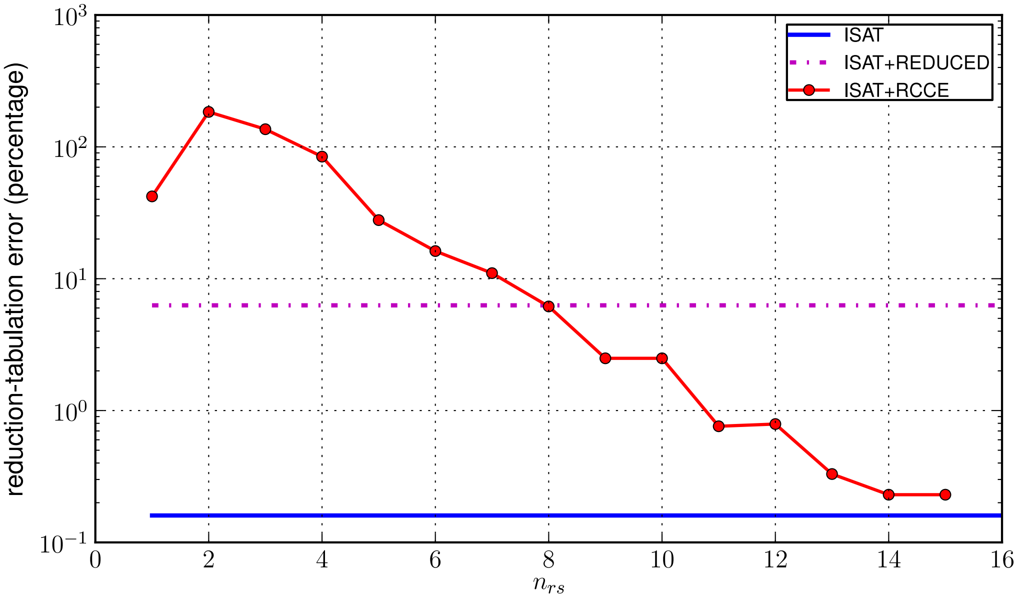


Figure 3.9: Error incurred using (i) ISAT alone (with detailed mechanism); (ii) ISAT+REDUCED (with the ARM1 reduced mechanism); and (iii) ISAT+RCCE (using n_{rs} represented species) with chemistry represented by the detailed mechanism. The error is computed by considering 10^5 compositions and their reaction mappings (computed using ODE integration) obtained from a PaSR run involving methane/air premixed combustion with chemistry represented using the 31-species GRI-Mech 1.2 detailed mechanism.

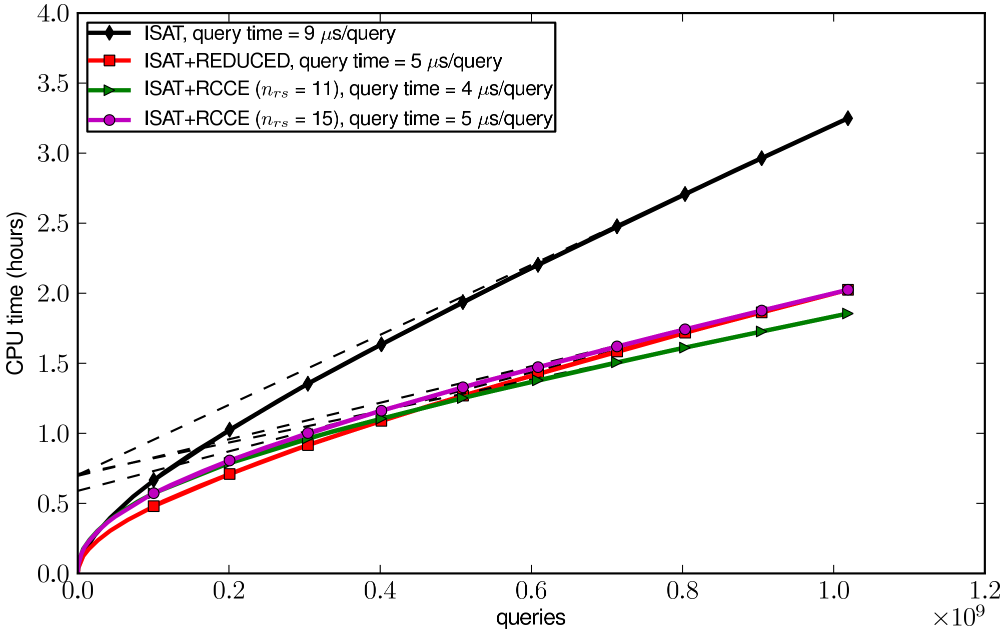


Figure 3.10: ISAT query time for a PaSR run (with over 10^9 queries) involving methane/air premixed combustion using (i) ISAT (with detailed mechanism); (ii) ISAT+REDUCED (with the ARM1 reduced mechanism); and (iii) ISAT+RCCE with n_{rs} represented species and chemistry represented by the detailed mechanism. The y-intercept of linear extrapolation (dashed-line) gives an estimate of the ISAT build time for each case. The CPU time is computed by performing (serial) runs on the TACC Ranger cluster.

Table 3.4: ISAT statistics for the methane/air premixed combustion in PaSR

method	n_{rs}	queries	retrieves	grows	adds	direct evals	CPU time
		$(\times 10^9)$	(percentage)	$(\times 10^4)$	$(\times 10^4)$		(hours)
ISAT	31	1.021	99.99	9.554	4.350	0	3.26
ISAT+REDUCED	16	1.021	99.99	10.034	4.251	0	2.03
ISAT+RCCE	11	1.021	99.99	8.265	1.892	0	1.86
ISAT+RCCE	15	1.021	99.99	8.091	1.836	0	2.03

The complete ISAT statistics for the various methods shown in Fig.3.10 are listed in Table 3.4. We see that in all the methods 99.99% of the queries result in retrieves, which shows the efficient use of ISAT tabulation. We also note that the number of queries resulting in adds in ISAT+RCCE is less than half of that in ISAT and ISAT+REDUCED methods, which as a consequence results in relatively smaller table size and lower query time for ISAT+RCCE method.

3.9.2 Ethylene Premixed Combustion

In this section we present results for the ethylene/air premixed combustion. The PaSR computations involving premixed combustion of stoichiometric ethylene/air mixture are performed as described in Section 3.4. The chemistry is represented using USC-Mech II detailed, skeletal and reduced mechanisms (details given in Table 3.2).

Represented Species

To select represented species for ethylene combustion, the greedy algorithm is applied on the USC-Mech II detailed mechanism. The reaction mapping error obtained using the greedy algorithm for $n_{rs} \leq 32$ with $N_G = 5000$ test points is shown in Fig.3.11. We see that the error decreases exponentially. In the same plot, we also show the error obtained by applying the additional local improvement step using GALI over the initial few values of $n_{rs} \leq 11$, and we see no improvement in error except at $n_{rs} = 3$. Since the GALI algorithm gets expensive at higher dimensions, and we may not get significant improvement in error, so we use the species obtained using the greedy algorithm to perform dimen-

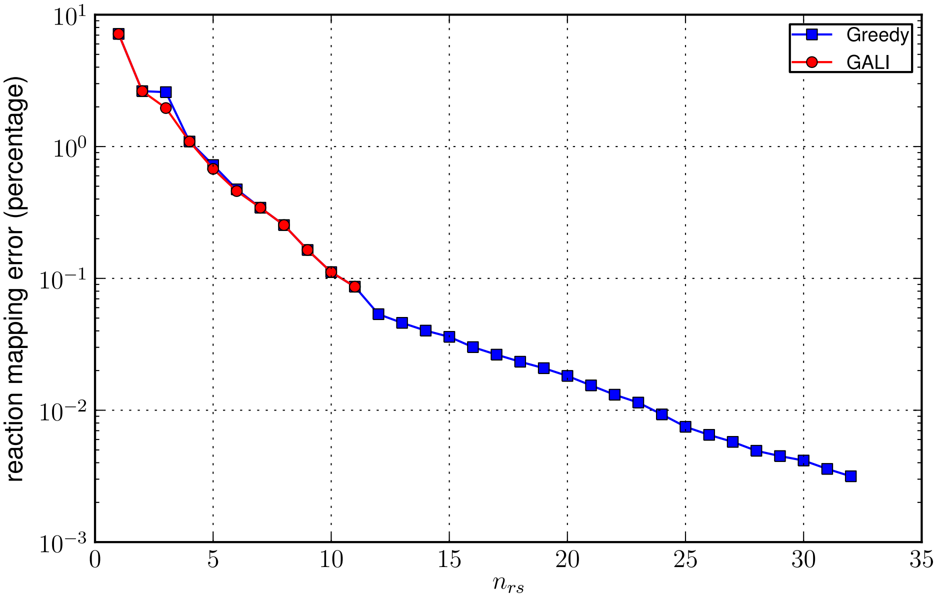


Figure 3.11: Reaction mapping error for ethylene/air premixed combustion as a function of number of represented species, n_{rs}, obtained with (i) Greedy algorithm (using $N_G = 5000$ test points) and (ii) GALI (at some initial values of n_{rs} using $N_L = 500$ test points for local improvement). The test points are selected from a PaSR run involving ethylene/air premixed combustion with chemistry represented by the 111-species USC-Mech II detailed mechanism.

sion reduction using RCCE for ethylene combustion. The represented species obtained with the greedy algorithm for $n_{rs} \leq 32$ are listed in Table 3.5.

Reduction-Tabulation Error

The errors incurred for the ethylene/air premixed combustion using different methods are shown in Fig.3.12. We see that ISAT alone with the detailed mechanism results in slightly less than 1% tabulation error; ISAT with the skeletal mechanism results in slightly over 3% error; and ISAT with the reduced

Table 3.5: Represented species selected using the greedy algorithm (with 111-species USC-Mech II mechanism) for dimension reduction of ethylene/air premixed combustion using RCCE for $n_{rs} = 1$ to 32.

Represented Species
C_2H_4, O_2, CO_2, H_2O, OH, H, CH_2O, HO_2, O, H_2, CH_4, C_2H_2, C_2H_6, H_2O_2, C_2H_3, CH_2CO, C_3H_8, HCCO, CH_2, CH_3, CO, CH_2CHO, C_3H_6, HCO, C_2H_5, C_4H_2, C_4H_6, CH_3O, C_2H, HCCOH, CH, aC_3H_5

mechanism results is slightly over 7% error. However, the combined reduction-tabulation methodology using ISAT+RCCE incurs error

- less than ISAT+REDUCED with just 18 or more represented species; and
- less than ISAT+SKELETAL with just 25 or more represented species; and
- less than 2% error with 31 or more represented species,

which shows that the combined reduction-tabulation approach shows good error control even with relatively large mechanisms involving more than 100 species.

ISAT Performance

The query time computed for PaSR involving ethylene/air premixed combustion is shown in Fig.3.13. Here the ISAT+RCCE is tested with 21, 25 and 32 represented species (which correspond to about 5%, 3% and 1% reduction-tabulation errors respectively, see Fig.3.12). We see that ISAT-RCCE provides more than fifteen-fold speedup, with a query time of around 24 μs compared to 400 μs using ISAT alone with the detailed mechanism. The query times for

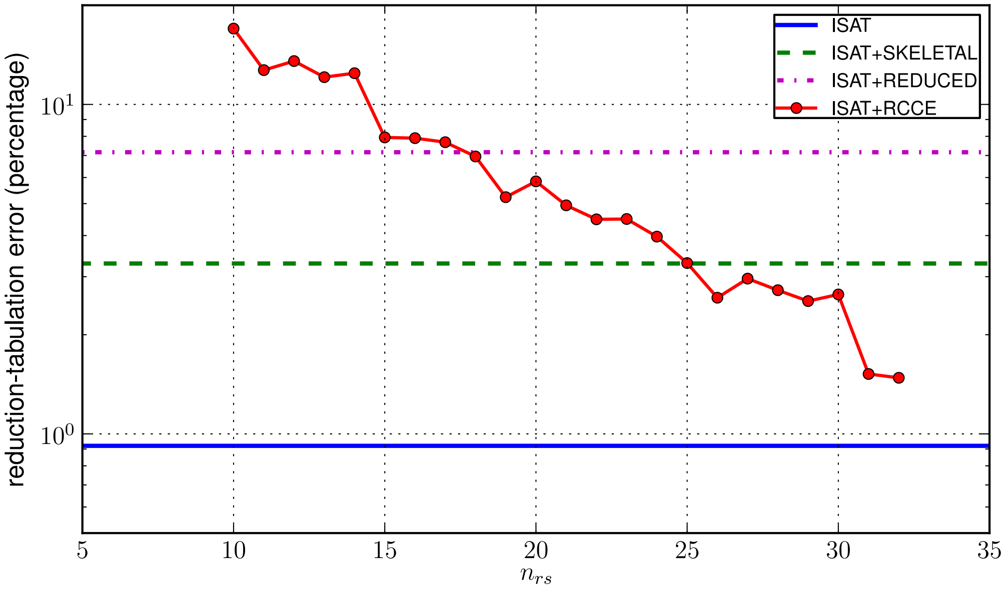


Figure 3.12: Error incurred using (i) ISAT alone (with detailed mechanism); (ii) ISAT+SKELETAL (using ISAT with skeletal mechanism); (iii) ISAT+REDUCED (using ISAT with reduced mechanism); and (iv) ISAT+RCCE (using n_{rs} represented species) with chemistry represented by the detailed mechanism. The error is computed by considering 10^5 compositions and their reaction mappings (computed using ODE integration) obtained from a PaSR run involving ethylene/air premixed combustion with chemistry represented using the 111-species USC-Mech II detailed mechanism.

ISAT+RCCE are comparable to ISAT+SKELETAL/REDUCED query times. In this case, the build times for ISAT+RCCE (around 20 hours) are significantly higher than for ISAT+SKELETAL/REDUCED (about 3 hours), owing to the expensive adds performed in ISAT+RCCE in the initial stages which involve species reconstruction and evaluation of reaction mapping based on the detailed mechanism (as described in Section 3.6.1). However, when applied to LES/PDF computations which require $\mathcal{O}(10^5)$ CPU hours, the build time for ISAT+RCCE is still insignificant; and typically for such large runs, the ISAT table can be built

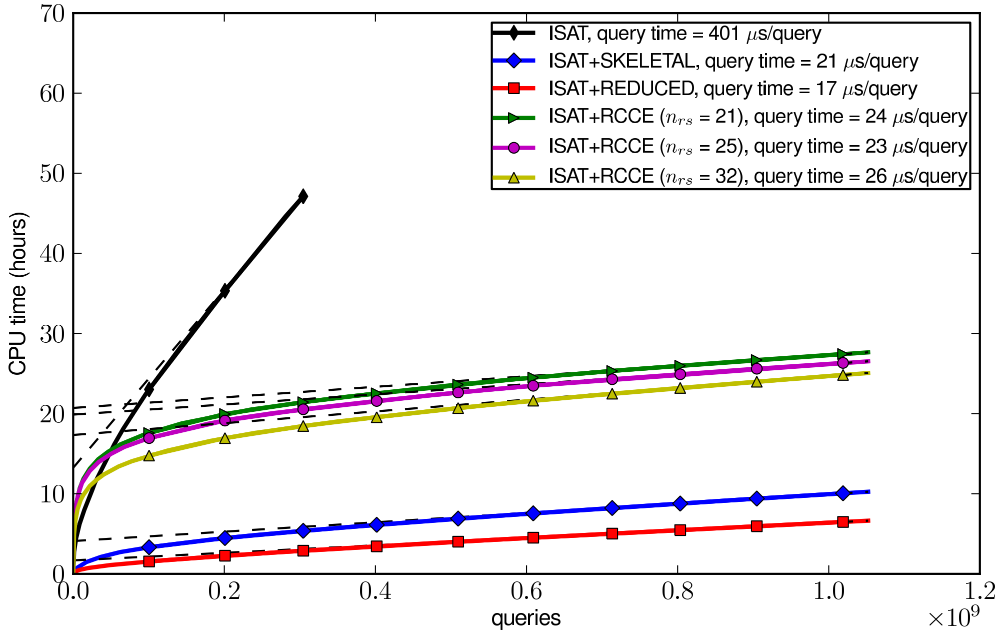


Figure 3.13: ISAT query time for a PaSR run (of maximum 48 hours with over 10^9 queries) involving ethylene/air premixed combustion using (i) ISAT (with detailed mechanism); (ii) ISAT+SKELETAL (using ISAT with skeletal mechanism); (iii) ISAT+REDUCED (using ISAT with reduced mechanism); and (iv) ISAT+RCCE with n_{rs} represented species and chemistry represented by the detailed mechanism. The y-intercept of linear extrapolation (dashed-line) gives an estimate of the ISAT build time for each case. The CPU time is computed by performing (serial) runs on the TACC Ranger cluster.

once, saved and reused for later computations.

In this case with ISAT-RCCE (unlike for the methane/air premixed combustion in Fig.3.10) we notice that:

1. the query time does not consistently decrease as the number of represented species is reduced; and
2. the overall runtime increases as the number of represented species is re-

duced.

This is presumably because the ISAT tables in the ISAT-RCCE cases have not have been fully built yet due to the large build time of around 20 hours. The estimates for the query time for such a short run may not be very accurate. As the number of represented species is reduced, we should expect to see a reduction in the query time (and thus the overall runtime) for longer runs involving 10^{11} to 10^{12} queries.

The full ISAT statistics for the various runs shown in Fig.3.13 are listed in Table 3.6. We again see that in all the methods more than 99.4% of the queries result in retrieves showing the high efficiency of ISAT tabulation. In this case, owing to the large number of species present in the detailed (111-species) and skeletal (38-species) mechanisms, the ISAT table size (1 GB) gets filled up quickly in the ISAT and ISAT+SKELETAL methods, resulting in direct evaluation (DE) of the unresolved queries using the expensive ODE integration. For ISAT alone and ISAT+SKELETAL methods, more than 1.4 and 0.1 million queries result in DEs, respectively. Here again, we note that ISAT+RCCE results in relatively fewer adds compared to the combined adds + DEs in ISAT alone and ISAT+SKELETAL/REDUCED methods.

Table 3.6: ISAT statistics for the ethylene/air premixed combustion in PaSR

method	n_{rs}	queries	retrieves	grows	adds	direct evals	CPU time
		$(\times 10^9)$	(percentage)	$(\times 10^5)$	$(\times 10^4)$	$(\times 10^6)$	(hours)
ISAT	111	0.304	99.49	1.447	0.498	1.392	47.12
ISAT+SKELETAL	38	1.052	99.94	5.070	3.692	0.125	10.25
ISAT+REDUCED	24	1.052	99.95	4.565	7.658	0	6.64
ISAT+RCCE	21	1.052	99.97	2.927	3.010	0	27.65
ISAT+RCCE	25	1.052	99.97	3.005	2.806	0	26.53
ISAT+RCCE	32	1.052	99.97	2.675	2.559	0	25.07

3.10 Conclusions

Based on the results presented here, we can draw the following conclusions:

- The reduction-tabulation error results show that the combined ISAT-RCCE-GALI algorithm can achieve errors comparable to ISAT alone (in this case for $\epsilon_{tol} = 10^{-5}$) using a relatively small number of represented species: 14 out of 31 species for the methane/air premixed case (see Fig.3.9) and 32 out of 111 species for the ethylene/air premixed combustion (see Fig.3.12).
- The ISAT-RCCE-GALI algorithm is also seen to achieve reduction-tabulation errors lower than ISAT with skeletal or reduced mechanisms with relatively fewer number of represented species. In methane/air premixed combustion, ISAT+REDUCED (16-species ARM1) error is achieved with ISAT-RCCE with just 8 or more represented species (see Fig.3.9); and in ethylene/air premixed combustion, ISAT-REDUCED (24-species) and ISAT-SKELETAL (38-species) errors are achieved with ISAT-RCCE with just 18 and 25 represented species, respectively (see Fig.3.12).
- The ISAT-RCCE-GALI algorithm is also computationally efficient resulting in a query time of around 4 μs (two-fold speedup) for the methane/air premixed combustion using 11 to 15 represented species compared to 9 μs using ISAT with 31-species detailed mechanism (see Fig.3.10); and a query time of around 24 μs (fifteen-fold speedup) for ethylene/air premixed combustion using 21 to 32 represented species compared to 400 μs using ISAT with 111-species detailed mechanism (see Fig.3.13).
- The combined approach shows both good error control and performance.

With fewer species to track in the CFD code, this combined ISAT-RCCE-GALI approach provides a computationally accurate and efficient way of representing combustion chemistry.

Here, we would also like to mention that the combined reduction-tabulation procedure described in this work can also be used with the ICE-PIC dimension reduction method [79, 77]. The GALI algorithm can be used to select represented species for ICE-PIC and the combined ISAT-ICE-PIC-GALI approach can be applied for representing combustion chemistry.

3.11 Acknowledgements

This work is supported in part by Department of Energy under Grant DE-FG02-90ER and by Grant Number FA9550-09-1-0611 funded by AFOSR and NASA ARMD, technical monitors Julian Tishkoff (AFOSR) and Aaron Auslender and Rick Gaffney (NASA). We are grateful to Jon Kleinberg for discussions on local improvement algorithm for GALI. SBP has a financial interest in Ithaca Combustion Enterprise, LLC., which has licensed the software ISAT-CK and CEQ used in this work.

CHAPTER 4

COMPUTATIONALLY-EFFICIENT AND SCALABLE PARALLEL IMPLEMENTATION OF CHEMISTRY IN SIMULATIONS OF TURBULENT COMBUSTION[†]

4.1 Abstract

Large scale combined *Large-Eddy Simulation* (LES)/*Probability Density Function* (PDF) parallel computations of reactive flows with detailed chemistry involving large numbers of species and reactions are computationally expensive. Among the various techniques used to reduce the computational cost of representing chemistry, the three approaches in widest use are: (1) *mechanism reduction*, (2) *dimension reduction*, and (3) *tabulation*. In addition to these approaches, in large scale parallel LES/PDF computations, we need strategies to distribute the chemistry workload among the participating cores to reduce the overall wall clock time of the computations. Here we present computationally-efficient strategies for implementing chemistry in parallel LES/PDF computations using *in situ* adaptive tabulation (ISAT) and *x2f_mpi* – a Fortran library for parallel vector-valued function evaluation (used with ISAT in this context). To test the strategies, we perform LES/PDF computations of the Sandia Flame D with chemistry represented using a) a 16-species augmented reduced mechanism; and b) a 38-species C_1-C_4 skeletal mechanism. We present three parallel strategies for redistributing the chemistry workload, namely (a) PLP, purely local processing; (b) URAN, the uniform random distribution of chemistry computations

[†] 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(12):3096-3109, 2012.

among all cores following an early stage of PLP; and (c) P-URAN, a Partitioned URAN strategy that redistributes the workload only among partitions or subsets of the cores. We show that among these three strategies, the P-URAN strategy (i) yields the lowest wall clock time, which is within a factor of 1.5 and 1.7 of estimates for the lowest theoretically achievable wall clock time for the 16-species and 38-species mechanisms, respectively; and (ii) for reaction, achieves a relative weak scaling efficiency of about 85% when scaling from 2,304 to 9,216 cores and a relative strong scaling efficiency of over 60% when scaling from 1,152 to 6,144 cores.

4.2 Introduction

Computations of turbulent combustion flows using detailed chemistry involving a large number of species and reactions are computationally expensive. Modern chemical mechanisms of real fuels involve hundreds or thousands of species and thousands of reactions [64, 100]. Incorporating such detailed chemistry in the combustion flow calculations is computationally prohibitive. Among the various efforts put into reducing the computational cost of representing chemistry, the three approaches in widest use are: (1) *mechanism reduction* to reduce the number of species and reactions involved [53, 60, 58]; (2) *dimension reduction* to represent chemistry using a reduced number of variables [41, 44, 55, 79]; and (3) *tabulation* to reduce significantly the cost of expensive evaluations of the reaction mappings involving ODE integrations [66, 49, 92, 93]. In recent times, combined methodologies have also been developed, wherein reduced reaction mechanisms or dimension reduction methods are used in conjunction with tabulation [31, 32, 76, 75].

Due to the high computational cost of turbulent combustion problems, most of the modern day simulations are performed in parallel on multiple cores using distributed computing. Thus, in addition to the aforementioned techniques, when performing large scale parallel LES/PDF computations, strategies are needed to efficiently distribute the chemistry workload among the participating cores to reduce the overall wall clock time of the computations [51, 50, 48, 86]. In this paper we present parallel strategies for the implementation of chemistry using *in situ* adaptive tabulation (ISAT) [49] and *x2f_mpi* – a Fortran 90/95 library for parallel vector-valued function evaluation (used with ISAT in this context) [48]. The parallel strategies are tested by performing LES/PDF simulations of the Sandia Flame D [9].

The work here is presented mainly in the context of large scale parallel LES/PDF computations of turbulent reactive flows, in which the ISAT algorithm has proved to be particularly efficient at reducing the computational cost by more than two to three orders of magnitude [49, 32]. However, the ISAT algorithm has been successfully applied in many other fields like chemical engineering [43], control [28], and solid mechanics [8]; and has also been implemented in the commercial CFD package ANSYS Fluent [76, 75]. Hence, the parallel strategies presented in this work for implementing chemistry in LES/PDF computations may have wider applications in other fields also.

The outline of the remainder of the paper is as follows: in Section 4.3 we describe our hybrid LES/PDF solver; in Section 4.4 we describe the parallel strategies implemented using ISAT and *x2f_mpi* for implementing chemistry; in Section 4.5 we describe the computational details for simulating Sandia Flame D; in Section 4.6 we present performance results for different parallel strategies;

and finally in Section 4.7 we state our conclusions.

4.3 Hybrid LES/PDF Solver

4.3.1 LES Solver

In this study we use an LES solver based on a Stanford LES code [62, 63]. The LES solver solves the Eulerian transport equations for mass, momentum and scalars using finite difference methods on structured non-uniform grids. It supports both Cartesian and cylindrical coordinate systems; is second order accurate in space and time; and is parallelized (using MPI) by domain decomposition in two dimensions.

4.3.2 PDF Solver

We use the PDF solver, *HPDF*, developed at the Turbulence and Combustion Research Group at Cornell [97]. The HPDF solver has second-order accuracy in space and time; supports Cartesian and cylindrical coordinate systems; is parallelized (using MPI) by domain decomposition in two dimensions; and has a general interface to facilitate coupling with existing LES (or RANS) solvers. In this work we use the “one-way” coupling in our LES/PDF solver as described in [97], i.e., LES flow field data is used in the PDF solution, however there is no feedback of temperature and density from the PDF to LES solution. The LES solver uses its own assumed-PDF Flamelet model to obtain density and temperature. However, the thermochemical statistics reported in this work are collected from the PDF particle data.

In HPDF, the thermochemical composition of the fluid within the solution domain is represented by a large number of particles (see Fig.4.1). The HPDF solver has three main components which account for:

1. *transport*: the motion of particles in the physical domain, including a random-walk component to represent the effects of subgrid-scale turbulent advection and molecular diffusion;
2. *mixing*: the change in composition of a particle due to mixing with neighboring particles (which models the effects of molecular mixing); and
3. *reaction*: the change in composition of a particle due to chemical reaction.

These components are implemented in *fractional steps* using splitting schemes [98].

In this study we use the first-order $\mathbb{TMR}$ splitting scheme (which is found to perform as well as the second-order splitting scheme for jet flames [97]), which denotes taking fractional steps of *transport*, $\mathbb{T}$; *mixing*, $\mathbb{M}$; and *reaction*, $\mathbb{R}$ in this order on each time-step. The Kloeden and Platen (KP) [42] stochastic differential equation (SDE) scheme is used to integrate the *transport* equations; and the *mixing* is represented using the modified Curl [37] mixing model. In the remainder of this section, we focus on the implementation details of the *reaction* fractional step.

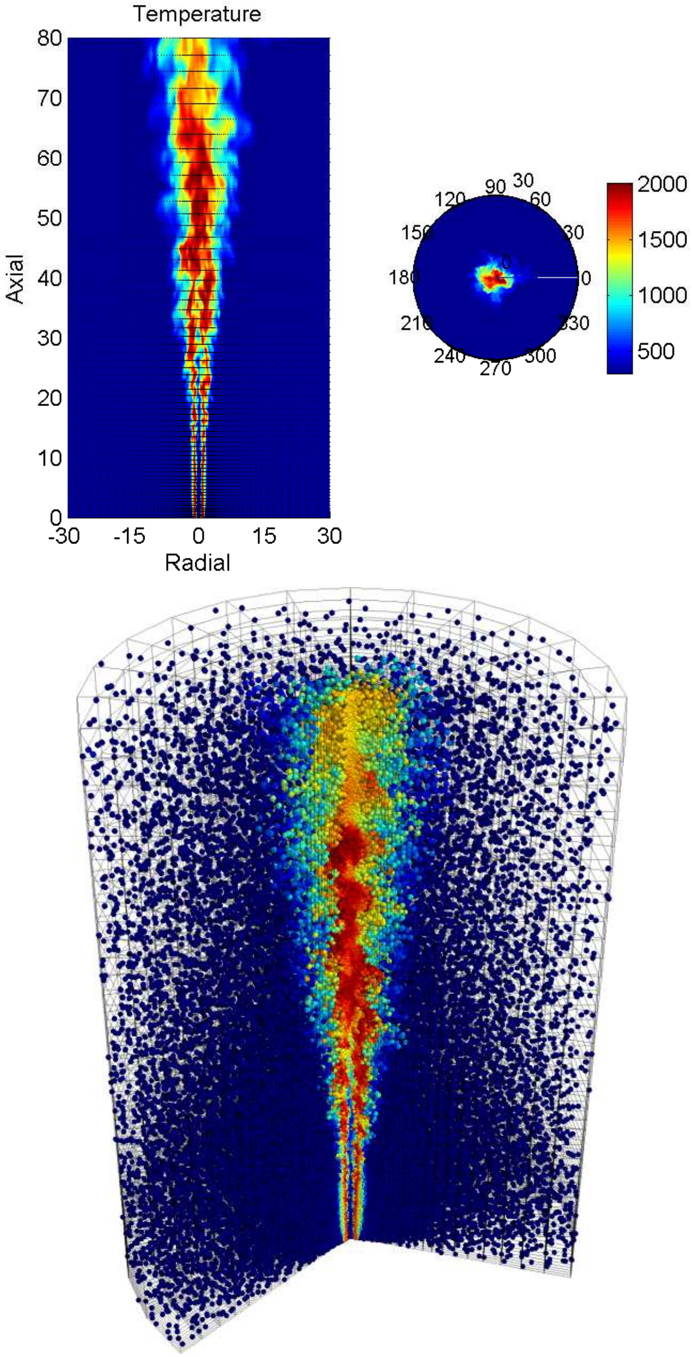


Figure 4.1: LES/PDF simulation of the Sandia Flame D. Top: Instantaneous temperature distribution on a 2D cut-plane through the axis of the computational domain. Dots in the plot indicate every third grid node in the axial and radial directions. Bottom: A 3D slice-view of the PDF particle temperature distribution in the computational domain. Every fourth LES grid line is shown.

4.3.3 Chemistry Representation

We consider a reacting gas-phase mixture consisting of n_s chemical species, composed of n_e elements. The thermochemical state of the mixture (at a given position and time) is completely characterized by the pressure p, the mixture enthalpy h, and the n_s-vector $\mathbf{z}$ of specific moles of the species.

In our LES/PDF computations, we neglect acoustic interactions and compressibility (under the "low Mach number" approximation), and assume that the thermodynamic variables are decoupled from the small variations in pressure about some fixed specified background pressure field, $p = p_0$. Thus only p_0 is coupled to the thermodynamic variables, and (given p_0) the thermochemical state is fully characterized by $\{\mathbf{z}, h\}$. In the HPDF solver, the particles carry the composition $\{\mathbf{z}, h\}$.

In the reaction fractional step, a particle's chemical composition $\mathbf{z}$ evolves (at constant enthalpy h) in time according to the following set of ordinary differential equations (ODEs)

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{S}(\mathbf{z}(t)), \tag{4.1}$$

where $\mathbf{S}$ is the n_s-vector of chemical production rates determined by the chemical mechanism used to represent the chemistry.

The *reaction mapping*, $\mathbf{R}(\mathbf{z}, t)$ is defined to be the solution to Eqn.4.1 after time t starting from the initial composition $\mathbf{z}$. The reaction mapping obtained by directly integrating the set of ODEs given by Eqn.4.1 is referred to as a *direct evaluation* (DE). We use DDASAC [15] for performing ODE integration.

Owing to the large cost of direct evaluation of reaction mappings involving large numbers of species, *in situ* adaptive tabulation (ISAT) is used in the

HPDF solver to reduce the cost of chemistry calculations. In addition, we use the *x2f_mpi* library to distribute the chemistry workload efficiently among the participating cores in large scale parallel LES/PDF simulations. The details of the implementation are discussed in Section 4.4.

4.3.4 Domain Decomposition

The LES computations are performed on structured non-uniform grids in Cartesian or cylindrical coordinate system. We denote the grid used for LES computations by $N_x \times N_y \times N_z$ (in the three principal directions). In performing parallel LES/PDF computations (using the hybrid LES/HPDF solver) on N_c cores, the computational domain is decomposed into N_c sub-domains and each core performs the computations of one sub-domain. The domain decomposition is done in the first two principal directions, and is denoted by $D_x \times D_y$, where $D_x D_y = N_c$. The domain decomposition is done such that N_x and N_y are exact multiples of D_x and D_y, respectively. In addition, the domain decomposition in the LES solver is restricted by the grid size such that $D_x \leq N_x/2$ and $D_y \leq N_y/2$, i.e., each slice in a given dimension must contain at least two grid points. The HPDF solver has the capability to use its own domain decomposition independent of the LES solver, but in the current study, we use the same domain decomposition in both the LES and HPDF solvers.

4.3.5 Compute Cluster Architecture and Parallelization

For the sake of consistency, all the results presented in this work are obtained on the TACC Ranger cluster. Each node on Ranger contains four AMD Opteron Quad-Core 64-bit processors, i.e., 16 cores in all, with 32 GB of memory (2 GB per core).

In our LES/HPDF solver, each sub-domain is assigned to one core (independent of the cluster architecture), and the inter-core communication is implemented using MPI. From the implementation point of view, the intra-node communication between two cores on a single node is not treated differently from the inter-node communication between two cores on two different nodes. This ensures the code is highly robust and portable, properties that would be difficult to achieve with a hybrid parallel implementation blending multithreading with message passing [73]. Obviously, though, the actual MPI core-to-core communication time is affected by details of the cluster architecture and connectivity; the influence of these factors will be discussed in Section 4.6.4.

4.4 Parallel Strategies for Implementing Chemistry

In performing parallel LES/PDF computations with chemistry tabulation using ISAT, each core has its own ISAT table for tabulating the chemistry. On the reaction fractional step, the reaction mappings for all the particles in the computational domain need to be evaluated. A particle whose reaction mapping has been evaluated is called a *resolved* particle; and the act of resolving a particle by successfully retrieving a linear approximation to the reaction mapping from an ISAT table is called a *retrieve*. In parallel computations, given a particle on a core, the following ISAT operations can be invoked using *x2f_mpi* in an attempt to resolve the particle:

1. attempt to retrieve from the local ISAT table (also referred to as a "quick try");
2. if "quick try" fails, make one (or more) attempt(s) to retrieve from remote ISAT table(s);
3. if all the retrieve attempts fail, do a direct evaluation (followed by addition to the ISAT table) on the local core or on a remote core.

The goal is of course to resolve all the particles in the minimum possible wall clock time by redistributing the chemistry workload among all the cores. But this is not a trivial task because:

1. the time required to resolve a given particle is unknown ahead of time;
2. the time to resolve a particle may vary by 4 orders of magnitude, as the retrieve time from an ISAT table is typically $\mathcal{O}(10)\mu s$ while a direct evaluation may take $\mathcal{O}(10^5)\mu s$ (using DDASAC [15] for mechanisms involving 100 or more species);
3. furthermore, the probability of retrieving from an ISAT table depends on the history and duration of the run.

The *x2f_mpi* library [48] is used as an interface between the HPDF solver (for the *reaction* fractional step) and the ISAT tables to redistribute the chemistry workload (of resolving all the particles) efficiently among the participating cores to reduce the overall wall clock time of the computations.

A thorough discussion of the use of *x2f_mpi* library in conjunction with ISAT to redistribute chemistry workload is provided in [48]. Two of the strategies, namely (i) Purely Local Processing (PLP) and (ii) Uniform Random Distribution (URAN), presented in [48], are described here in the context of LES/PDF computations. Additionally, here we present a new strategy, Partitioned URAN (P-URAN), which is shown to perform better than the PLP and URAN strategies, and which scales well to large number of cores.

1. Purely Local Processing (PLP):

   In this strategy, all the particles on a core are resolved (i.e., the reaction mapping is evaluated) using the local ISAT table without any message passing or load redistribution. This in some sense is the same as invoking ISAT directly from HPDF on each core without using the *x2f_mpi* interface.

   The main advantages of this strategy are:

   (a) ease of implementation;

   (b) no communication cost;

   (c) higher probability of retrieving particles from the local table;

   and the main disadvantage is:

   (a) load imbalance, especially between cores handling computation of sub-domains in the reactive zone versus cores handling computation of sub-domains in the coflow/air, leading to relatively high wall clock time.

2. Uniform Random Distribution (URAN):

   This strategy aims at achieving statistically ideal load balancing by evenly distributing the chemistry workload among all the participating cores. The strategy involves one initial HPDF step of PLP to initialize the local ISAT tables. In the subsequent steps, on each core, a "quick try" is first made to attempt to resolve particles by retrieving from the local ISAT table; following this, there is a uniform random distribution of all the unresolved particles to all the cores. This strategy thus ensures that every core receives (approximately) the same number of particles to resolve, with a similar distribution of particles from the reactive and non-reactive zones of the computational domain.

The main advantage of this strategy is:

(a) close to ideal load balancing, after the initial "quick try" lookup;

and the disadvantages are:

(a) relatively costly all-to-all communication;

(b) lower probability of retrieving particles (due to the random distribution of unresolved particles over all the cores);

(c) poor scaling (to large number of cores) due to all-to-all communication.

3. Partitioned Uniform Random Distribution (P-URAN):

   This is a new strategy which is a combination of the previous two, PLP and URAN, strategies. This strategy works in two stages: in stage 1 (for a specified duration of time) the PLP strategy is used to resolve particles on all cores at each time step; then in stage 2 (for the remainder of the time steps), the participating cores are partitioned into smaller groups, and within each partition the URAN strategy is used to uniformly distribute the chemistry workload among the cores in that partition.

The advantages of this strategy are:

(a) relatively higher probability of retrieving from the local tables due to the initial PLP stage;

(b) reduced communication cost compared to URAN (communication restricted to within smaller partitions);

(c) good load balancing within partitions;

(d) good scaling to large number of cores;

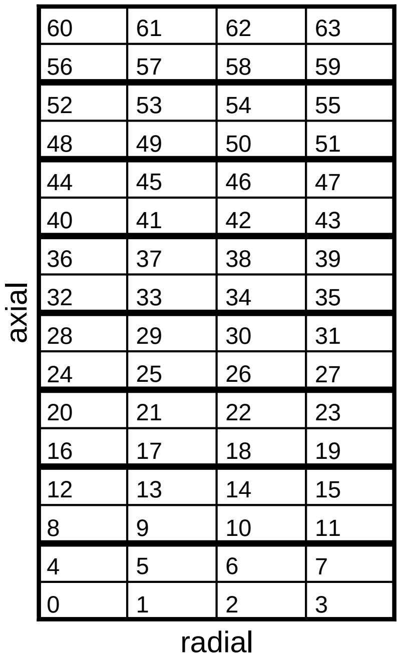
and some of the disadvantages are:

(a) load imbalance among different partitions;

(b) the need to determine additional parameters: (i) duration of the PLP stage; and (ii) size of the partitions.

To specify the parameters used in the P-URAN strategy, in the remainder of the text we use the notation: P-URAN[τ, κ], where τ denotes the time (in hours) spent in the PLP stage (in addition to the first initialization time step); and κ denotes the size of the partitions, i.e., partitions of κ cores are formed from the overall N_c cores used in the computations (which means that the number of partitions used is N_c/κ).

Since the dominant load imbalance is caused in the radial direction in the simulation of jet flames, in this study we choose κ to be a multiple of the domain decomposition in the radial direction, D_y, and form partitions in the axial direction by grouping together the domains in the radial direction as shown in the schematic Fig.4.2 for applying P-URAN[τ, $\kappa = 8$] strategy with $N_c = 64$, $D_x = 16$, $D_y = 4$. (As discussed later in Section 4.6.4, it



60	61	62	63
56	57	58	59
52	53	54	55
48	49	50	51
44	45	46	47
40	41	42	43
36	37	38	39
32	33	34	35
28	29	30	31
24	25	26	27
20	21	22	23
16	17	18	19
12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3

Figure 4.2: A schematic showing LES/HPDF domain decomposition of 16×4 for $N_c = 64$ cores (indicated by cores ranked from 0 to 63), and formation of partitions of size $\kappa = 8$ (indicated by thick lines) for applying the P-URAN[τ, $\kappa = 8$] strategy.

is also desirable to have D_y and κ to be exact multiples of the number of cores per node on the compute cluster (for example 16 on Ranger)).

In addition to the PLP and URAN strategies, in [48] two more strategies are presented: (i) the preferential distribution (PREF) strategy; and (ii) an "on the fly" *adaptive* distribution strategy, which blends PLP, URAN and PREF. In [48], the PREF and *adaptive* strategies are tested using the Partially-Stirred Reactor (PaSR) using up to 64 cores, and are found to perform better than the PLP and URAN strategies. However, in the current study, we do not find the PREF and *adaptive* strategies performing any better than the PLP or URAN strategy when applied to the LES/PDF simulation of the Sandia Flame D using more than 1,000 cores, presumably due to one or more of the following reasons:

1. LES/PDF simulation of the Sandia Flame D exhibits significantly more load imbalance (among the jet, pilot and coflow regimes) compared to the PaSR setup used in [48], and so efficient redistribution of chemistry workload is harder for Sandia Flame D;
2. the PREF and *adaptive* strategies involve significantly more MPI communication than other strategies, and the communication may not scale well to a large number of cores (more than 1,000) used in the current study.

We are still investigating these reasons, but nonetheless, the new P-URAN strategy presented here is shown to perform within a factor of 1.5 to 1.7 of estimates for the best that can be achieved (in terms of simulation wall clock time); and scales well up to 9,216 cores.

4.5 LES/PDF Simulation of Sandia Flame D

To test the chemistry implementation we perform LES/PDF simulations of the Sandia Flame D.

4.5.1 Sandia Flame D

The Sandia Flame D is a piloted CH_4/Air jet flame operating at a jet Reynolds number, Re = 22,400. All the details about this flame and the burner geometry can be found at [9]. Here we mention only some of the important aspects of this flame.

The jet fluid consists of 25% CH_4 and 75% air by volume. The jet flows in at 49.6 m/s velocity at 294K temperature and 0.993 atm pressure. The jet diameter, $D = 7.2$ mm. The pilot is a lean (equivalence ratio, $\Phi = 0.77$) mixture of C_2H_2, H_2, air, CO_2, and N_2 with the same nominal enthalpy and equilibrium composition as that of CH_4/Air at this equivalence ratio. The pilot velocity is 11.4 m/s. The coflow is air flowing in at 0.9 m/s at 291K and 0.993 atm.

4.5.2 Computational Details

We perform LES/PDF simulation of the Sandia Flame D using the coupled LES/HPDF solver. The simulation is performed in a cylindrical coordinate system. A computational domain (see Fig.4.1) of $80D \times 30D \times 2\pi$ is used in the axial, radial and azimuthal directions, respectively. A non-uniform structured grid of size 192×192×96 (in the axial, radial and azimuthal directions, respectively) is used for the LES solver. In the HPDF solver (for the base case), the number of particles per LES cell (N_{pc}) used is $N_{pc} = 40$. With a total of 192×192×96 ≈ 3.5×10^6 LES cells, an overall 140×10^6 particles are used in the computational domain.

To represent the chemistry we consider two different mechanisms:

1. 16-species augmented reduced mechanism (ARM1) [88]; and
2. 38-species C_1-C_4 skeletal mechanism [23]. (This mechanism is developed especially for ethylene combustion, but is also applicable to methane

flames.)

In this study, we are not interested in comparing the accuracy of representing chemistry using these two mechanisms, but are only interested in studying the parallel performance of chemistry implementation using these two mechanisms involving different numbers of species.

A fixed ISAT error tolerance, $\epsilon_{tol} = 10^{-4}$ (which yields less than 3% tabulation error for both the mechanisms), is used in this study. In addition, we specify a maximum allowed ISAT table size, S, per core. In the simulations, when an ISAT table on a core becomes completely filled, then subsequent unresolved queries on that core are resolved using direct evaluation. We typically specify a maximum ISAT table size of $S \leq 1$ GB because for tables of size over 1 GB, the *search* and *add* times in ISAT become large and are sometimes comparable to direct evaluation time. For the 16-species mechanism we specify a maximum ISAT table size of $S = 600$ MB per core; and for the 38-species mechanism we specify $S = 1,000$ MB per core. In simulations with the 16-species mechanism, none of the tables become filled over the duration of the runs covered in this report, however with the 38-species mechanism some tables (near the flame front with the PLP or P-URAN strategy) reach the maximum specified size limit. (For mechanisms involving over 40 species, we typically use a dimension reduction method like the rate-controlled constrained-equilibrium (RCCE) [41, 32] to reduce the number of tabulated variables to 20-30, thereby reducing the ISAT table size [30]).

4.6 Results

The LES/PDF simulation tests are performed in three phases (more computational details are given in the later sections):

- Phase 1: Base Case.

  In this phase we perform an LES/PDF simulation of the Sandia Flame D to obtain a statistically-stationary flame.

- Phase 2: Comparison of Parallel Strategies.

  In this phase, starting from the statistically-stationary base case, we compare the performance of various parallel strategies implemented using *x2f_mpi*.

- Phase 3: Scaling Studies.

  In this phase we perform weak and strong scaling studies with different *x2f_mpi* strategies using up to 9,216 cores.

All the simulations are performed on the Texas Advanced Computing Center (TACC) Ranger cluster.

4.6.1 Base Case

We perform separate LES/PDF simulations of the Sandia Flame D with chemistry represented using the 16-species and 38-species mechanisms on 1,024 cores (using 64x16 domain decomposition) until a statistically-stationary state is reached.

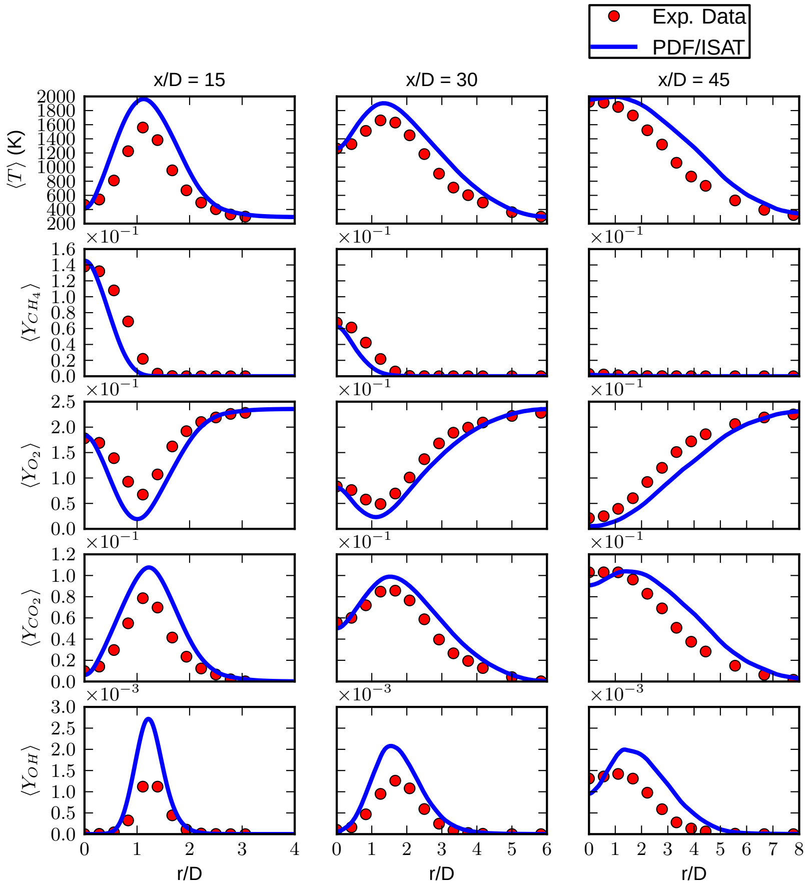


Figure 4.3: Radial profiles of time-averaged density-weighted mean temperature T, and mass fraction of species CH_4, O_2, CO_2, OH at axial locations $x/D = 15, 30, 45$ obtained from experimental data and an LES/PDF simulation using the 38-species mechanism.

After reaching the statistically-stationary state, for the simulation using the 38-species mechanism, we collect statistics for thermochemical quantities from the PDF particle data time-averaged over 10,000 time steps (which corresponds to about three flow through times based on the jet inlet velocity). Fig.4.3 and Fig.4.4 show comparisons of radial profiles of azimuthally-averaged and time-averaged density-weighted mean and standard-deviation statistics with the experimentally measured statistics [9] at axial locations $x/D = 15, 30, 45$. A

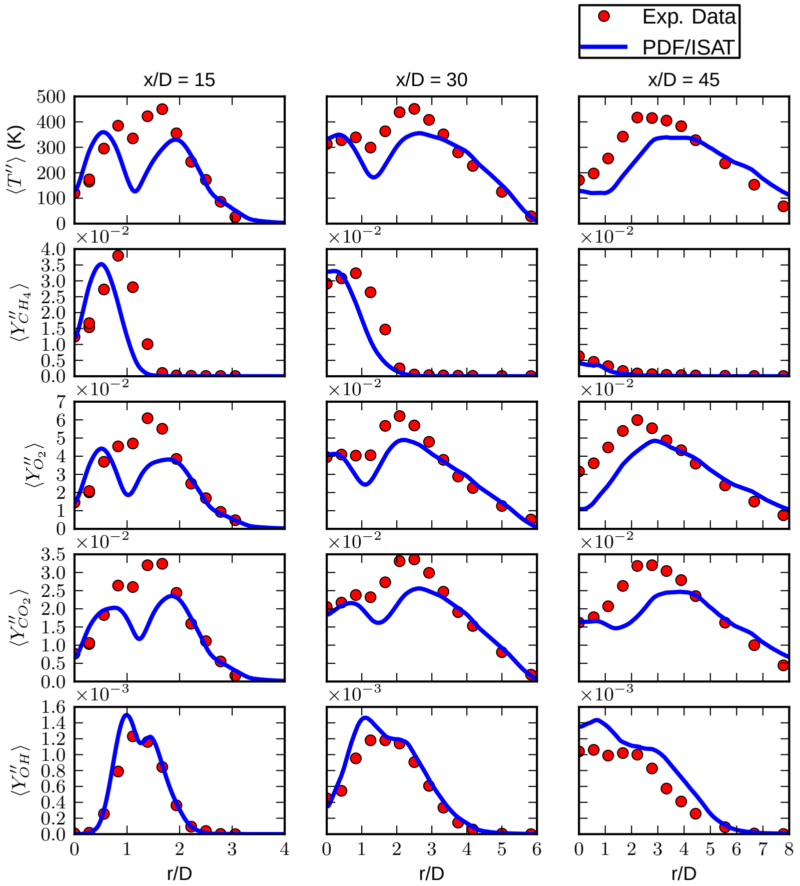


Figure 4.4: Radial profiles of time-averaged density-weighted standard deviation of temperature T, and mass fraction of species CH_4, O_2, CO_2, OH at axial locations $x/D = 15,\ 30,\ 45$ obtained from experimental data and an LES/PDF simulation using the 38-species mechanism.

good qualitative agreement between the simulated and experimentally measured statistics is observed, which is adequate for the current study as the main focus is on the efficient parallel implementation of chemistry. There have been many previous studies of the Sandia Flame D using PDF and LES based methods [101, 74, 85, 57, 59]. A slightly better prediction for the peak value of the mean temperature at $x/D = 15$ is obtained in [101, 74], however overall a similar level of agreement for species mass fractions is observed in these studies.

4.6.2 Comparison of Parallel Strategies

Starting from the respective base cases for the 16-species and 38-species mechanisms, we employ the PLP, URAN and P-URAN parallel strategies, and compare the overall wall clock time for running a fixed number of simulation time steps on 1,024 cores. To test P-URAN, we consider the P-URAN[0.2h,32] strategy for the 16-species mechanism, and the P-URAN[0.1h,32] strategy for the 38-species mechanism. P-URAN sensitivity results to changes in the time spent in the PLP stage (τ) and the partition size (κ) are presented in the next section.

For the 16-species and 38-species mechanisms we perform $N_t = 2,000$ and $N_t = 1,000$ simulation time steps, respectively. (We perform fewer simulation time steps with the 38-species mechanism due to relatively expensive chemistry and limited availability of compute hours on the TACC Ranger cluster.) The timing results using different strategies are shown in Fig.4.5 and Fig.4.6 using bar charts for the 16-species and 38-species mechanisms, respectively. The bars show the breakdown of time spent in LES, HPDF (outside reaction) and Reaction (including *x2f_mpi* communication time, if any). Also shown is the Waiting time, which is indicative of the load imbalance caused by reaction, with a lower bound of zero indicating perfect reaction load balancing, and an upper bound equal to the Reaction time for the extreme case where the complete reaction load is concentrated on a single core at each time step. The method used to compute these wall clock time statistics is explained in Appendix B. In these figures, for comparison, we also show the wall clock time for the case where the chemistry in the LES/HPDF simulation is represented using a single scalar (mixture-fraction) based flamelet implementation (without using ISAT). Additionally we show two estimates of the best wall clock time that can be achieved

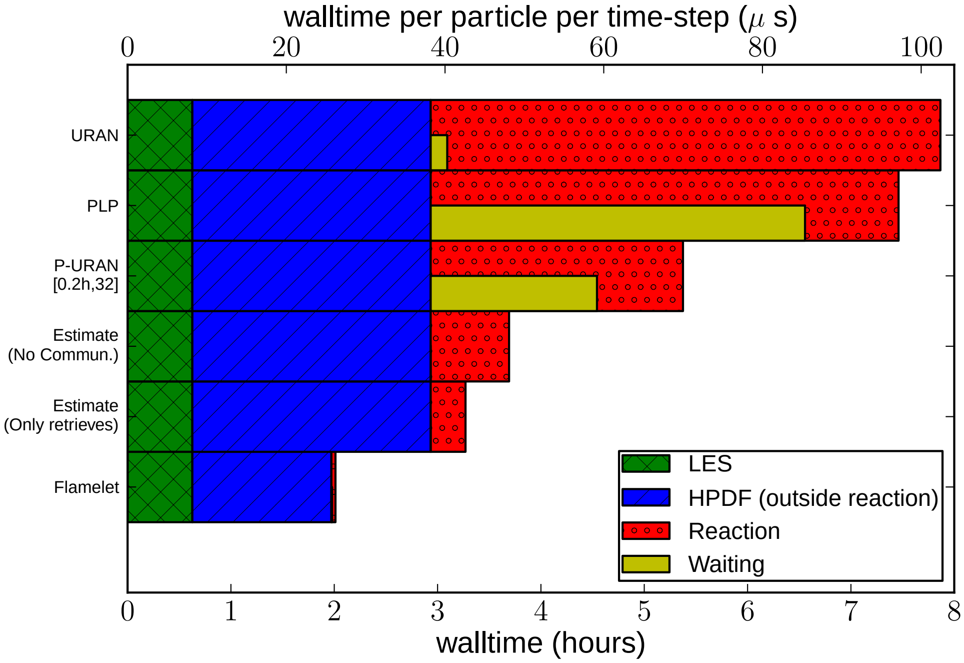


Figure 4.5: LES/PDF simulation of Sandia Flame D with the 16-species mechanism on $N_c = 1,024$ cores. Wall clock time for 2,000 time steps along with breakdown of time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) using different parallel strategies. From bottom to top: (1) Flamelet - using mixture-fraction based flamelet representation of chemistry; (2) Estimate (only retrieves) - estimate based on performing only local retrieves using pre-built ISAT tables; (3) Estimate (No Commun.) - estimate based on perfect load balancing with no communication cost; (4) P-URAN[0.2h, 32]; (5) PLP and (6) URAN.

using ISAT/*x2f_mpi* if (i) all the cores have pre-built ISAT tables, and all the particles can be resolved by retrieving reaction mappings from the local tables; and (ii) the communication cost is zero, and the chemistry workload, allowing for a typical fraction of direct evaluations in addition to retrieves, is perfectly balanced among all the cores. The method used to make the best wall clock time estimates is explained in Appendix C.

A summary of relative wall clock times required for simulating the Sandia

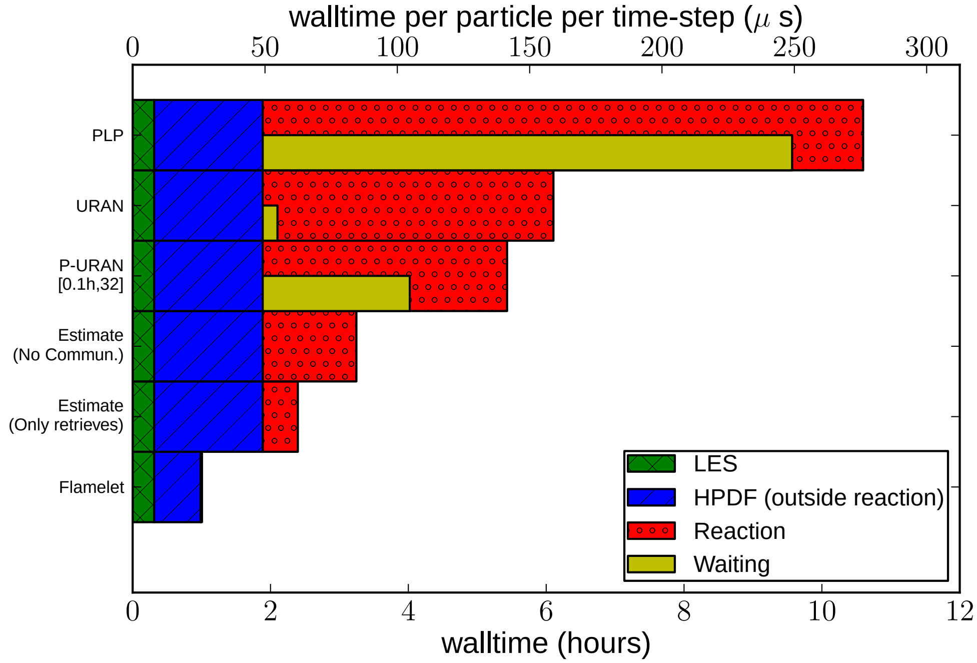


Figure 4.6: LES/PDF simulation of Sandia Flame D with the 38-species mechanism on $N_c = 1,024$ cores. Wall clock time for 1,000 time steps along with breakdown of time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) using different parallel strategies. From bottom to top: (1) Flamelet - using mixture-fraction based flamelet representation of chemistry; (2) Estimate (only retrieves) - estimate based on performing only local retrieves using pre-built ISAT tables; (3) Estimate (No Commun.) - estimate based on perfect load balancing with no communication cost; (4) P-URAN[0.1h, 32]; (5) PLP and (6) URAN.

Flame D with the chemistry represented using different methods is given in Table 4.1.

Based on these results, we can draw the following conclusions:

1. for both the mechanisms, the waiting time, which is indicative of the extent of the load imbalance, is maximum for PLP, minimum for URAN (due to near-ideal load balancing), and moderate for P-URAN (mainly due to

Table 4.1: Summary of relative overall wall clock times required for simulating the Sandia Flame D (based on results presented in Fig.4.5 and Fig.4.6) using different methods for representing chemistry.

Method	Chemistry	Wall Clock Time	
	(species, n_s)	(rel. to LES/Flamelet)	(rel. to LES/HPDF/Flamelet)
LES/Flamelet	1	1	0.3
LES/HPDF/Flamelet	1	3.2	1
LES/HPDF/PLP	16	11.9	3.7
LES/HPDF/URAN	16	12.6	3.9
LES/HPDF/P-URAN	16	8.6	2.7
LES/HPDF/PLP	38	33.9	10.6
LES/HPDF/URAN	38	19.5	6.1
LES/HPDF/P-URAN	38	17.3	5.4

load imbalance across partitions);

2. in the P-URAN strategy, for the 16-species and 38-species mechanisms respectively, more than 40% and 60% of the overall wall clock time is spent on reaction, confirming that reaction is the most expensive part of these computations;

3. the P-URAN strategy yields the lowest wall clock time for both the mechanisms: more than 25% less than PLP or URAN for the 16-species mechanism; and about 10% and 50% less than URAN and PLP, respectively for the 38-species mechanism;

4. the wall clock time with P-URAN is within a factor of 1.5 and 1.7 of the best wall clock time estimates (based on no communication) for the 16-species and 38-species mechanisms, respectively;

5. the P-URAN strategy, compared to the simple single scalar based flamelet representation, is more expensive by a factor of only 2.7 for 16-species and 5.4 for 38-species.

In short, we have shown that the P-URAN strategy performs much better than the PLP and URAN strategies, and yields wall clock time within a factor of 1.5 and 1.7 of estimates for the lowest theoretically achievable wall clock time for the 16-species and 38-species mechanisms, respectively.

Here we have compared the relative performance of the three strategies after the flame has reached a statistically-stationary state, and we find that the P-URAN strategy performs the best. A great deal of computational time can be expended reaching the statistically-stationary state, but even during these initial computations the P-URAN strategy is expected to perform the best. A relatively brief time should be spent in the PLP stage (in P-URAN) during these initial computations, because the chemistry in the domain is evolving quickly, which reduces the chances of a local retrieve.

P-URAN: Sensitivity Tests

In the previous section we considered specific strategies P-URAN[0.2h,32] (for the 16-species mechanism) and P-URAN[0.1h,32] (for the 38-species mechanism) for testing P-URAN. Here we perform sensitivity studies to see how P-URAN performs with changes in the time spent in the PLP stage (τ) and the partition size (κ).

By definition, the P-URAN strategy has the following limits in which it reduces to the PLP or URAN strategy:

- P-URAN[$\tau = \infty$,κ] = PLP
- P-URAN[τ,$\kappa = 1$] = PLP
- P-URAN[$\tau = 0$,$\kappa = N_c$] = URAN

In our tests with the P-URAN strategy, we typically use a value of τ less than 0.5 hours, and choose a partition size, κ, which is a multiple of the domain decomposition in the radial (or lateral) direction, D_y, and is preferably closer to $\sqrt{N_c}$. In the sensitivity results presented here, we show that the wall clock time with the P-URAN strategy shows very little sensitivity to changes in the values of τ and κ in the typical range of values that we might use in our computations, and consistently performs better than the PLP and URAN strategies. We have not tried to study how the P-URAN strategy approaches the aforementioned limits for extreme values of τ and κ.

First, we fix the partition size, $\kappa = 32$, and vary the time spent in the PLP stage, τ, from 0 to 5 hours, and compute the overall wall clock time for running $N_t = 2,000$ time steps for the 16-species mechanism. The wall clock time with different strategies is shown in Fig.4.7 along with breakdown of time spent in LES, HPDF (outside reaction), Reaction and Waiting in the P-URAN tests. We see that the P-URAN strategy shows very little sensitivity to changes in the time spent in the PLP stage, and the lowest wall clock time is achieved near $\tau = 0.3$h. The overall wall clock time for the simulation increases by only about 1 hour as the time spent in the PLP stage is increased from 0 to 5 hours. And in this entire range, P-URAN consistently performs better than PLP and URAN strategies. The reason for this can be understood by studying the breakdown of wall clock time for the P-URAN tests presented in Fig.4.7. We see that as more time is spent in the PLP stage, the Waiting time increases due to the initial load imbalance in

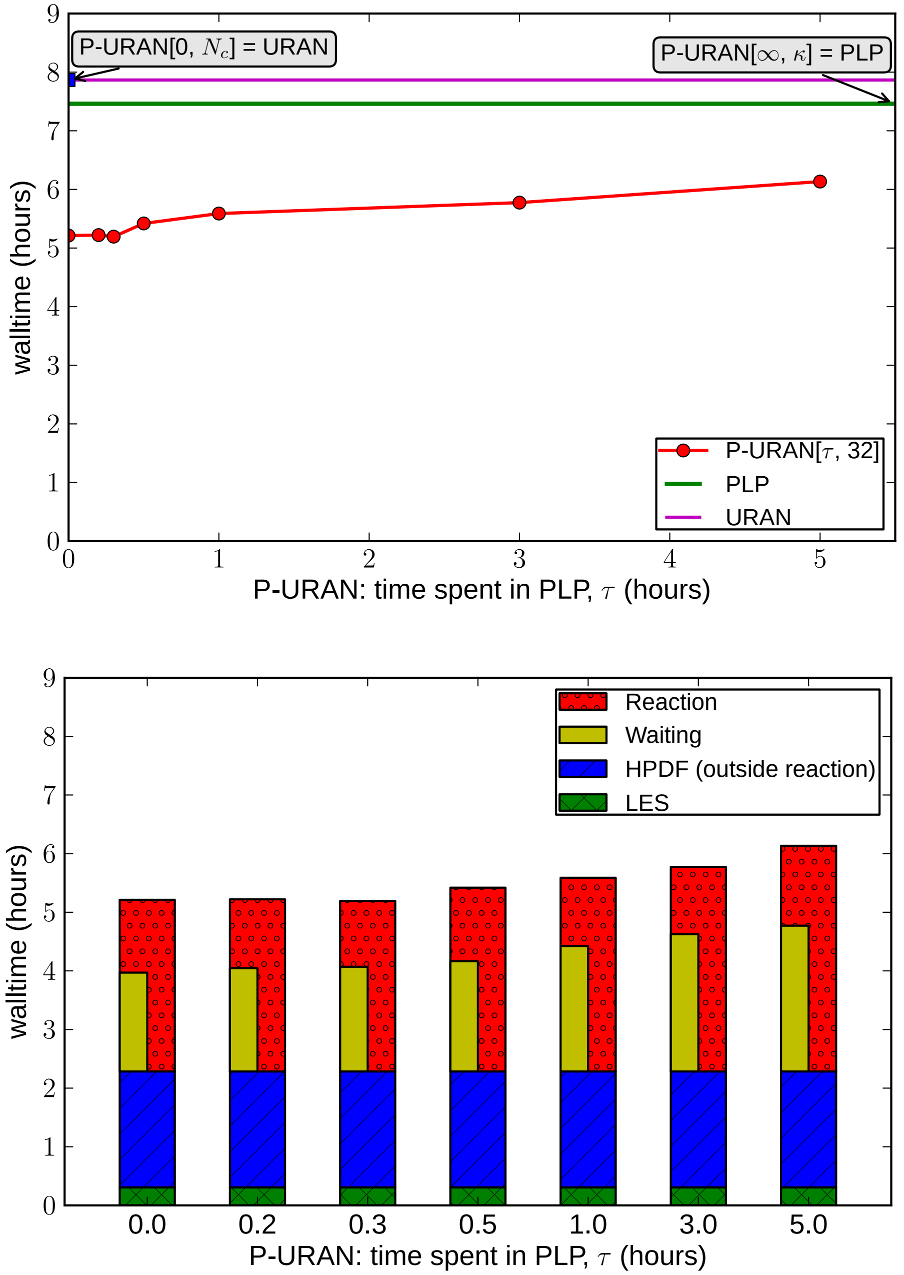


Figure 4.7: P-URAN sensitivity tests with the 16-species mechanism. Top: wall clock time for 2,000 time steps with (i) PLP; (ii) URAN; and (iii) P-URAN[τ, 32] with time τ spent in PLP varied from 0 to 5 hours. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) in the P-URAN tests.

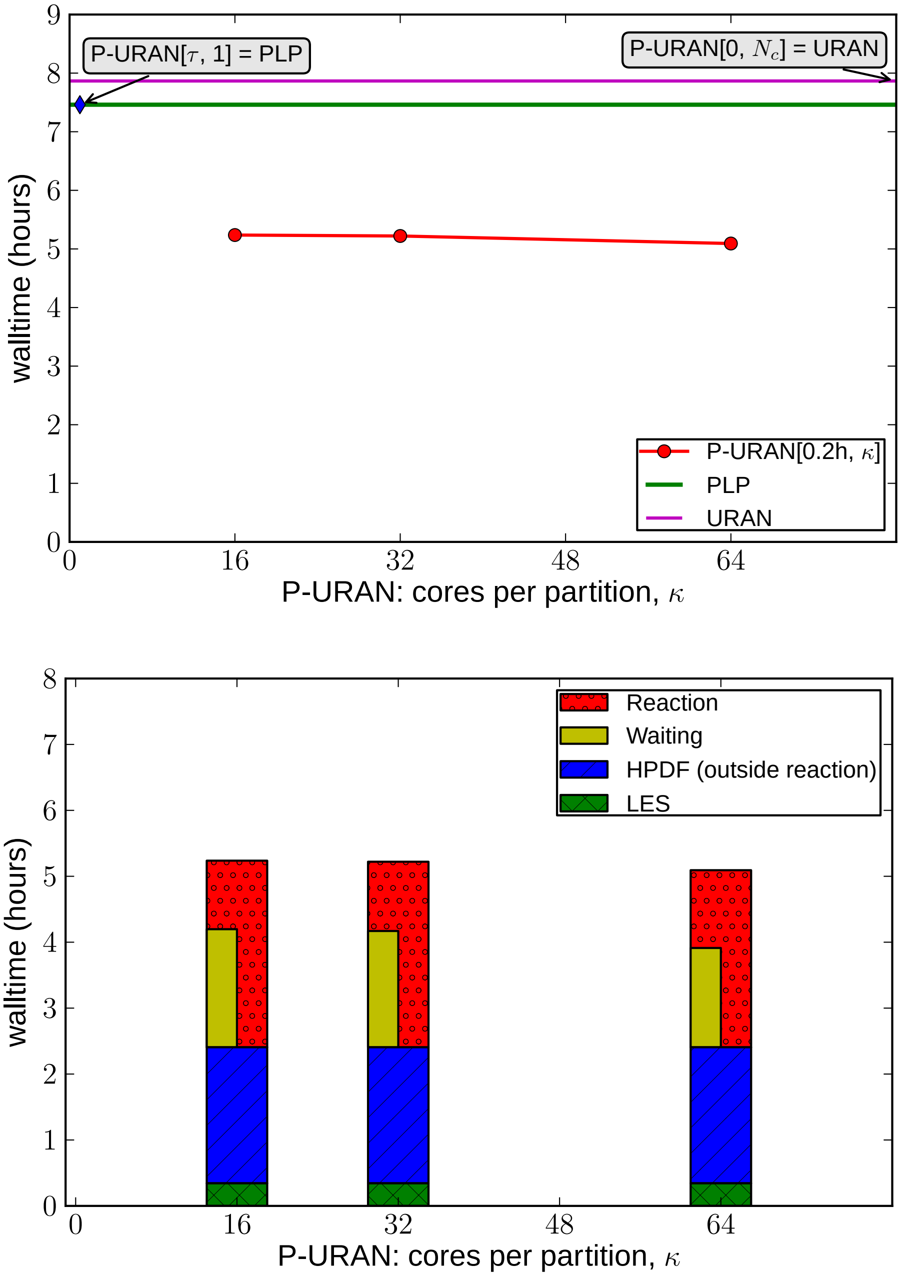


Figure 4.8: P-URAN sensitivity tests with the 16-species mechanism. Top: wall clock time for 2,000 time steps with (i) PLP; (ii) URAN; and (iii) P-URAN[0.2h, κ], with the partition size, $\kappa = 16, 32, 64$. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) in the P-URAN tests.

the PLP stage. However, a longer time spent in PLP in stage 1 increases the chances of retrieving from the local ISAT tables in stage 2, and so reduces the Reaction time in the stage 2 of P-URAN, thereby yielding approximately the same overall wall clock time for the simulation. The relatively slow approach of the P-URAN wall clock time towards the PLP time also shows that the stage 2 in P-URAN significantly reduces the overall wall clock time relative to using only PLP for the entire simulation. The P-URAN strategy with more time spent in the PLP stage is expected to show better relative performance for longer simulations due to increased chances of local retrieves. In these tests, out of the 2,000 time steps, the number of steps completed in the PLP stage with $\tau = 0.5$, 1, 3 and 5 hours is 112, 246, 977 and 1,665 respectively. In general when performing 24-hour simulations using the P-URAN strategy, the data suggests using a value of τ between 0.5 to 1 hour, which corresponds to performing about 100-200 time steps (i.e., resolving $\mathcal{O}(10^7)$ particles per core) in the PLP stage.

Next, we fix the time spent in the PLP stage, $\tau = 0.2$h and change the partition size, κ, to 16, 32 and 64. The results are shown in Fig.4.8. Here again, the P-URAN strategy shows very little sensitivity to changes in the partition size. This again is because of a balance achieved between the communication cost and load imbalance. As seen in the breakdown of the wall clock time for the P-URAN tests presented in Fig.4.8, smaller partitions reduce the communication cost, but increase the load imbalance between partitions and thus increase the Waiting time. On the other hand, bigger partitions achieve better load balance and reduce the Waiting time, but result in more communication cost and thus increase the Reaction time. For this reason, we suggest using a partition size close to $\sqrt{N_c}$ to strike a balance between the communication cost and load imbalance.

Table 4.2: Computational details of the weak-scaling tests performed for the 16-species and 38-species mechanisms using the URAN and P-URAN strategies.

			Strategies	Strategies
N_c	$D_x \times D_y$	N_{pc}	(16-species mech.)	(38-species mech.)
2,304	48×48	30	URAN, P-URAN[0.2h, 48]	URAN, P-URAN[0.1h, 48]
4,608	96×48	60	URAN, P-URAN[0.2h, 48]	URAN, P-URAN[0.1h, 48]
6,144	96×64	80	URAN, P-URAN[0.2h, 64]	URAN, P-URAN[0.1h, 64]
9,216	96×96	120	URAN, P-URAN[0.2h, 96]	URAN, P-URAN[0.1h, 96]

Sensitivity studies with the 38-species mechanism yielded similar results, and hence are not presented here for conciseness.

4.6.3 Parallel Scalability

To assess the parallel scalability of our combined LES/HPDF solver to large numbers of cores, in the next two subsections we study the *weak* and *strong* scaling of our solver.

Weak Scaling

The weak scaling study determines how the computational time varies with number of cores for a *fixed workload per core*. Accordingly, the overall problem size is increased in proportion to the number of cores to measure the scalability of the solver. Here the weak scaling tests consist of increasing the number of particles per cell (N_{pc}) in proportion to the number of cores.

We perform weak scaling tests with the URAN and P-URAN strategies. Typically we use 30-50 particles per cell (N_{pc}) in LES/PDF computations. We start the weak scaling tests from 2,304 cores with $N_{pc} = 30$ and go up to 9,216 cores with $N_{pc} = 120$. (We do not start the weak scaling tests from around 1,000 cores to avoid an unrealistically large value of N_{pc} at 9,216 cores.) The details of the tests performed for the 16-species and 38-species mechanisms are provided in Table 4.2. In each case, we perform an LES/PDF simulation for $N_t = 1,000$ time steps. The wall clock time per time step (averaged over 1,000 time steps) along with breakdown of time spent in LES, HPDF (outside reaction), Reaction and Waiting for the 16-species and 38-species mechanisms is shown in Fig.4.9 and Fig.4.10, respectively. We see that the P-URAN strategy consistently performs better than URAN on all four of the core counts for both the mechanisms, and also shows better weak-scaling up to 9,216 cores than the URAN strategy. For the 16-species mechanism, compared to results presented in Fig.4.5, in the weak scaling results (Fig.4.9) the P-URAN strategy does not perform significantly better than URAN because these are relatively smaller runs (1,000 time steps) and we expect the wall clock time with P-URAN to improve for longer runs due to increased probability of local retrieves and reduced communication cost. In Fig.4.10, we notice that the Reaction time with the P-URAN strategy is not monotonic, and the wall clock time slightly reduces when moving from 4,608 to 6,144 cores. This could simply be due to load variations on the compute cluster. As mentioned in Appendix B, we typically observe a 5% variation in the computed wall clock times on repeated runs of our solver.

In order to quantify the weak scaling behavior more accurately, we define the

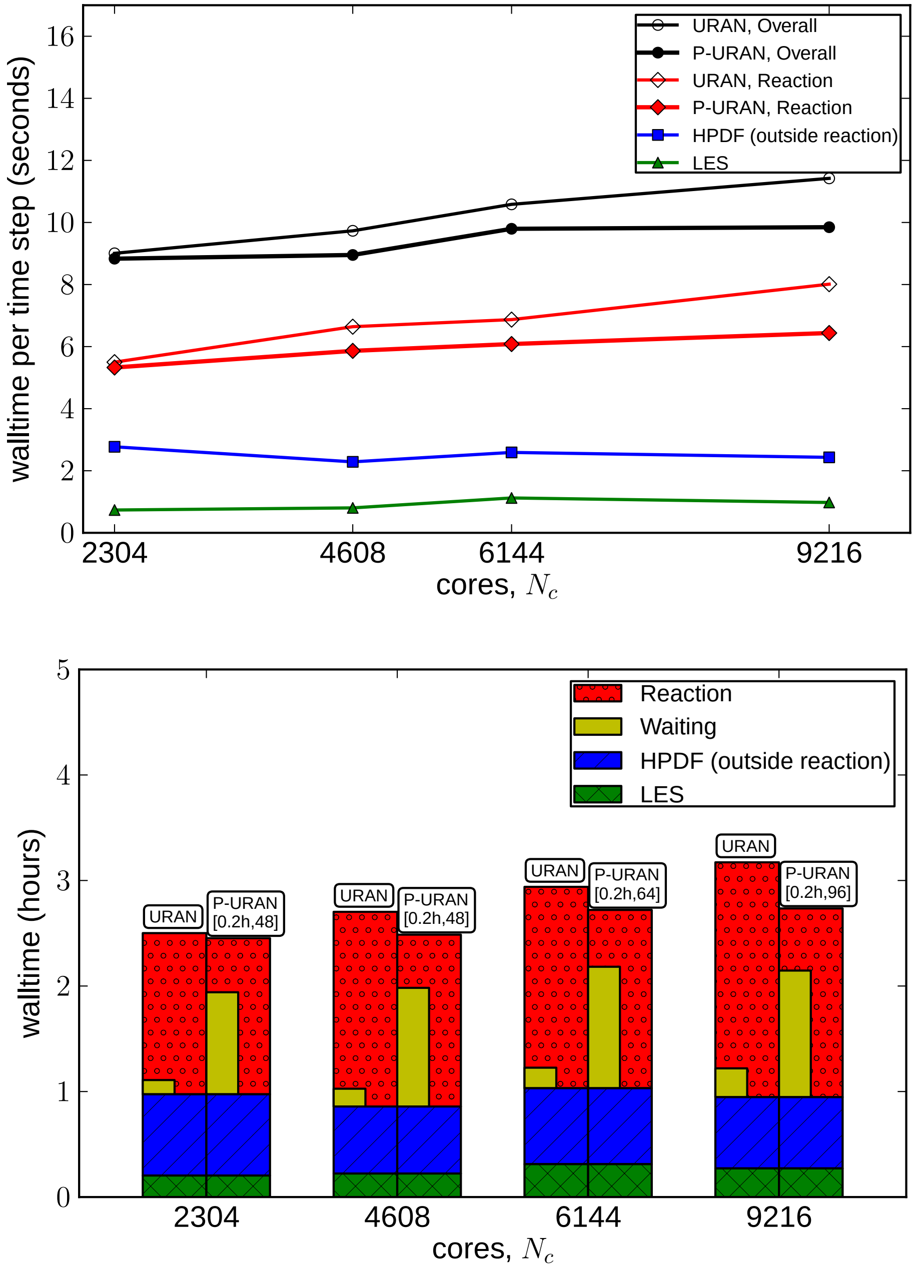


Figure 4.9: Weak scaling tests using the URAN and P-URAN strategies with the 16-species mechanism (test details provided in Table 4.2) for $N_t = 1,000$ time steps. Top: weak scaling - wall clock time per time step spent in LES, HPDF (outside reaction), Reaction and Overall. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time).

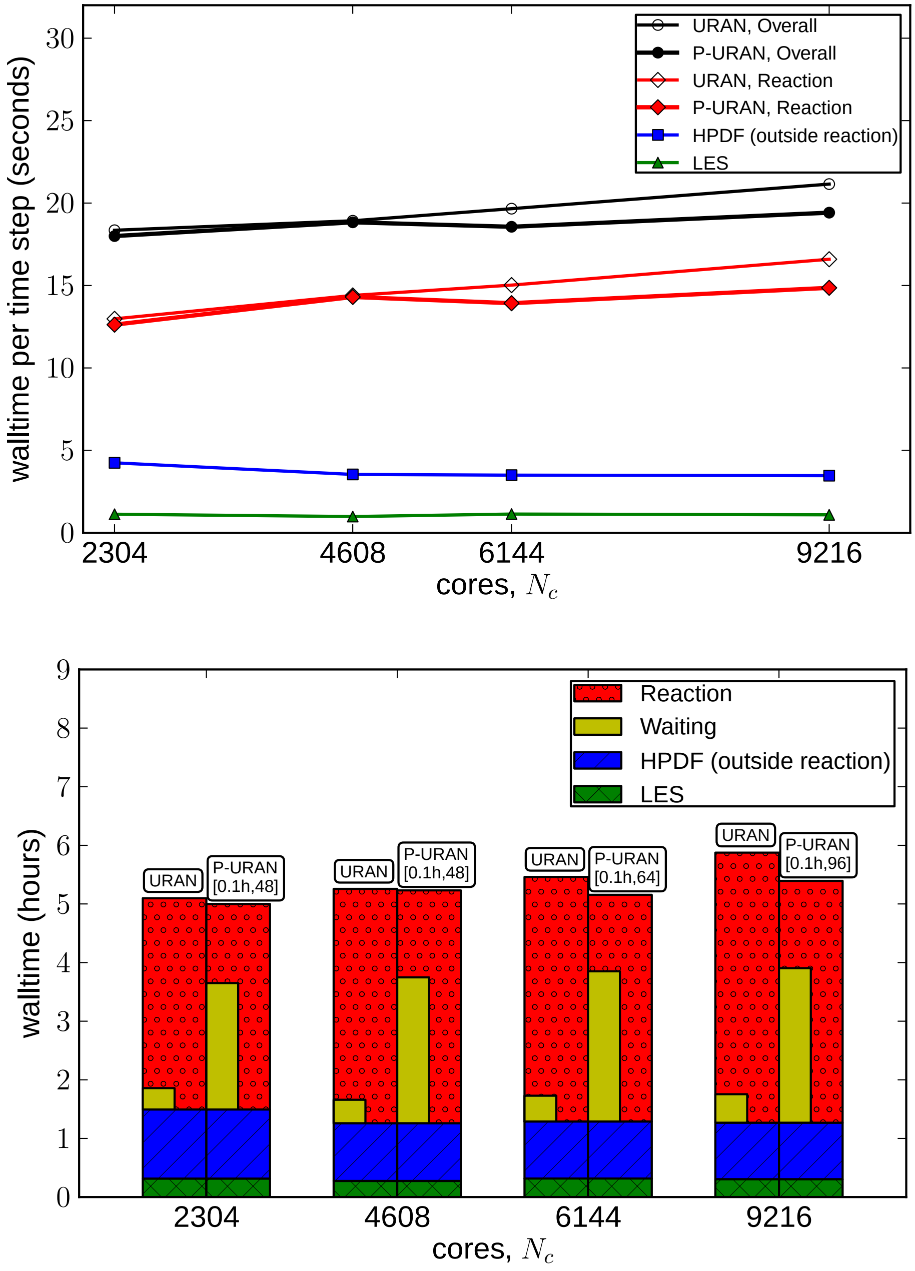


Figure 4.10: Weak scaling tests using the URAN and P-URAN strategies with the 38-species mechanism (test details provided in Table 4.2) for $N_t = 1,000$ time steps. Top: weak scaling - wall clock time per time step spent in LES, HPDF (outside reaction), Reaction and Overall. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time).

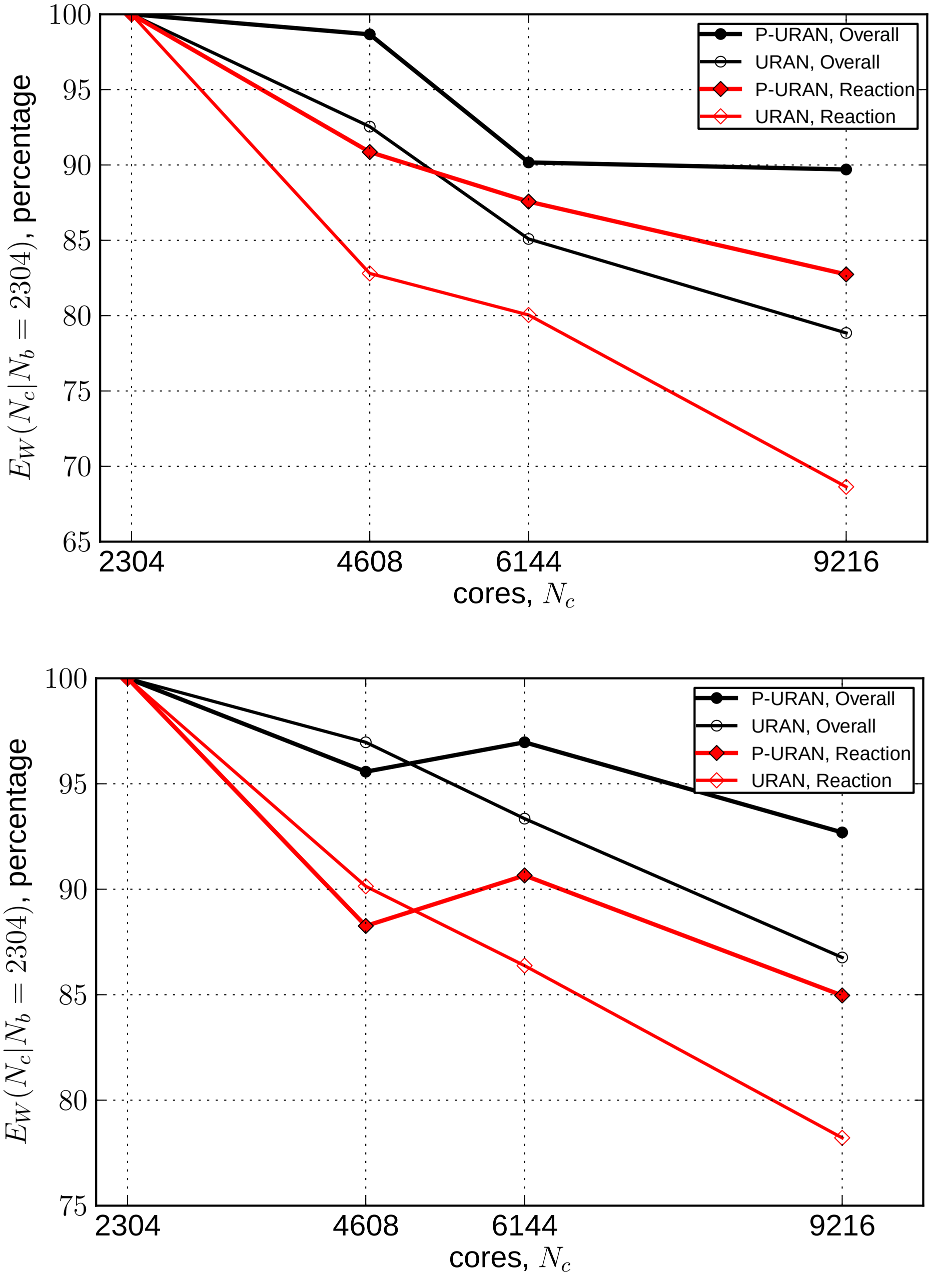


Figure 4.11: Relative weak scaling efficiency of Reaction and Overall using the URAN and P-URAN strategies. Top: with the 16-species mechanism for a simulation of $N_t = 1,000$ time steps. Bottom: with the 38-species mechanism for a simulation of $N_t = 1,000$ time steps.

weak scaling efficiency at N_c cores relative to a base case at N_b cores, denoted by $E_W(N_c|N_b)$, as

$$E_W(N_c|N_b) = \frac{T(N_b)}{T(N_c)}, \tag{4.2}$$

where $T(N_b)$ and $T(N_c)$ denote the wall clock time using N_b and N_c cores, respectively. Here we take the base case to be $N_b = 2,304$ cores. The relative weak scaling efficiency overall and of reaction for the 16-species and 38-species mechanisms using the URAN and P-URAN strategies is shown in Fig.4.11. We see that

1. at higher core counts the relative weak scaling efficiency with P-URAN is better than URAN by 5 to 10%;
2. with P-URAN the relative weak scaling efficiency of reaction at 9,216 cores is close to 85% for both the mechanisms, and over 90% overall.

Strong Scaling

The strong scaling study determines how the computational time varies with number of cores for a *fixed overall problem size*. For ideal strong scalability, the time to solution for the LES/PDF solver would decrease in inverse proportion to the number of cores employed (so-called linear speedup).

We perform strong scaling studies with just the P-URAN strategy on 1,152 to 9,216 cores. In these strong scaling tests, we use a fixed number of particles per cell, $N_{pc} = 40$, and increase the number of cores to see how different parts of the code scale. We perform $N_t = 2,000$ and $N_t = 1,000$ time steps with the 16-species and 38-species mechanisms, respectively. The computational details of the tests performed are listed in Table 4.3. Here to determine the strong scaling,

Table 4.3: Computational details of the strong-scaling tests performed for the 16-species and 38-species mechanisms using the P-URAN strategy.

			Strategies	Strategies
N_c	$D_x \times D_y$	N_{pc}	(16-species mech.)	(38-species mech.)
1,152	96×12	40	P-URAN[0.2h, 36]	P-URAN[0.1h, 36]
2,304	96×24	40	P-URAN[0.2h, 48]	P-URAN[0.1h, 48]
4,608	96×48	40	P-URAN[0.2h, 48]	P-URAN[0.1h, 48]
6,144	96×64	40	P-URAN[0.2h, 64]	P-URAN[0.1h, 64]
9,216	96×96	40	P-URAN[0.2h, 96]	P-URAN[0.1h, 96]

we estimate the wall clock time per time step for each core count using two methods:

1. by computing the average wall clock time per time step over the first $N_t \times N_c/9216$ time steps; this corresponds to the same average number of particles evaluated per core for each core count;
2. by computing the average wall clock time per time step over the complete N_t time steps; this corresponds to same overall workload independent of core count.

Among these, the first estimate gives a more accurate measure of strong scaling for the reaction due to approximately same number of particles evaluated per core. The wall clock time per time step using the above two estimates (along with breakdown of time spent in LES, HPDF (outside reaction), Reaction and Waiting) for the 16-species and 38-species mechanisms is shown in Fig.4.12 and Fig.4.13, respectively.

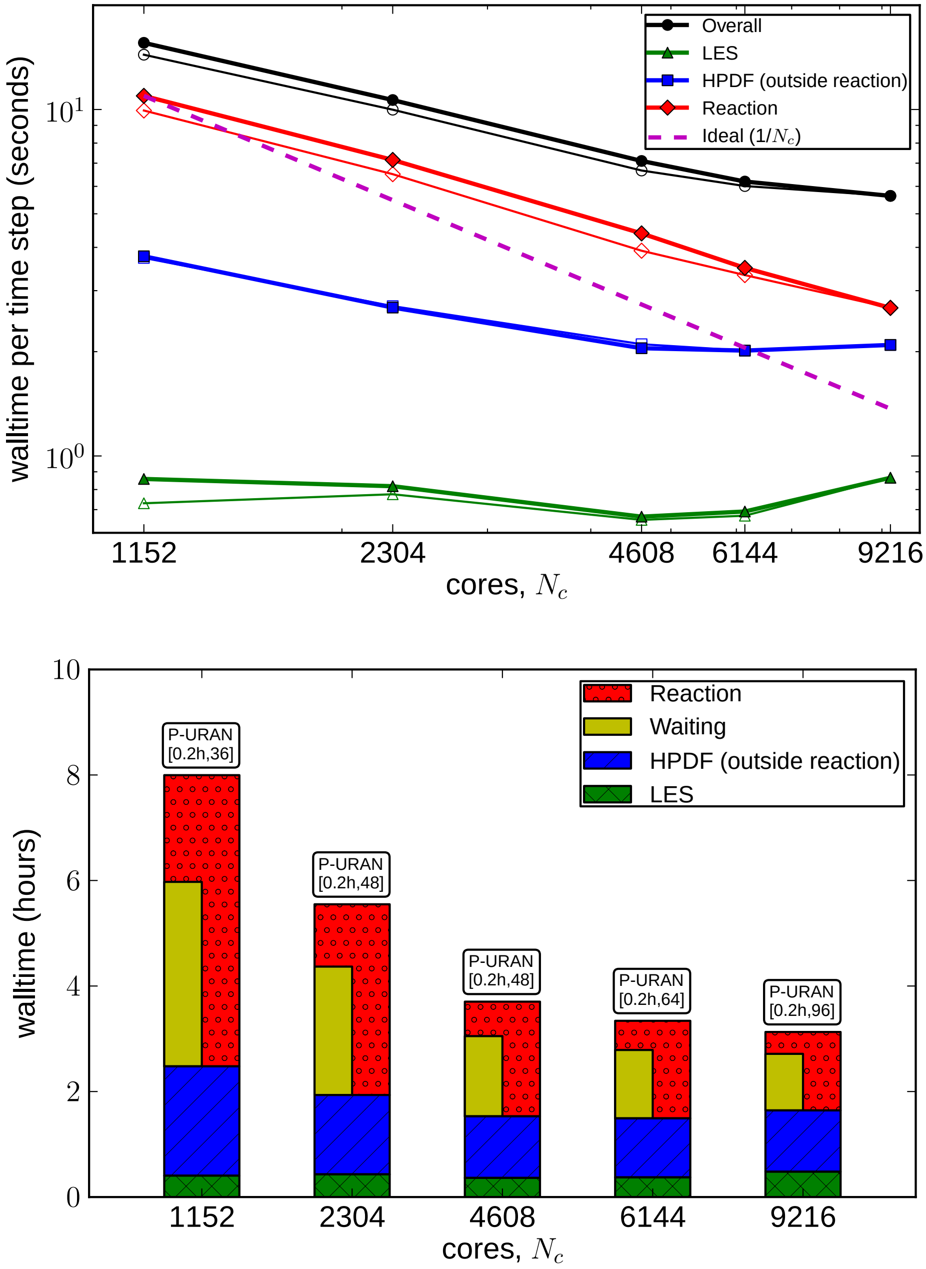


Figure 4.12: Strong scaling tests using the P-URAN strategy with the 16-species mechanism (test details provided in Table 4.3) for $N_t = 2,000$ time steps. Top: strong scaling - wall clock time per time step spent in LES, HPDF (outside reaction), Reaction and Overall. solid symbols, average over $N_t \times N_c/9216$ time steps; hollow symbols, average over N_t time steps. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) for N_t time steps.

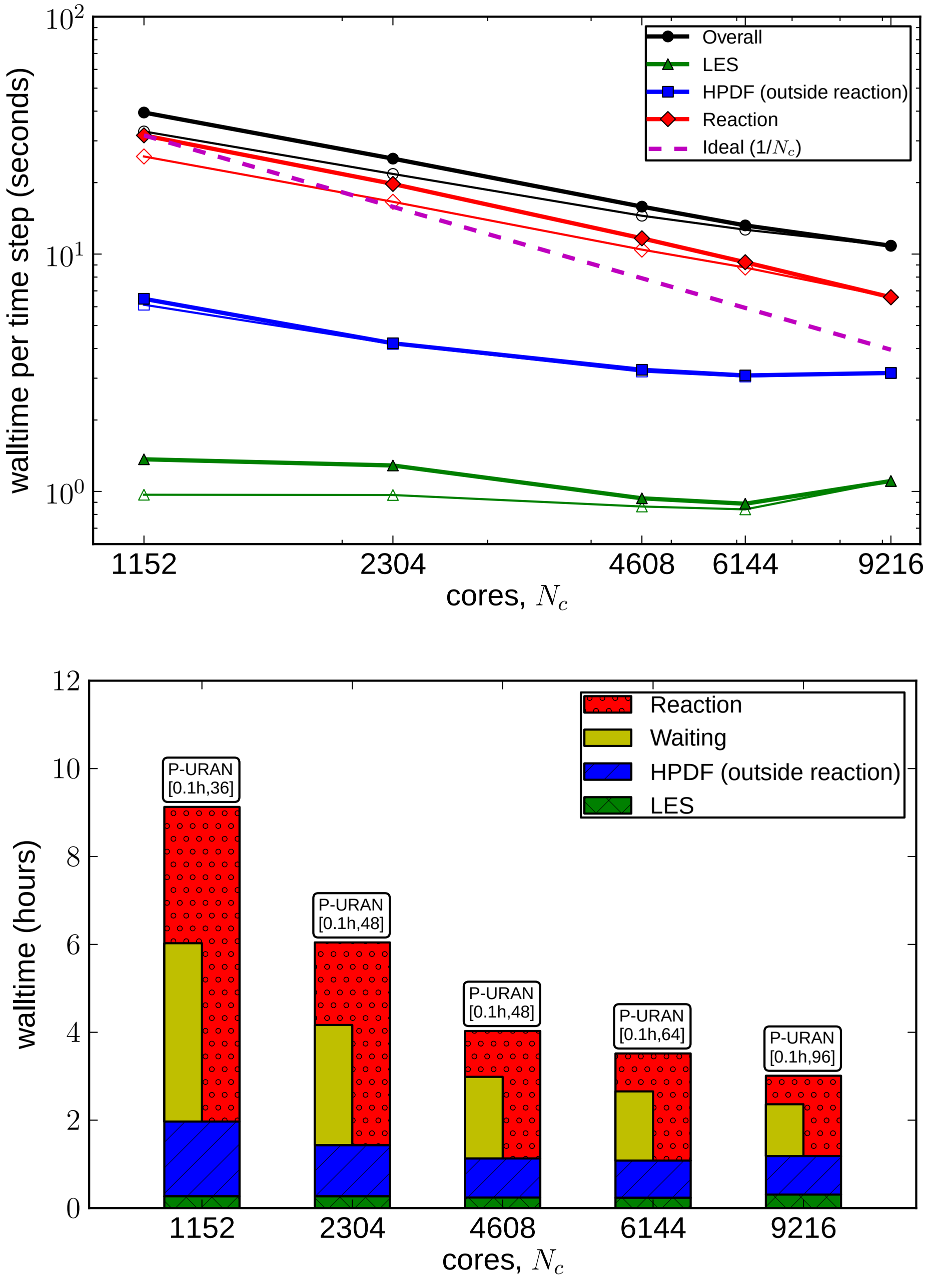


Figure 4.13: Strong scaling tests using the P-URAN strategy with the 38-species mechanism (test details provided in Table 4.3) for $N_t = 1,000$ time steps. Top: strong scaling - wall clock time per time step spent in LES, HPDF (outside reaction), Reaction and Overall. solid symbols, average over $N_t \times N_c/9216$ time steps; hollow symbols, average over N_t time steps. Bottom: breakdown of wall clock time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) for N_t time steps.

In these results, we observe the following:

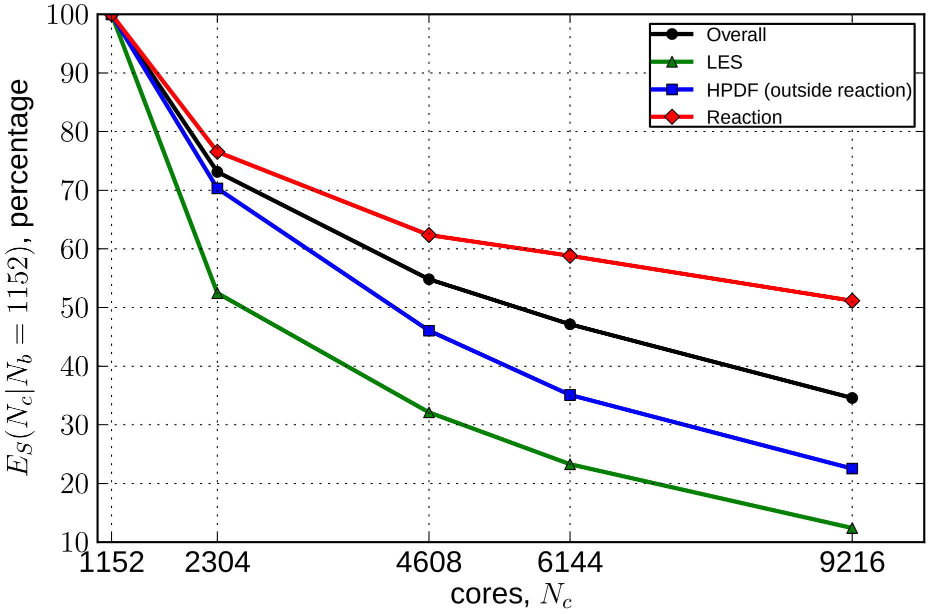
1. The LES solver does not show much, if any, parallel speedup in this range of cores. (The Stanford LES code is found not to scale well beyond 500 to 1,000 cores.) However, even on 9,216 cores, the LES time represents just 10% of the total time, and so this is not a critical issue in these tests.
2. The HPDF solver shows some parallel speedup on up to 4,608 cores, but flattens out beyond that.
3. The reaction part shows a monotonically decaying wall clock time on up to 9,216 cores for both the mechanisms. Superficially, the data appear to fit a power law, but the behavior could equally well be explained by a model such as Amdahl's Law.
4. Beyond 4,608 cores, the wall clock time of the parts of the code that do not scale well starts becoming comparable to the reaction time, and hence the overall speedup begins to deteriorate.

Similar to the relative weak scaling efficiency (Eqn.4.2), in order to quantify the strong scaling behavior more accurately, we define the strong scaling efficiency at N_c cores relative to a base case at N_b cores, denoted by $E_S(N_c|N_b)$, as

$$E_S(N_c|N_b) = \frac{N_b}{N_c} \times \left(\frac{T(N_b)}{T(N_c)} \right), \tag{4.3}$$

where in this case we take the base case to be $N_b = 1,152$ cores. We measure the relative strong scaling efficiency up to 9,216 cores, individually for LES, HPDF (outside reaction), Reaction and Overall. The relative strong scaling efficiency (with respect to the base case at $N_b = 1,152$ cores) for the 16-species and 38-species mechanisms is shown in Fig.4.14.

In these strong scaling results we notice:



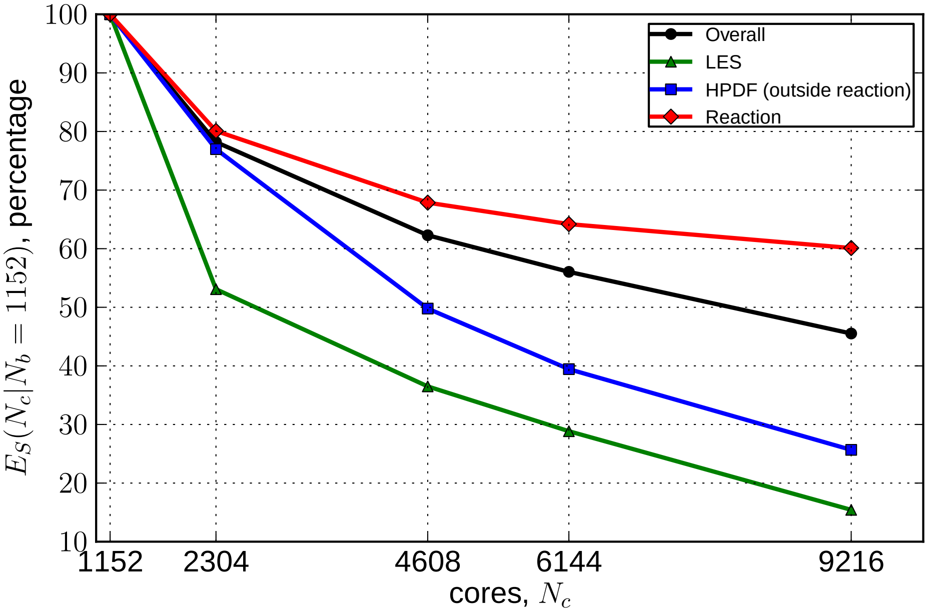


Figure 4.14: Relative strong scaling efficiency of LES, HPDF (outside reaction), Reaction and Overall using the P-URAN strategy. Top: with the 16-species mechanism for a simulation of $N_t = 2,000$ time steps. Bottom: with the 38-species mechanism for a simulation of $N_t = 1,000$ time steps.

1. at $N_c = 9,216$ cores, the relative strong scaling efficiency of reaction for the 16-species mechanism is about 50% and for the 38-species mechanism is close to 60%. However, up to $N_c = 6,144$ cores, the relative strong scaling efficiency of reaction stays above 60% for both the mechanisms;
2. the LES and HPDF parts show poor scaling, with respective relative strong scaling efficiencies dropping below 30% and 40% beyond $N_c = 6,144$ cores for both the mechanisms;
3. the overall relative strong scaling efficiency drops below 50% and 60% beyond $N_c = 6,144$ cores for the 16-species and 38-species mechanisms, respectively.

In short, these results show that the overall scaling still needs significant improvement, especially in the LES and HPDF parts. However, the reaction part alone with the P-URAN strategy shows acceptable strong scaling up to 6,144 cores.

Below we list a couple of possible reasons for not being able to achieve better strong scaling for reaction with the current implementation of the P-URAN strategy, and some ideas for improvement:

1. One possible reason for not achieving perfect scaling could be the increase in the partition size (κ) used in the P-URAN strategy with the increase in cores, N_c. As the partition size increases, the communication cost increases, thereby worsening the scaling. Under the current partitioning scheme (described in Fig.4.2), we choose the partition size, κ, to be an exact multiple of the domain decomposition in the radial direction, D_y, to achieve good load balance. So, at larger number of cores like $N_c =$

9,216 with a domain decomposition of 96×96, we are forced to use a minimum partition size of $\kappa = 96$. For example, under the current partitioning scheme, if we use $\kappa = D_y/2 = 48$ for the 96×96 domain decomposition, then this leads to a significant load imbalance between the partitions involving the first 48 cores (handling the chemically reactive sub-domains), and those involving the last 48 cores (handling the coflow/air sub-domains) in the radial direction. One possible way to avoid this load imbalance is to use $\kappa = 48$, but group every alternating ranked core in the radial direction into one partition, i.e. for every D_y cores in the radial direction, group all the even ranked cores in one partition, and the remaining odd ranked cores in another partition. This way we achieve good load balancing with smaller partition size and reduced communication cost. This new partitioning scheme can similarly be extended to any partition size $\kappa = D_y/m$, where m is some positive integer, by grouping every m^{th} ranked core into one partition. However, both the current and the new suggested partitioning schemes use the *a priori* knowledge about the direction of load imbalance (which in the simulation of turbulent reacting jet flows is the radial direction) to form the partitions. Ideally, one would like to have an adaptive partitioning strategy to form the partitions "on the fly" without using any *a priori* knowledge about the computational problem being studied, and we are currently working on developing such an adaptive strategy which might help improve the scaling.

2. Another possible reason for not achieving perfect scaling could be the use of one ISAT table per core and MPI alone for communication and parallelization. As the number of cores increase, the use of one ISAT table per core leads to significant duplication of reaction mapping evaluation and

addition of similar particle compositions to ISAT tables on multiple cores. One possible way to reduce the duplication of work and data is to use one ISAT table per processor shared by all the cores (4 to 16) on that processor. In addition, hybrid MPI/OpenMP or Graphics Processing Unit (GPU) implementation can be used to achieve better scaling. It would involve enormous work for us to incorporate these ideas in our current implementation of the LES/PDF solver, but it could be considered in future applications of ISAT for other problems.

4.6.4 Generality of the Results

To conclude this section, we discuss the generality of the conclusions drawn from this work. In particular, we consider the sensitivity of our results to changes in the combustion problem being simulated and to changes in the compute cluster architecture.

In this study the results are based on the simulation of a relatively simple turbulent jet flame, the Sandia Flame D. However, we expect the conclusions drawn from this work to be valid over a wider range of combustion problems of interest. For instance, for the simulation of Sandia Flames E and F, which exhibit significantly more local extinction than Flame D and are computationally more expensive, we expect the P-URAN strategy to again yield the lowest wall clock time. Relative to Flame D, the Flames E and F are expected to exhibit greater load-imbalance between regions of flame front and coflow due to stronger turbulent chemistry interactions, and as a result the PLP strategy should perform poorly. The P-URAN strategy (with a partitioning scheme similar to the one used in this study) should perform better than the PLP and URAN strategies, especially at large numbers of cores. Even for the simulation of turbulent flames with different geometry, the P-URAN strategy with an appropriate partitioning scheme is expected to perform better than the PLP and URAN strategies as it is able to strike a good balance between the communication cost and load imbalance.

In this study, all the simulations are performed on the TACC Ranger cluster. Due to the generality of the MPI implementation in our solver, we expect the results presented here to be relatively insensitive to changes in the cluster architecture. However, we do realize that by taking into account the cluster architecture and core-connectivity we may be able to come up with a superior mapping of the LES/PDF sub-domains onto nodes or cores. In the present study, we have attempted to do this by assigning the MPI ranks in radial order, then axial (as shown in Fig.4.2). As a consequence, each block of 16 sub-domains in the radial direction is assigned to a single node (16 cores) on Ranger. Since in the P-URAN strategy the communication is restricted to sub-domains in the same partition, and partitions are set according to axial location, the current MPI rank-assignment and partitioning schemes ensure that most of the MPI communication happens intra-node. To reduce inter-node communication still further, we try to use a partition size κ which is an exact multiple of the number of cores per node, which is 16 on Ranger. We thereby minimize the relatively slow inter-node communication and take the best advantage of the faster intra-node connectivity. Similar considerations are easily extended to other clusters, given the ubiquity of architectures that feature multiple cores per node.

4.7 Conclusions

We have successfully developed an integrated LES/HPDF/x2f_mpi/ISAT solver for performing turbulent combustion calculations with realistic combustion chemistry. We have demonstrated a new parallel strategy, P-URAN, implemented using the *x2f_mpi* library for performing chemistry calculations efficiently. In particular, we have shown that for performing LES/PDF calculations, the P-URAN strategy:

- yields the lowest wall clock time among all the strategies tested;
- yields a wall clock time within a factor of 1.5 and 1.7 of estimates for the lowest theoretically achievable wall clock time for the 16-species and 38-species mechanisms, respectively;
- compared to the single scalar mixture-fraction based flamelet implementation, is more expensive by only a factor of 2.7 and 5.4 for the 16-species and 38-species mechanisms, respectively;
- achieves a relative weak scaling efficiency for reaction of about 85% when scaling from 2,304 to 9,216 cores; and
- achieves a relative strong scaling efficiency for reaction of over 60% when scaling from 1,152 to 6,144 cores.

4.8 Acknowledgements

V.H.'s work on the methane ARM1 mechanism is supported by Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division of the US Department of Energy (DOE) under Grant DE-FG02-90ER. The work of V.H. and S.R.L. on the ethylene mechanism is supported by Grant Number FA9550-09-1-0611 funded by the National Center for Hypersonic Combined Cycle Propulsion, sponsored by the AFOSR and NASA ARMD. S.R.L.'s initial work on this project was supported by NASA grant NNX08AB36A. This research was also supported in part by the National Science Foundation through TeraGrid resources provided by the Texas Advanced Computing Center under Grant No. TG-CTS090020. S.B.P. has a financial interest in Ithaca Combustion Enterprise, LLC., which has licensed the software ISAT-CK used in this work.

CHAPTER 5
LARGE-SCALE PARALLEL SIMULATIONS OF TURBULENT COMBUSTION USING COMBINED DIMENSION REDUCTION AND TABULATION OF CHEMISTRY[†]

5.1 Abstract

Simulations of turbulent reacting flows with chemistry represented using detailed kinetic model involving a large number of species and reactions are computationally expensive. Here we present a combined dimension reduction and tabulation strategy for implementing chemistry in large scale parallel *Large-Eddy Simulation* (LES)/*Probability Density Function* (PDF) computations of turbulent reacting flows. In this approach, the dimension reduction is performed using the *Rate Controlled Constrained-Equilibrium* (RCCE) method, and tabulation of the reduced space is performed using the *In Situ Adaptive Tabulation* (ISAT) algorithm. In addition, we use *x2f_mpi* – a Fortran library for parallel vector-valued function evaluation (used with ISAT in this context) – to efficiently redistribute the chemistry workload among the participating cores in parallel LES/PDF computations to reduce the overall wall clock time of the simulation. We test three parallel strategies for redistributing the chemistry workload, namely (a) PLP, purely local processing; (b) URAN, the uniform random distribution of chemistry computations among all cores following an early stage of PLP; and (c) P-URAN, a Partitioned URAN strategy that redistributes the workload within partitions or subsets of the cores. To demonstrate

[†] 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. Proceedings of the Combustion Institute, 34, 2013 (in press). DOI:10.1016/j.proci.2012.06.004

the efficiency of this combined approach, we perform parallel LES/PDF computations (on 1,024 cores) of the Sandia Flame D with chemistry represented using a 38-species C_1-C_4 skeletal mechanism. We show that relative to using ISAT alone with the 38-species full representation, the combined ISAT/RCCE approach with 10 represented species (i) predicts time-averaged mean and standard deviation statistics with a normalized root-mean-square difference of less than 3% (30 K) in temperature, less than 2% (0.02 kg/m^3) in density, less than 2.5% in mass fraction of major species, and less than 8% in mass fraction of minor species of interest; and (ii) reduces the simulation wall clock time by over 40% with the P-URAN strategy.

5.2 Introduction

Detailed chemical mechanisms of hydrocarbon fuels may involve hundreds or thousands of species and thousands of reactions [100, 82]. Incorporating directly such detailed chemistry in the combustion flow calculations is computationally prohibitive, even using distributed parallel computing. The current efforts aimed at reducing the computational cost of representing chemistry can be placed under three main categories: (1) *mechanism reduction*, the generation of reaction mechanisms involving fewer species and reactions [53, 60, 58]; (2) *dimension reduction*, the representation of chemistry using a reduced number of variables [41, 44, 55, 79]; and (3) *tabulation*, the use of storage-retrieval methods to reduce significantly the cost of expensive evaluations of the reaction mappings involving ODE integrations [66, 49, 92, 93]. Combined methodologies have also been developed to use reduced reaction mechanisms or dimension reduction methods in conjunction with tabulation [32, 76, 89].

Since most of the modern day reactive flow simulations are performed in parallel on multiple cores using distributed computing, in addition to the aforementioned techniques, strategies are needed to efficiently distribute the chemistry workload among the participating cores to reduce the overall wall clock time of the simulation [51, 50, 48, 86]. We recently demonstrated scalable parallel strategies implemented using *x2f_mpi* for the efficient redistribution of chemistry workload in large scale parallel Large-Eddy Simulation (LES)/Probability Density Function (PDF) computations [29].

In this paper we further extend our LES/PDF solver with the capability of representing chemistry using our combined dimension reduction and tabulation approach [32]. In this approach, the dimension reduction is performed using the Rate Controlled Constrained-Equilibrium (RCCE) [41, 40, 38] method followed by tabulation using the *In Situ* Adaptive Tabulation (ISAT) [66, 49] algorithm. In [32], we tested our combined dimension reduction and tabulation approach using the partially-stirred reactor for methane and ethylene chemistry, and the main conclusions drawn from this work are that the ISAT/RCCE approach

- yields the same level of accuracy as other reduced (based on the Quasi-Steady State Assumption, QSSA) or skeletal mechanisms with relatively fewer represented species;
- yields significant speedup relative to using ISAT alone with the detailed mechanism.

Here, for the first time, the ISAT/RCCE approach is being demonstrated in the context of full-scale LES/PDF simulations of turbulent reacting flows using realistic chemistry. In this study, the accuracy and efficiency of this combined approach is demonstrated by performing full-scale LES/PDF simulation of the Sandia Flame D [9].

The outline of the remainder of the paper is as follows: in Section 5.3 we describe our combined LES/PDF solver; in Section 5.4 we describe the combined dimension reduction and tabulation strategy; in Section 5.5 we briefly describe the parallel strategies implemented using *x2f_mpi* for redistributing the chemistry workload in large scale LES/PDF computations; in Section 5.6 we present computational details for simulating the Sandia Flame D; in Section 5.7 we present simulation results; and finally in Section 5.8 we state our conclusions.

5.3 Combined LES/PDF Solver

In this study we use a combined LES/PDF solver developed at Cornell as described in more detail in [29, 97]. Below we mention some of the key aspects of this solver.

5.3.1 LES Solver

The LES solver is based on a Stanford LES code [62, 63]. The solver uses structured non-uniform grids; supports cylindrical coordinate system; is second-order accurate in space and time; and is parallelized (using MPI) by domain decomposition in two dimensions.

5.3.2 PDF Solver

The PDF solver, *HPDF* [97], has second-order accuracy in space and time; supports Cartesian and cylindrical coordinate systems; is parallelized (using MPI) by domain decomposition in two dimensions; and has a general interface to facilitate coupling with existing LES (or RANS) solvers.

In the PDF approach, the thermochemical composition of the fluid within the solution domain is represented by a large number of particles. The HPDF solver has three main components

1. *transport*: to account for the change in *position* of a particle due to advection in the physical space (including a random-walk component to represent the effects of subgrid-scale turbulent advection and molecular diffusion);
2. *mixing*: to account for the change in *composition* of a particle due to mixing with neighboring particles (which models the effects of molecular mixing); and
3. *reaction*: to account for the change in *composition* of a particle due to chemical reaction.

These components are implemented in *fractional steps* using splitting schemes [98]. In this study, to simulate the Sandia Flame D, we use the first-order $\mathbb{TMR}$ splitting scheme (which is found to perform as well as the second-order splitting scheme for jet flames [97]). The $\mathbb{TMR}$ splitting scheme denotes taking fractional steps of *transport*, $\mathbb{T}$; *mixing*, $\mathbb{M}$; and *reaction*, $\mathbb{R}$ in this order on each time-step. The Kloeden and Platen (KP) [42] stochastic differential equation (SDE) scheme is used to integrate the *transport* equations; and the *mixing* is represented using the modified Curl [37] mixing model. The *reaction* fractional step is implemented using the combined dimension reduction and tabulation approach which we will discuss in more detail in the later sections.

5.3.3 Domain Decomposition

The LES computations are performed on a structured non-uniform grid in the cylindrical coordinate system. We denote the grid used for LES computations by $N_x \times N_r \times N_\theta$ (in the axial, radial and azimuthal directions). In performing parallel LES/PDF computations (using the combined LES/HPDF solver) on N_c cores, the computational domain is decomposed into N_c sub-domains and each core performs the computations of one sub-domain. The domain decomposition is done in the first two principal directions x and r, and is denoted by $D_x \times D_r$, where $D_x D_r = N_c$. For instance, in this study we perform LES/PDF simulations of the Sandia Flame D using a non-uniform LES grid of size $N_x = 192$, $N_r = 192$, $N_\theta = 96$ on $N_c = 1,024$ cores using a domain decomposition with $D_x = 64$ and $D_r = 16$.

5.4 Combined Dimension Reduction and Tabulation

In this section we briefly describe the combined dimension reduction and tabulation approach used for representing chemistry using ISAT/RCCE. More detailed description can be found in [32].

5.4.1 Particle Representation

We consider a reacting gas-phase mixture consisting of n_s chemical species, composed of n_e elements. We consider an isobaric system with a fixed specified pressure p, and so the thermochemical state of the mixture (at a given position and time) is completely characterized by the mixture enthalpy h, and the n_s-vector $\mathbf{z}$ of specific moles of the species.

In the *reaction fractional step*, a particle's chemical composition $\mathbf{z}$ evolves (at constant enthalpy h) in time according to the following set of n_s coupled ordinary differential equations (ODEs)

$$\frac{d\mathbf{z}(t)}{dt} = \mathbf{S}(\mathbf{z}(t)), \tag{5.1}$$

where $\mathbf{S}$ is the n_s-vector of chemical production rates determined by the chemical mechanism used to represent the chemistry.

Given a reaction fractional time step Δt, the *reaction mapping*, $\mathbf{z}(\Delta t)$, is defined to be the solution to Eqn.5.1 after time Δt starting from the initial composition $\mathbf{z}(0)$. The reaction mapping obtained by directly integrating the set of ODEs given by Eqn.5.1 is referred to as a *direct evaluation* (DE). We use DDASAC [15] for performing ODE integration.

Owing to the large cost of direct evaluation of reaction mappings involving large numbers of species, we use a combined dimension reduction (using RCCE) and tabulation (using ISAT) strategy for representing chemistry. This combined methodology can be applied to chemical systems involving a large number of species (100 to 1000) by first applying the dimension reduction to reduce the dimensionality of the system to say around 20 (depending on the level of accuracy needed) and then using ISAT to tabulate the reaction mappings in the reduced dimension.

5.4.2 Dimension Reduction

In this section we briefly describe the procedure followed for dimension reduction in our implementation of the RCCE method; a more detailed description can be found in [31, 32].

In our implementation of RCCE, to perform the dimension reduction we specify a set of n_{rs} *represented* (constrained) species selected from the full set of n_s species. Consequently, we have $n_{us} = n_s - n_{rs}$ *unrepresented* species.

The selection of good represented species is crucial for the accuracy of the RCCE dimension reduction method. We have devised an automated Greedy Algorithm with Local Improvement (GALI) [31, 32] to select good represented species based on a specified measure of dimension reduction error. The greedy algorithm selects represented species in stages one-by-one which minimizes the specified measure of dimension reduction error [31].

The *reduced representation* of the species composition is denoted by $\mathbf{r} \equiv \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$, where $\mathbf{z}^r$ is an n_{rs}-vector of specific moles of the represented species; and $\mathbf{z}^{u,e}$ is an n_e-vector of specific moles of the elements in the unrepresented species (for atom conservation). Thus, $\mathbf{r}$ is a vector of length $n_r = n_{rs} + n_e$, and the dimension of the system is reduced from n_s to n_r. At any time t, the *reduced representation*, $\mathbf{r}(t)$, is related to the *full representation*, $\mathbf{z}(t)$, by

$$\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t), \tag{5.2}$$

where $\mathbf{B}$ is a constant $n_s \times n_r$ matrix which can be determined for a specified set of represented species. (In general, the reduced representation in RCCE can be a linear or non-linear function of the full representation [10].)

In the HPDF solver, with dimension reduction, the particles carry only the reduced representation $\{\mathbf{r}, h\}$. Given the reduced representation $\mathbf{r}$, the temperature T and density ρ are approximated using the method described in the Appendix of [32]. Later in the Results (Section 5.7.1) we show that this approximation method yields values of temperature and density that match closely with those obtained with the full representation $\mathbf{z}$.

Given the reduced representation at the beginning of the reaction fractional step $\mathbf{r}(0)$, and the reaction fractional time step Δt, the *reduced reaction mapping* $\mathbf{r}(\Delta t)$ (at constant enthalpy) is computed using the following steps:

1. *species reconstruction*: given $\mathbf{r}(0)$, we compute the constrained-equilibrium composition at constant enthalpy, $\mathbf{z}^{CE}(\mathbf{r}(0))$, using CEQ [69];
2. *reaction mapping*: starting from $\mathbf{z}^{CE}(\mathbf{r}(0))$, we solve the full system of n_s ODEs Eqn.5.1 to obtain the reaction mapping, $\mathbf{z}(\Delta t)$;
3. *reduction*: we obtain the *reduced reaction mapping* as, $\mathbf{r}(\Delta t) = \mathbf{B}^T\mathbf{z}(\Delta t)$.

The above steps of course make the computation of the reaction mapping even more expensive than directly solving the full set of ODEs Eqn.5.1, due to the additional species reconstruction and reduction steps. However, when ISAT is used in conjunction with dimension reduction, the computational cost is reduced significantly as explained in the next section.

A more efficient way of obtaining the reduced reaction mapping, $\mathbf{r}(\Delta t)$, is to directly solve a reduced system of n_r ODEs for the constraints, $\mathbf{r}$, or for the constraint-potentials (as is done in the classical RCCE approach [35, 36]). We are currently working on implementing this method which should give a further improvement in performance. Nevertheless, even with our current implementation of RCCE, we achieve significant reduction in computational cost relative to the detailed chemistry calculation as shown in this work.

5.4.3 Tabulation

We use *in situ* adaptive tabulation (ISAT) [66] for tabulating the reaction mappings. The ISAT algorithm has been successfully applied in many combustion chemistry calculations involving up to $n_s \leq 50$ species [49, 32]. However, with chemistry involving more than 50 species, the direct use of ISAT may not be very efficient, due to the large table size and search times [32].

Hence, for chemistry involving more than say $n_s \geq 30$ species, we use the RCCE dimension reduction method to represent the chemistry using a reduced representation involving fewer n_r variables. Note, for very large mechanisms involving thousands of species, the direct use of RCCE/GALI may still result in $n_r \gg 30$ to achieve an acceptable level of accuracy. In such cases it will be more efficient to use ISAT/RCCE with a skeletal mechanism (based on the detailed mechanism) involving hundreds of species.

We use ISAT to tabulate the *reduced reaction mapping* in the reduced dimension n_r, which reduces significantly the overall computational cost because

1. the exact reduced reaction mapping is computed (using the steps listed in the previous section) only for a small fraction of particles (typically less

than 1%); and for the majority of the particles a linear approximation to the reduced reaction mapping is obtained using the tabulated data;

2. since the tabulation is performed in a reduced dimension, n_r, the ISAT table size is reduced, which in turn reduces the table search and retrieve times; and
3. since the particle compositions are also stored in a reduced dimension, fetching particles from the memory is faster.

Consequently, the combined dimension reduction and tabulation approach using ISAT/RCCE is found to give an additional speedup by a factor of $\mathcal{O}(10)$ relative to using ISAT alone with the full representation for tests performed using the 111-species C_1-C_4 USC Mech II detailed mechanism [32]. A more detailed description of our combined dimension reduction and tabulation approach is provided in [32].

5.5 Parallel Strategies for Implementing Chemistry

In performing parallel LES/PDF computations on multiple cores using our LES/HPDF solver with chemistry represented using the combined dimension reduction and tabulation approach, each core has its own ISAT table for tabulating the chemistry. On the reaction fractional step, the reaction mappings for all the particles in the computational domain need to be evaluated. However, the chemical reactivity is in general not uniform across the entire domain. For example, in simulation of jet flames, the sub-domains in the flame front are chemically more reactive than sub-domains in the outer coflow/air. Thus, a direct call to ISAT on each core at the reaction fractional step can create *load imbalance* among the cores, leading to increase in the overall simulation wall clock time. Hence, at the reaction fractional step, we use parallel strategies implemented using *x2f_mpi* to redistribute the particles among the participating cores for reaction mapping evaluation, thereby achieving a better load balance and reducing the overall simulation wall clock time.

In [29], we presented three parallel strategies, denoted by PLP, URAN and P-URAN, for redistributing the chemistry workload. We give a brief description of these strategies below.

1. Purely Local Processing (PLP):

   In this strategy, the reaction mapping of all the particles on a core is evaluated using the local ISAT table without any message passing or load redistribution. This in some sense is the same as invoking ISAT directly from HPDF on each core without using the *x2f_mpi* interface. This strategy thus leads to significant load imbalance.

2. Uniform Random (URAN):

   This strategy is the extreme opposite of the PLP strategy and aims at achieving statistically ideal load balancing by evenly distributing the chemistry workload among all the participating cores. The strategy involves one initial step of PLP to initialize the local ISAT tables. In the subsequent steps, on each core, first an attempt is made to retrieve the reaction mapping of particles from the local ISAT table (also referred to as a "quick try"). Following this, there is a uniform random distribution of all the unresolved particles (for which "quick try" failed) to all the cores. This strategy thus ensures that the workload is evenly balanced over all the cores, however, it also results in a large amount of all-to-all message passing.

3. Partitioned Uniform Random (P-URAN):

   This strategy aims at achieving a balance between communication cost and load imbalance by using the PLP and URAN strategies over smaller partitions of cores. The P-URAN strategy works in two stages: in stage 1, for a specified duration of time τ (hours) the PLP strategy is used to resolve particles; then in stage 2, for the remainder of the time, the participating cores are partitioned into smaller groups of κ cores, and within each partition the URAN strategy is used to uniformly distribute the chemistry workload among the cores in that partition. We use the notation P-URAN[τ, κ] to describe the P-URAN strategy.

In [29], based on LES/PDF simulations of Sandia Flame D using ISAT alone we showed that among the aforementioned three strategies, the P-URAN strategy yields the lowest wall clock time. We also showed that the P-URAN strategy shows good scaling up to 9,000 cores. In this work we use these strategies in conjunction with combined dimension reduction and tabulation to compare their relative performance. Here we focus more on the gains achieved using the combined dimension reduction and tabulation approach and show that the simulation wall clock time can be further reduced using our combined ISAT/RCCE approach without losing too much accuracy.

5.6 LES/PDF Simulation of Sandia Flame D

To test the chemistry implementation we perform LES/PDF simulations of the Sandia Flame D.

5.6.1 Sandia Flame D

The Sandia Flame D is a piloted CH_4/Air jet flame operating at a jet Reynolds number, Re = 22,400. All the details about this flame and the burner geometry can be found at [9]. Here we mention only some of the important aspects of this flame.

The jet fluid consists of 25% CH_4 and 75% air by volume. The jet flows in at 49.6 m/s velocity at 294 K temperature and 0.993 atm pressure. The jet diameter is $D = 7.2$ mm. The pilot is a lean (equivalence ratio, $\Phi = 0.77$) mixture of C_2H_2, H_2, air, CO_2, and N_2 with the same nominal enthalpy and equilibrium composition as that of CH_4/Air at this equivalence ratio. The pilot velocity is 11.4 m/s. The coflow is air flowing in at 0.9 m/s at 291 K and 0.993 atm.

5.6.2 Computational Details

We perform LES/PDF simulation of the Sandia Flame D using the coupled LES/HPDF solver. The simulation is performed in a cylindrical coordinate system. A computational domain of $80D \times 30D \times 2\pi$ is used in the axial, radial and azimuthal directions, respectively. A non-uniform structured grid of size 192×192×96 (in the axial, radial and azimuthal directions, respectively) is used for the LES solver. In the HPDF solver, the number of particles per LES cell (N_{pc}) used is $N_{pc} = 40$. With a total of 192×192×96 $\approx 3.5 \times 10^6$ LES cells, an overall 140×10^6 particles are used in the computational domain. The simulations are performed on 1,024 cores using a domain decomposition of 64×16 in the axial and radial directions, respectively. All the simulations are performed on the Texas Advanced Computing Center (TACC) Ranger cluster.

The chemistry is represented using the combined dimension reduction (using RCCE) and tabulation (using ISAT) approach with a C_1-C_4 skeletal mechanism [23] involving $n_s = 38$ species composed of $n_e = 5$ elements. This mechanism is developed especially for ethylene combustion, but is also applicable to methane flames. In the future, we want to apply the methodology developed here to study ethylene combustion.

The RCCE dimension reduction is performed by specifying $n_{rs} = 10$ represented species (which is found to be a good number of represented species to achieve less than 2% dimension reduction error based on our previous tests with chemical mechanisms involving around 30 species [31, 32]), and so the reduced representation has a dimension $n_r = n_{rs} + n_e = 15$. This dimension reduction from $n_s = 38$ to $n_r = 15$ results in a 60% reduction in the storage needed for particle composition and an 84% reduction in the storage per ISAT table entry. In this preliminary study, we specify the represented species manually (to include the major species of interest for which statistics had already been collected in some of our previous LES/PDF simulations and for which experimental data is available). However, in future studies with bigger mechanisms we will use GALI [32] to select the represented species. In this work, we use the following 10 species as the represented species: CH_4, O_2, CO_2, H_2O, CO, H_2, OH, H, O

and HO_2.

A fixed ISAT error tolerance, $\epsilon_{tol} = 10^{-4}$, is used in this study. At this error tolerance, the ISAT *tabulation error* relative to direct evaluation (as defined in [32]) is found to be less than 3%. In addition, we specify a maximum ISAT table size of 1000 MB per core. In simulations with the 38-species full representation, some ISAT tables become completely filled. However in simulations with the combined ISAT/RCCE approach with 10 represented species, none of the ISAT tables have a size of more than 200 MB.

5.7 Results

In this section we compare the computational time and statistics of thermochemical quantities obtained using the combined dimension reduction and tabulation approach with 10 represented species relative to using tabulation alone with the 38-species C_1-C_4 skeletal mechanism.

In order to make the comparisons, we perform separate LES/PDF simulations of the Sandia Flame D on 1,024 cores with chemistry represented using the following two methods

1. ISAT: tabulation alone (no dimension reduction) with the 38-species full representation; and
2. ISAT/RCCE: combined dimension reduction and tabulation with a reduced representation involving 10 represented species (specified in the previous section) and 5 elements.

In each case, we perform LES/PDF simulation to reach a statistically stationary state. We then collect statistics for thermochemical quantities like temperature, density, and species mass fractions time-averaged over 10,000 time steps. In addition, in each case we perform simulations for 2,000 time steps using the three parallel strategies PLP, URAN and P-URAN to compare their relative performance. These simulations start from the statistically stationary state with empty ISAT tables.

5.7.1 Comparison of Statistics

In this section we compare the radial profiles of mean and standard deviation statistics of thermochemical quantities obtained from the PDF particle data from the LES/PDF simulation using ISAT alone and using the combined ISAT/RCCE approach.

The radial statistics are azimuthally-averaged at each time step, and are also time-averaged over 10,000 time steps after reaching the statistically stationary state. For a quantity ξ, we denote the density-weighted mean statistics by $\langle \xi \rangle$, and the standard deviation by $\langle \xi'' \rangle$ which is defined as $\langle \xi'' \rangle \equiv \sqrt{\langle \xi^2 \rangle - \langle \xi \rangle^2}$.

In Fig.5.1 and Fig.5.2, we show respectively the radial profiles of mean and standard deviation of temperature T, density ρ, and mass fraction of species CH_4, O_2, CO_2, H_2O, CO, H_2, OH at axial locations $x/D = 15,\ 30,\ 45,\ 60$ obtained from (i) an LES/PDF simulation using ISAT alone with the 38-species full representation; (ii) an LES/PDF simulation using ISAT/RCCE with 10 represented species; and (iii) experimentally measured statistics [9] (for reference).

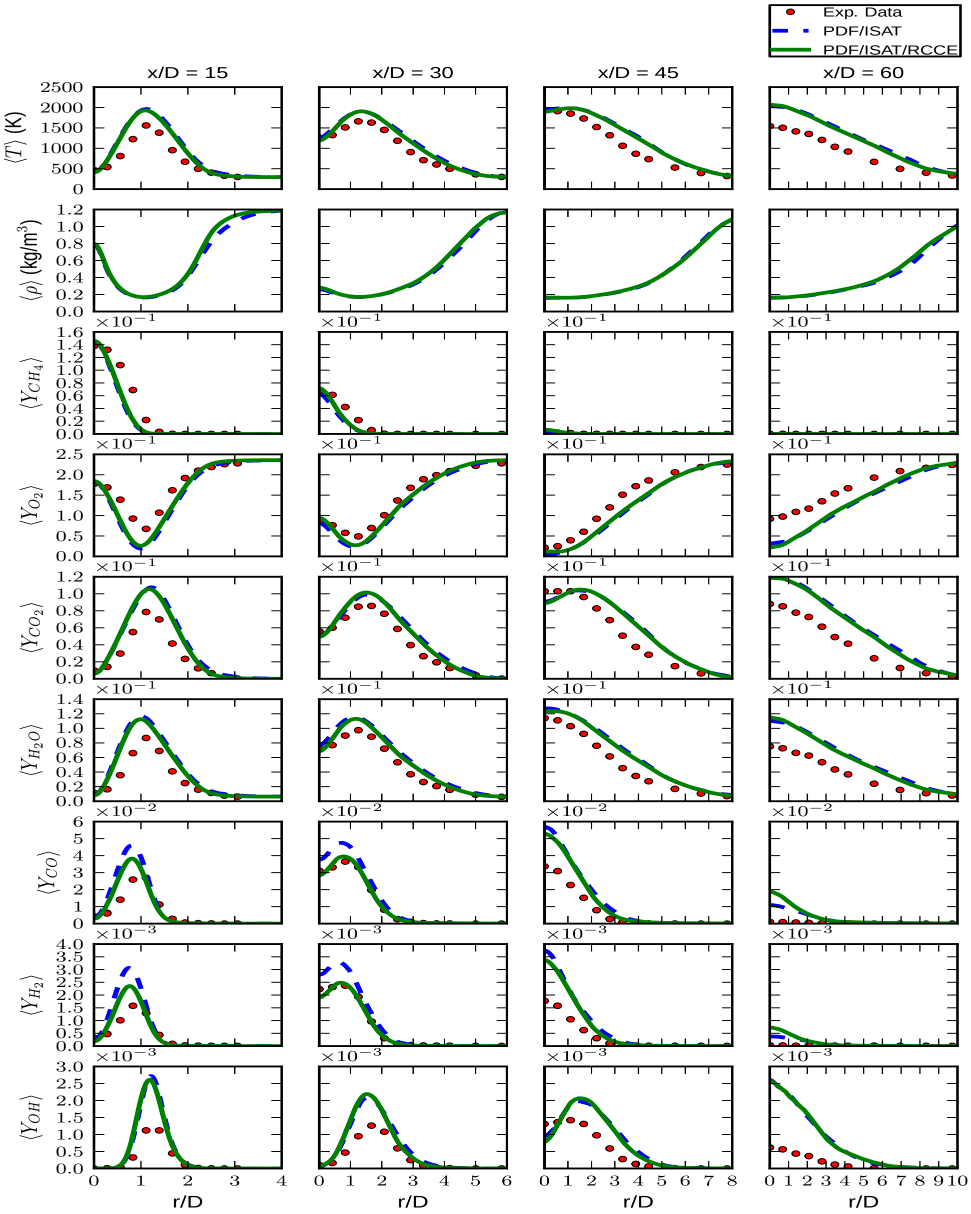


Figure 5.1: Radial profiles of time-averaged mean temperature T, density ρ, and mass fraction of species CH_4, O_2, CO_2, H_2O, CO, H_2, OH at axial locations $x/D = 15,\ 30,\ 45$ and 60 obtained from (i) experimental data; (ii) an LES/PDF simulation using ISAT alone with the 38-species full representation; and (iii) an LES/PDF simulation using ISAT/RCCE with 10 represented species.

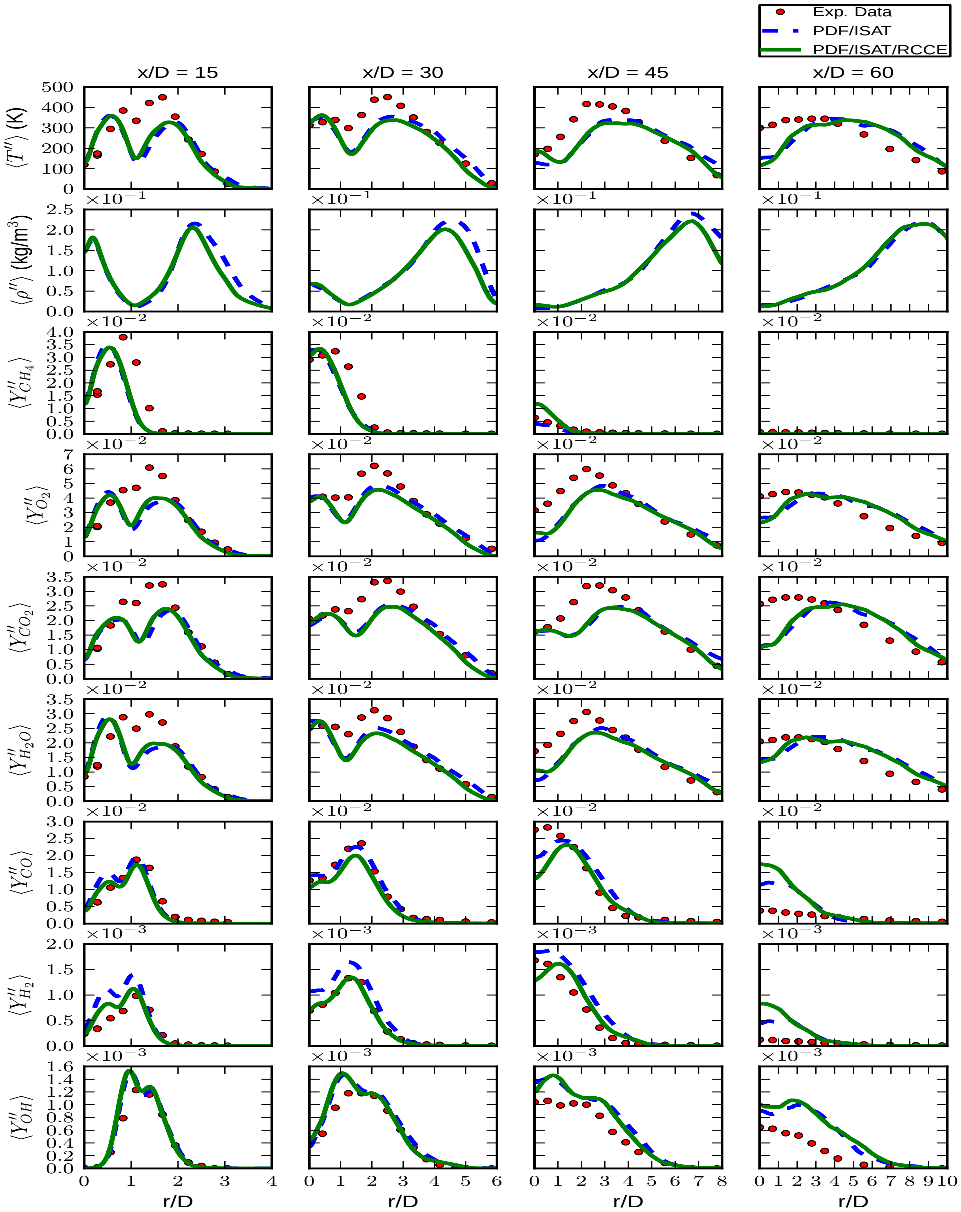


Figure 5.2: Radial profiles of time-averaged standard deviation of temperature T, density ρ, and mass fraction of species CH_4, O_2, CO_2, H_2O, CO, H_2, OH at axial locations $x/D = 15,\ 30,\ 45$ and 60 obtained from (i) experimental data; (ii) an LES/PDF simulation using ISAT alone with the 38-species full representation; and (iii) an LES/PDF simulation using ISAT/RCCE with 10 represented species.

We notice that the statistics obtained with ISAT/RCCE using 10 represented species match very closely with the statistics obtained using ISAT alone with the 38-species full representation. To quantify the difference between the statistics obtained using ISAT/RCCE and ISAT alone, for each quantity ξ (mean or standard deviation), we compute the normalized root-mean-square difference (RMSD) denoted by $\epsilon(\xi)$ as follows

$$\epsilon(\xi) = \frac{\left[\xi^r - \xi^f\right]_{rms}}{\xi_{ref}}, \tag{5.3}$$

where ξ^r and ξ^f denote respectively the quantities obtained using the reduced representation with ISAT/RCCE and the full representation with ISAT alone; and the operator $[\,]_{rms}$ denotes the RMSD computed over all the radial locations at all the considered axial locations $x/D = 15,\ 30,\ 45,\ 60$. Here ξ_{ref} is a reference value used for normalization, which is taken to be 1000 K for temperature and 1 kg/m^3 for density. For the species mass fractions, we take ξ_{ref} to be the maximum value of the mean statistics obtained for that species, $\max(\langle\xi\rangle^f)$.

The reference value and the normalized RMSD computed using Eqn.5.3 for all the quantities of interest is summarized in Table 5.1. We notice that the normalized RMSD in the mean and standard deviation statistics is less than 3% (i.e. 30 K) for temperature; less than 2% (i.e. 0.02 kg/m^3) for density; less than 2.5% for species mass fractions of major species CH_4, O_2, CO_2, H_2O; and less than 8% for species mass fractions of minor species CO, H_2, OH. In summary, these results show that the combined ISAT/RCCE approach shows good error control and the predicted statistics are well within acceptable level of accuracy (relative to using ISAT alone with the full representation) for most engineering applications. These results also show that the density and temperature approximation method used with the reduced representation in ISAT/RCCE [32] yields values that match closely with those obtained with the full representation. A more care-

Table 5.1: Normalized root-mean-square difference (RMSD) (see Eqn.5.3) in mean and standard deviation statistics obtained using the reduced representation with ISAT/RCCE relative to full representation with ISAT alone. The quantities listed are temperature T, density ρ, and mass fraction of species CH_4, O_2, CO_2, H_2O, CO, H_2, OH.

Quantity	Reference Value	RMSD in $\langle\xi\rangle$	RMSD in $\langle\xi''\rangle$
ξ	ξ_{ref}	$\epsilon(\langle\xi\rangle)$, percent	$\epsilon(\langle\xi''\rangle)$, percent
T (K)	1000.0	2.92	1.95
ρ (kg/m^3)	1.0	1.72	1.07
Y_{CH_4}	1.5×10^{-1}	1.47	1.17
Y_{O_2}	2.4×10^{-1}	2.10	0.93
Y_{CO_2}	1.2×10^{-1}	1.50	0.81
Y_{H_2O}	1.3×10^{-1}	2.24	0.92
Y_{CO}	5.7×10^{-2}	6.36	4.10
Y_{H_2}	3.7×10^{-3}	7.87	5.29
Y_{OH}	2.7×10^{-3}	2.18	1.92

ful selection of represented species using GALI [32] should help further reduce the differences between the reduced and full descriptions.

The experimentally measured statistics are qualitatively well captured by the LES/PDF simulation, yet quantitatively we notice that some of the statistics differ by around 20%. The discrepancies between the LES/PDF simulation statistics and experimentally measured statistics can be attributed to (i) numerical and statistical errors in the simulation; (ii) experimental measurement errors; and (iii) errors in the chemical kinetic models. However, study of these errors is not the primary focus of this work. Similar level of agreement between the simulated and experimentally measured statistics is found in some of the previous studies of Sandia Flame D [85, 33, 57, 59].

5.7.2 Computational Performance

In this section we compare the wall clock time required to perform LES/PDF simulation of Sandia Flame D for 2,000 time steps using the combined ISAT/RCCE approach relative to using ISAT alone. In addition we compare the relative performance of the PLP, URAN and P-URAN parallel strategies. In each case, the LES/PDF simulation is started from a fixed statistically stationary state with empty ISAT tables. We measure a moderate ISAT *build time* (see [32]) of about 1 hour in these simulations, i.e., after 1 hour of simulation, most of the particles are resolved by ISAT retrieves.

In Fig.5.3, the bottom three bars show the wall clock time taken to perform 2,000 simulation time steps using the combined ISAT/RCCE approach with 10 represented species with the PLP, URAN and P-URAN[0.2h,32] parallel strategies. In each case, we also show the breakdown of time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication), and Waiting (average idle time) as defined in [29]. We see that the P-URAN strategy yields the lowest wall clock time among the three strategies. The Waiting time (average idle time), which is indicative of the *load imbalance* is maximum for PLP, minimum for URAN and moderate for P-URAN (as also seen in our previous studies [29]).

The LES/PDF simulation of Sandia Flame D using ISAT alone with the 38-species full representation is performed using the PLP, URAN and P-URAN

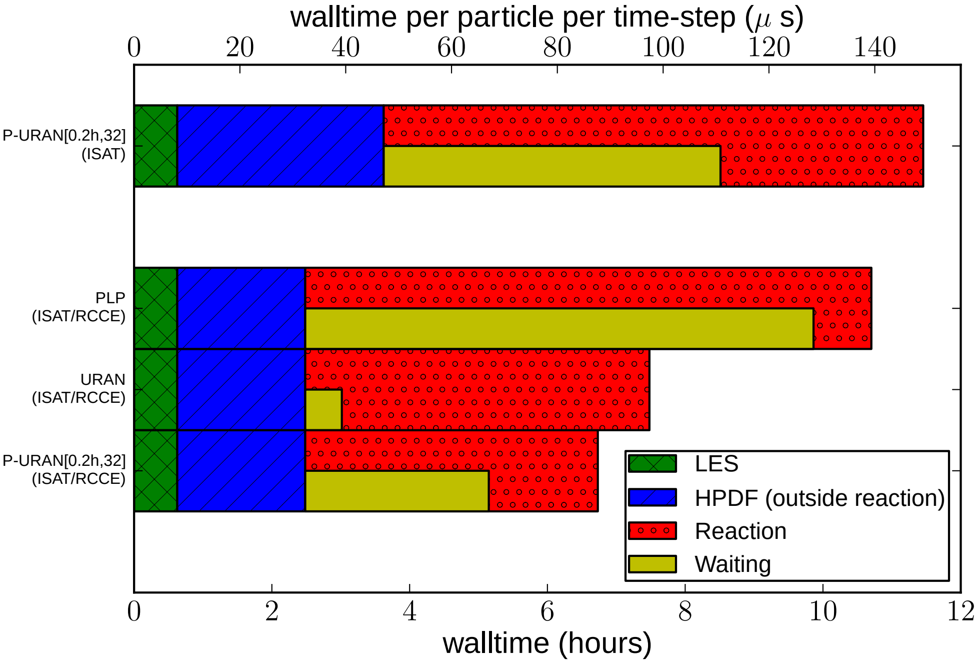


Figure 5.3: For LES/PDF simulation of Sandia Flame D, wall clock time for 2,000 time steps along with breakdown of time spent in LES, HPDF (outside reaction), Reaction (including *x2f_mpi* communication) and Waiting (average idle time) using different parallel strategies. Top: using ISAT alone with the 38-species full representation with the P-URAN[0.2h, 32] parallel strategy. Bottom three: using combined ISAT/RCCE with 10 represented species using (i) PLP; (ii) URAN; and (iii) P-URAN[0.2h, 32] parallel strategies.

strategies [29]. Among these the P-URAN strategy again yields the lowest wall clock time. In Fig.5.3, for comparison, the top bar shows the wall clock time for 2,000 time steps using the 38-species full representation with the P-URAN[0.2h,32] strategy.

We notice that relative to the simulation using ISAT alone with the 38-species full representation, the combined ISAT/RCCE approach with 10 represented species using the P-URAN strategy

Table 5.2: Cumulative ISAT statistics for the LES/PDF simulation of Sandia Flame D using (i) ISAT alone and (ii) ISAT/RCCE with the P-URAN parallel strategy.

method	variables	queries	retrieves (percent)	adds (percent)	direct evals$^{(*)}$ (percent)	retrieve time (μs)
ISAT	38	3×10^{11}	99.974	5.572×10^{-3}	1.665×10^{-3}	8
ISAT/RCCE	15	3×10^{11}	99.984	2.742×10^{-3}	0.0	3

$^{(*)}$ performed only if the ISAT table is completely filled.

1. yields more than 40% reduction in HPDF time (outside Reaction). This is because with dimension reduction, the particles in PDF simulation carry only the reduced representation (in this case involving 15 variables). As a result, a) particles require 60% less storage, which in turn reduces the particle communication cost; and b) less time is required for collecting species (LES cell mean) statistics.
2. reduces the Reaction time by over 40% due to smaller ISAT table sizes and faster retrieve times (statistics given in Table 5.2 and explained below); and
3. consequently, reduces the overall wall clock time of the simulation by more than 40%.

In Table 5.2, we list the ISAT statistics collected from the simulations with the P-URAN strategy using a) ISAT alone with the 38-species full representation; and b) ISAT/RCCE with 10 represented species. We see that in both the cases, over 99.9% of the queries result in retrieves which shows the high efficiency of ISAT tabulation. Compared to ISAT/RCCE with 10 represented species, the simulation with 38-species full-representation results in almost twice the number of adds, and also results in some direct evaluations because some of the tables get completely filled. The average retrieve time with ISAT/RCCE is only 3 μs compared to 8 μs with ISAT alone.

5.8 Conclusions

We have successfully extended our LES/PDF solver with the capability of performing turbulent combustion calculations with realistic combustion chemistry, wherein the chemistry in the PDF solver is represented using the combined dimension reduction (using RCCE) and tabulation (using ISAT) approach. The chemistry workload is efficiently redistributed using the P-URAN strategy implemented using the *x2f_mpi* library. We have shown that for performing LES/PDF simulation of Sandia Flame D, relative to using ISAT alone with the 38-species full representation, the ISAT/RCCE approach with 10 represented species (i) yields acceptable level of accuracy in mean and standard deviation statistics of major thermochemical quantities of interest like temperature, density and species mass fractions; and (ii) reduces the overall simulation wall clock time by more than 40% with the P-URAN strategy.

5.9 Acknowledgements

V.H.'s work on dimension reduction methodologies is supported by Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division of the US Department of Energy (DOE) under Grant No. DE-FG02-90ER. The work of V.H. and S.R.L. on the ethylene mechanism is supported by Grant No. FA9550-09-1-0611 funded by the National Center for Hypersonic Combined Cycle Propulsion, sponsored by the AFOSR and NASA ARMD. S.R.L.'s initial work on this project was supported by NASA Grant No. NNX08AB36A. This research was also supported in part by the National Science Foundation through TeraGrid resources provided by the Texas Advanced Computing Center under Grant No. TG-CTS090020. S.B.P. has a financial interest in Ithaca Combustion Enterprise, LLC., which has licensed the software ISAT-CK and CEQ used in this work.

CHAPTER 6

A STUDY OF THE RATE-CONTROLLED CONSTRAINED-EQUILIBRIUM DIMENSION REDUCTION METHOD AND ITS DIFFERENT IMPLEMENTATIONS[†]

6.1 Abstract

The Rate-Controlled Constrained-Equilibrium (RCCE) is a thermodynamic based dimension reduction method which enables representation of chemistry involving n_s species in terms of fewer n_r constraints. Here we focus on the application of the RCCE method to Lagrangian particle Probability Density Function based computations. In these computations, at every reaction fractional step, given the initial particle composition (represented using RCCE), we need to compute the *reaction mapping*, i.e., the particle composition at the end of the time step. In this work we study three different implementations of RCCE for computing this reaction mapping, and compare their relative accuracy and efficiency. These implementations include: (1) RCCE/TIFS (Trajectory In Full Space): this involves solving a system of n_s rate-equations for all the species in the full composition space to obtain the reaction mapping. The other two implementations obtain the reaction mapping by solving a reduced system of n_r rate-equations obtained by *projecting* the n_s rate-equations for species evaluated in the full space onto the constrained subspace. These implementations include (2) RCCE: this is the classical implementation of RCCE which uses a direct projection of the rate-equations for species onto the constrained sub-

[†] V. Hiremath and S. B. Pope. A Study of the Rate-Controlled Constrained-Equilibrium Dimension Reduction Method and its Different Implementations, Combustion Theory and Modelling, 2012, (in press). DOI:10.1080/13647830.2012.752109

space; and (3) RCCE/RAMP (Reaction-mixing Attracting Manifold Projector): this is a new implementation introduced here which uses an alternative projector obtained using the RAMP approach. We test these three implementations of RCCE for methane/air premixed combustion in the partially-stirred reactor with chemistry represented using the n_s=31 species GRI-Mech 1.2 mechanism with n_r=13 to 19 constraints. We show that: a) the classical RCCE implementation involves an inaccurate projector which yields large errors (over 50%) in the reaction mapping; b) both RCCE/RAMP and RCCE/TIFS approaches yield significantly lower errors (less than 2%); and c) overall the RCCE/TIFS approach is the most accurate, efficient (by orders of magnitude) and robust implementation.

6.2 Introduction

Recent advances on the experimental and theoretical fronts of the study of real fuel chemistry have led to more accurate chemical mechanisms of real fuels involving hundreds to thousands of species and thousands of reactions [82]. A major challenge in the numerical study of turbulent combustion problems is the accurate and efficient use of this detailed chemistry information in computations.

In a reacting flow computation, the species composition evolves by three processes: advection, diffusion and chemical reaction. Here we focus on the general class of solution methods in which a splitting scheme is used to account for these processes in separate fractional steps. In particular we focus on the turbulent combustion modeling using the Probability Density Function (PDF) methods [65] in which the chemical composition in the computational domain is represented by a large number of particles. The particle composition evolves due to mixing and chemical reaction which are treated in two separate fractional steps. The main advantage of using PDF methods is that the chemical source term in the species evolution equation is represented exactly, which enables the use of detailed chemistry in combustion calculations. The PDF methods are typically used in conjunction with Reynolds-averaged Navier-Stokes (RANS) or Large-Eddy Simulation (LES) based approaches to perform turbulent combustion simulations [56, 74, 97, 29].

A turbulent combustion simulation using PDF methods with detailed chemistry (without any simplification) entails solving (at each reaction fractional step for each particle) a coupled set of ordinary differential equations (ODEs) for the chemical species composition. These systems of ODEs are generally *stiff* owing to the wide range of chemical time-scales present in the system, and thus computing the solution is expensive. Due to this expensive cost involved in representing chemistry in turbulent combustion simulations, incorporating detailed chemistry involving thousands of species is computationally prohibitive.

The current challenges of representing chemistry in turbulent combustion simulations and the various approaches used are highlighted in [71]. In the past two decades, numerous methods have been developed to tackle the expensive cost involved in representing combustion chemistry. These methods can be broadly classified into the following three categories:

1. Mechanism Reduction: this includes methods designed to generate smaller skeletal mechanisms from the detailed mechanism by systematically removing unimportant species. Two prominent methods in this cat-

egory are the Directed Relations Graph (DRG) [53]; and the DRG with error propagation (DRGEP) [60].

2. Dimension Reduction: this includes methods used to represent chemistry using fewer "represented" variables based on the detailed chemistry. Methods in this category include the Quasi Steady-State Assumption (QSSA) [11, 87]; Rate-Controlled Constrained-Equilibrium (RCCE) [41, 40]; Computational Singular Perturbation (CSP) [44]; Intrinsic Low-Dimensional Manifolds (ILDM) [55]; Trajectory-Generated Low-Dimensional Manifolds (TGLDM) [72]; Invariant Constrained Equilibrium-Edge Pre-Image Curve (ICE-PIC) [79].

3. Tabulation: this includes storage-and-retrieval based methods, such as the In *Situ* Adaptive Tabulation (ISAT) [66, 49]; Piecewise Reusable Implementation of Solution Mapping (PRISM) [92]; Artificial Neural Network (ANN) [18].

The aforementioned methods have been successfully applied in various combustion chemistry calculations, and they have enabled the use of detailed chemistry information in computations with acceptable level of accuracy and efficiency.

In our research, we have focused on developing combined methodologies [91, 89, 77, 32] which enables us to extract the best out of the aforementioned three categories, thereby further reducing the cost of chemistry computations. In particular, our recent efforts have been focused on developing a combined reduction-tabulation strategy [32, 30] which involves dimension reduction of chemistry using the RCCE method followed by tabulation using ISAT. This combined ISAT/RCCE approach can be used with both detailed and skeletal mechanisms. We have also developed an automated Greedy Algorithm with Local Improvement (GALI) [32] to select "good" represented species for performing dimension reduction with the RCCE method.

We have extensively tested this combined ISAT/RCCE/GALI methodology for methane and ethylene chemistry with chemical mechanisms involving 20 to 100 species in a Partially-Stirred Reactor (PaSR) [31, 32] and also in full-scale Large-Eddy Simulation (LES)/Probability Density Function (PDF) computations of Sandia Flame D [30]. The main conclusions drawn from these tests are that the ISAT/RCCE/GALI approach: (a) yields the same level of accuracy as other reduced (using QSSA) and skeletal mechanisms with relatively fewer represented species; and (b) results in speed-up by a factor of 2 to 15 relative to using ISAT alone [32, 29].

In the RCCE dimension reduction method, the chemistry involving n_s-species is represented in terms of fewer n_r-constraints. The reduced representation of chemistry using the RCCE dimension reduction method is denoted by an n_r-vector $\mathbf{r}$. In PDF based simulations of reacting flows using the RCCE method, given the initial reduced composition of a particle at the beginning of a reaction time step, $\mathbf{r}(0)$, the task is to compute the *reaction mapping*, $\mathbf{r}(t)$, at the end of the reaction time step t. There are different ways of implementing the RCCE dimension reduction method to obtain this reaction mapping, and the present paper studies the relative merits of these implementations.

Our implementation of RCCE [32] is different from the classical RCCE approach first introduced in [41] and further developed and tested in [40, 27, 38, 35, 36]. The Close Parallel Inertial Manifold (CPIM) method [90] describes yet another way of implementing RCCE. The main focus of this paper is to compare these different implementations of RCCE for their relative accuracy and efficiency. In particular we look at the following three implementations of the RCCE:

1. RCCE/TIFS (Trajectory In Full Space): this is the implementation used in our previous work [31, 32, 30] (described in Sec.6.4.1);
2. RCCE: this is the classical implementation introduced in [41] and further developed in [40, 27, 38, 35, 36] (described in Sec.6.4.2);
3. RCCE/RAMP (Reaction-mixing Attracting Manifold Projector): this is a new implementation (based on the CPIM [90] method) proposed here (described in Sec.6.6).

To compute the reaction mapping, the RCCE/TIFS implementation solves a system of n_s ODEs in the full composition space for all the species. In contrast, the RCCE and RCCE/RAMP implementations solve a reduced system of n_r ODEs for the constraints by *projecting* the full system of ODEs onto the constrained subspace. In this work we show that for small reaction time steps, all the three aforementioned implementations yield similar levels of error. However, as the reaction time step increases, the RCCE/RAMP and RCCE/TIFS implementations yield orders of magnitude smaller error than the RCCE implementation. We show that the projector used in the RCCE implementation is inaccurate which results in large errors in the reaction mapping at large reaction time steps. We show that the RAMP approach provides a more accurate projector and significantly reduces the error. We also show that the RCCE/TIFS implementation is the most accurate, efficient and robust among the above three implementations.

The outline of the remainder of the paper is as follows: in Sec.6.3 we describe the general framework and notation used for representing chemistry in our work; in Sec.6.4 we give an overview of the RCCE dimension reduction method and describe the RCCE/TIFS and the classical RCCE implementations; in Sec.6.5 we describe the projection issues involved in the implementation of RCCE; in Sec.6.5.2 we describe the Close Parallel Inertial Manifold (CPIM) method [90] and then in Sec.6.6 we extend it to the new Reaction-mixing Attracting Manifold Projector (RAMP) method; in Sec.6.7 we describe the partially-stirred reactor used for testing the different implementations of RCCE; and finally in Sec.6.8 we describe results to compare the relative accuracy and efficiency of the three implementations; and in Sec.6.9 we state our conclusions.

6.3 Chemistry Representation

We consider a gaseous phase reacting flow consisting of n_s chemical species composed of n_e elements. The thermochemical state of the fluid (at a given position and time) is fully characterized by the pressure p, enthalpy h and an n_s-vector of species specific moles $\mathbf{z}$. (It is often convenient to view $\mathbf{z}$ as a point in the n_s-dimensional composition space.)

For definiteness, we focus on the combined Large-Eddy Simulation (LES)/Probability Density Function (PDF) simulations of turbulent reacting flows, which has been our recent focus of attention [97, 29, 30]. In these simulations the thermochemical composition of the fluid is represented by a large number of particles in the computational domain. The particle chemical composition evolves due to *mixing* and *reaction*, which are treated in separate *fractional* *steps* [98].

Furthermore, here to focus solely on the reaction fractional step and study different implementations of the RCCE method, we use the computationally cheaper representative test case of a Partially-Stirred Reactor (PaSR), which will be described in more detail in Sec.6.7. The methodology described here though is applicable to other time-stepping based reacting flow simulations as well.

To simplify the exposition, here we consider an adiabatic and isobaric reaction fractional step, i.e., the enthalpy h and pressure p of a particle remain constant over the reaction fractional step. In addition, to simplify things further, we consider an isobaric flow so that pressure p is the same for all the particles. Hence the thermochemical state is fully characterized by $\mathbf{z}$ and h. (Note that these assumptions are made only to simplify the exposition and can be easily relaxed if needed. In general pressure p can vary from particle to particle, and changes in particle enthalpy h can be incorporated in other fractional steps in the flow solver.)

As a consequence of the aforementioned assumptions, in the reaction fractional step, the chemical composition of each of the particles in the computational domain evolves (at constant h and p) by the following set of rate-equations

$$\frac{d\mathbf{z}}{dt} = \mathbf{S}(\mathbf{z}), \tag{6.1}$$

where the n_s-vector $\mathbf{S}$ denotes the *chemical source term* obtained from the chemical mechanism used for representing the chemistry. More precisely, the chemical source term is given as

$$\mathbf{S}(\mathbf{z}) \equiv \mathbf{S}(\mathbf{z}, T(h, \mathbf{z}), p). \tag{6.2}$$

Since the temperature T is known in terms of h and $\mathbf{z}$, and the pressure p is assumed to be constant, we henceforth use the more concise notation $\mathbf{S}(\mathbf{z})$, where dependence on T and p is implicitly assumed.

Given the initial particle composition $\mathbf{z}(0)$ at $t = 0$, and a reaction fractional time step t, we refer to the composition at the end of the reaction fractional step $\mathbf{z}(t)$ as the *reaction mapping*. The reaction mapping obtained by directly integrating the system of ODEs given by Eqn.6.1 is referred to as the *exact solution* (since the ODE integration errors are relatively small). We use DDASAC [15] for performing ODE integration.

6.4 Rate-Controlled Constrained-Equilibrium

Here we give a brief overview of the Rate-controlled Constrained-Equilibrium (RCCE) dimension-reduction method and introduce the notation used in our implementation. A fuller description can be found in [32].

In the RCCE method, the chemistry is represented by a reduced number of n_r (with typically $n_r \ll n_s$) represented scalars or constraints. This *reduced representation* of chemistry is denoted by an n_r-vector $\mathbf{r}$.

In our implementation of RCCE, to represent the chemistry using a reduced representation we specify a set of n_{rs} represented species selected from the full set of n_s species present in the chemical mechanism. The reduced representation of chemistry is given as $\mathbf{r} = \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$, where the n_{rs}-vector $\mathbf{z}^r$ denotes the species specific moles of the represented species and the n_e-vector $\mathbf{z}^{u,e}$ denotes the specific moles of elements in the unrepresented species (for element conservation). Thus the chemistry is represented in a reduced dimension of size $n_r = n_{rs} + n_e$

instead of the full dimension n_s. At any time t, the reduced representation $\mathbf{r}$ is related to the full representation $\mathbf{z}$ as

$$\mathbf{r}(t) = \mathbf{B}^T \mathbf{z}(t), \tag{6.3}$$

where $\mathbf{B}$ is a fixed $n_s \times n_r$ matrix determined by the choice of the represented species.

It is often convenient to view the n_s-dimensional composition space to be composed of the n_r-dimensional represented subspace (spanned by the columns of the constraint matrix $\mathbf{B}$) and its orthogonal complement n_u-dimensional (with $n_u = n_s - n_r$) *unrepresented* subspace, such that together the represented and unrepresented subspaces span the entire composition space. This helps visualize the n_s-dimensional composition space in a 2D sketch (for example see Fig.6.1) indicated by the represented (denoted by $\mathbf{r}$) and unrepresented (denoted by $\mathbf{u}$) subspaces.

In general the reduced representation $\mathbf{r}$ can be any linear or non-linear function of the full representation $\mathbf{z}$ as described in [36, 10]. However, our choice of the reduced representation for RCCE as described above makes the user interface very simple – the user only needs to specify a set of represented species and the rest is taken care of by the implementation. (This simple interface for RCCE has also been incorporated in the commercial CFD package ANSYS Fluent [76, 75].) In addition we have also developed an automated Greedy Algorithm with Local Improvement (GALI) [32] which can be used to select "good" represented species for the RCCE method. Selection of good represented species or constraints is crucial for the overall accuracy of the RCCE method [36, 31]. In the remainder of the text, we use the simple reduced representation given by Eqn.6.3 to describe different implementations of the RCCE method.

When using the RCCE method to represent chemistry in PDF based computations, the particles carry only the reduced representation of chemistry given by $\mathbf{r}$. In the reaction fractional step, given the initial reduced representation $\mathbf{r}(0)$, the task is to compute the *reduced reaction mapping* $\mathbf{r}(t)$ at the end of the reaction fractional time step t. In the following subsections we discuss the different implementations of RCCE used to compute the reduced reaction mapping.

6.4.1 RCCE Implementation using Trajectory In Full Space

Here we briefly describe our implementation of RCCE. A more detailed description can be found in [31, 32].

In our implementation, the reduced reaction mapping is computed by following the three steps (which are illustrated in the sketch Fig.6.1) listed below:

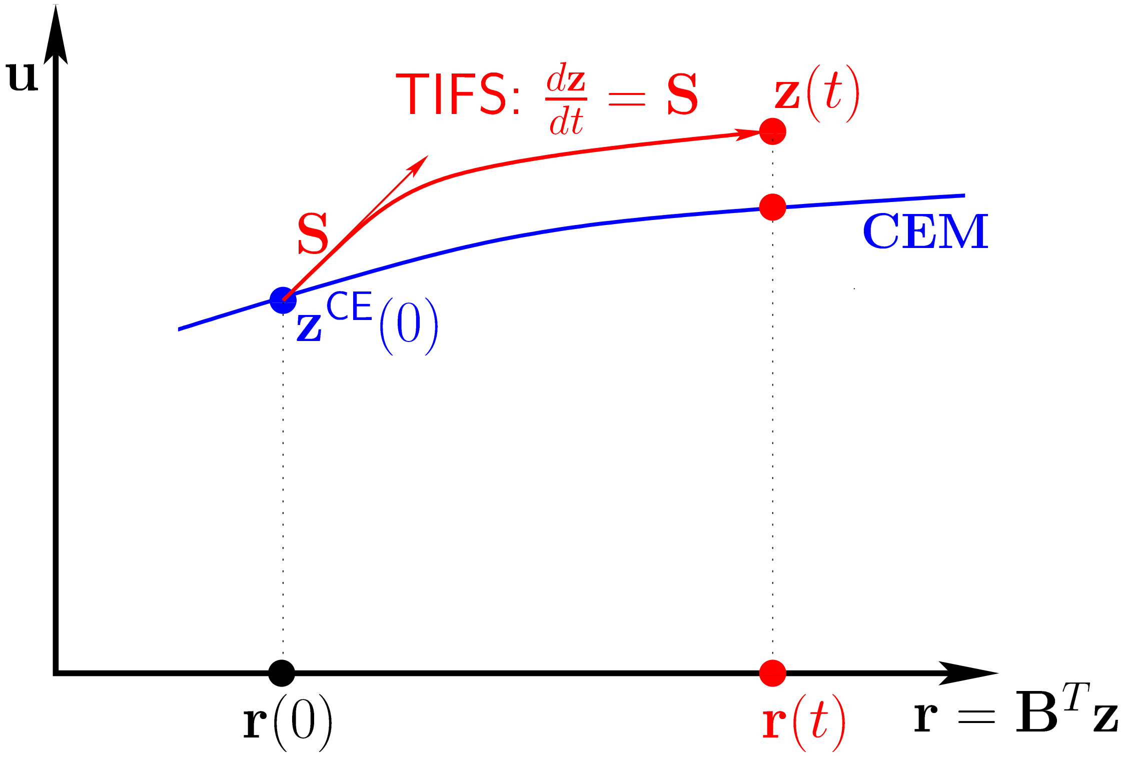
1. *species reconstruction*: given the reduced representation $\mathbf{r}(0)$, the constrained-equilibrium composition (at constant enthalpy h) is computed using CEQ [68] and is denoted as

   $$\mathbf{z}^{CE}(0) \equiv \mathbf{z}^{CE}(\mathbf{r}(0), h). \tag{6.4}$$

   $\mathbf{z}^{CE}$ is a point on the Constrained Equilibrium Manifold (CEM).

2. *trajectory in full space*: starting from $\mathbf{z}^{CE}(0)$, the reaction trajectory (given by Eqn.6.1) is integrated in the full space to obtain the reaction mapping $\mathbf{z}(t)$.

3. *reduction*: from $\mathbf{z}(t)$, the *reduced reaction mapping* is obtained as $\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t)$.



u
TIFS: $\frac{d\mathbf{z}}{dt} = \mathbf{S}$
$\mathbf{z}(t)$
S
CEM
$\mathbf{z}^{\mathsf{CE}}(0)$
$\mathbf{r}(0)$
$\mathbf{r}(t)$
$\mathbf{r} = \mathbf{B}^T\mathbf{z}$

Figure 6.1: Sketch of the composition space (indicated by represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ subspaces) illustrating the reaction mapping computation using the RCCE/TIFS implementation. Given the initial reduced composition at $t = 0$ denoted by $\mathbf{r}(0)$, the reaction mapping $\mathbf{r}(t)$ is obtained in three steps (1) computing the constrained-equilibrium composition at $\mathbf{r}(0)$ denoted by $\mathbf{z}^{CE}(0)$; followed by (2) integrating the trajectory in full space (TIFS) to obtain $\mathbf{z}(t)$; followed by (3) reduction $\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t)$.

We henceforth refer to our implementation of RCCE as RCCE/TIFS – Trajectory In Full Space.

It is important to note here that the CEM is not an *invariant manifold*, i.e., a reaction trajectory originating from a point on the CEM does not necessarily remain on the manifold. Thus the reaction mapping $\mathbf{z}(t)$ obtained in the step (2) of the above RCCE/TIFS implementation need not be on the CEM (as depicted in the sketch Fig.6.1). However, it should be appreciated that the primary objective of the RCCE method being considered here is to obtain an accurate reduced reaction mapping $\mathbf{r}(t)$, and it does not matter if the reaction mapping in the full composition space, $\mathbf{z}(t)$, is not on the CEM. In fact, there exists an infinite number of compositions in the full composition space that yield the same reduced reaction mapping given by $\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t)$. Here, using the TIFS approach, we seek to obtain a reaction mapping $\mathbf{z}(t)$ that yields an accurate reduced reaction mapping $\mathbf{r}(t)$.

6.4.2 Classical RCCE Implementation

Rate-Equations for the Constraints

Here we describe the classical implementation of RCCE first introduced in [41], which involves solving a reduced set of n_r rate-equations for the constraints to compute the reduced reaction mapping $\mathbf{r}(t)$.

Given a composition $\mathbf{z}(t)$ at an instant of time, the reduced composition is given using Eqn.6.3 as $\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t)$. From this relation we get the rate-of-change of $\mathbf{r}(t)$ as

$$\frac{d\mathbf{r}}{dt} = \mathbf{B}^T\frac{d\mathbf{z}}{dt} = \mathbf{B}^T\mathbf{S}(\mathbf{z}). \tag{6.5}$$

Denoting the right-hand-side source vector by $\dot{\mathbf{r}}^e(\mathbf{z}) \equiv \mathbf{B}^T\mathbf{S}(\mathbf{z})$, we get the exact rate-equations for $\mathbf{r}$ as

$$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}^e(\mathbf{z}). \tag{6.6}$$

The classical RCCE implementation seeks to solve directly for the constraints $\mathbf{r}$ using a set of rate-equations based on Eqn.6.6. To solve the rate-equations Eqn.6.6 explicitly for the constraints $\mathbf{r}$, we need a closure for the exact source vector on the right-hand-side $\dot{\mathbf{r}}^e(\mathbf{z})$ such that

$$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}(\mathbf{r}), \tag{6.7}$$

where we denote the right-hand-side approximated source vector by $\dot{\mathbf{r}}$ and henceforth refer to it simply as the *source vector*.

In the classical RCCE method [40], this closure is provided by assuming that $\mathbf{z}$ is always on the CEM (see Eqn.(5.11) to Eqn.(5.17) in [40]), i.e., $\mathbf{z} \equiv \mathbf{z}^{CE}(\mathbf{r})$, which yields

$$\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CE} \equiv \mathbf{B}^T\mathbf{S}(\mathbf{z}^{CE}(\mathbf{r})), \tag{6.8}$$

and gives an explicit set of rate-equations for the constraints $\mathbf{r}$ as

$$\frac{d\mathbf{r}}{dt} = \mathbf{B}^T\mathbf{S}(\mathbf{z}^{CE}(\mathbf{r})). \tag{6.9}$$

Now given $\mathbf{r}(0)$, the reduced reaction mapping $\mathbf{r}(t)$ can be obtained by directly integrating the reduced set of rate-equations given by Eqn.6.9. We henceforth refer to this method as simply RCCE.

The closure provided by Eqn.6.8 in the classical RCCE implementation [40] appears to be a simple and straightforward result. However, there is a logical flaw in the result given by Eqn.6.8, which is based on inconsistent premises, namely

1. $\mathbf{z}$ remains on the CEM, i.e. $\mathbf{z} \equiv \mathbf{z}^{CE}(\mathbf{r})$; and
2. $d\mathbf{z}/dt = \mathbf{S}(\mathbf{z}^{CE}(\mathbf{r}))$.

The composition $\mathbf{z}$ remains on the CEM, only if the chemical source term $\mathbf{S}(\mathbf{z}^{CE}(\mathbf{r}))$ is entirely in the CEM, i.e., the CEM is an *invariant manifold*. However, since the CEM is not an invariant manifold, the reaction trajectory starting from a point on the CEM does not necessarily remain on the manifold, which makes the above RCCE assumptions inherently flawed.

This hidden flawed assumption implied by Eqn.6.8 was first exposed in [90]. The notation used in [90] is different from our notation, and so for consistency, we explain again this hidden assumption using our notation. In the RCCE method, by solving a reduced system of rate-equations for the constraints, the full composition $\mathbf{z}$ is assumed to remain on the CEM. Thus to obtain the source vector $\dot{\mathbf{r}}$, the chemical source term $\mathbf{S}$ is being implicitly *projected* onto the CEM. In the closure provided by Eqn.6.8, the chemical source term $\mathbf{S}$ is implicitly being projected in the unrepresented subspace i.e., the orthogonal complement of the represented subspace spanned by the columns of $\mathbf{B}$. This hidden projection is illustrated in the sketch Fig.6.2. It is not obvious if this is an accurate projection, and in fact it is shown in [90] that a more accurate projection is obtained by computing the reaction source vector on a Close Parallel Inertial Manifold (CPIM). We present results in the later sections which will confirm that the source vector approximation provided by the RCCE method is not accurate.

Rate-Equations for the Constraint Potentials

Numerical integration of the rate-equations for the constraints given by Eqn.6.9 requires the computation of the constrained-equilibrium composition, $\mathbf{z}^{CE}(\mathbf{r})$, at each sub-step of the integration (as described in more detail in Sec.6.8.4). This makes the numerical integration of Eqn.6.9 expensive.

To reduce the computational cost, an alternative implementation of RCCE is described in Section 5.3 of [40] using the *Rate Equations for the Lagrange Multi-*

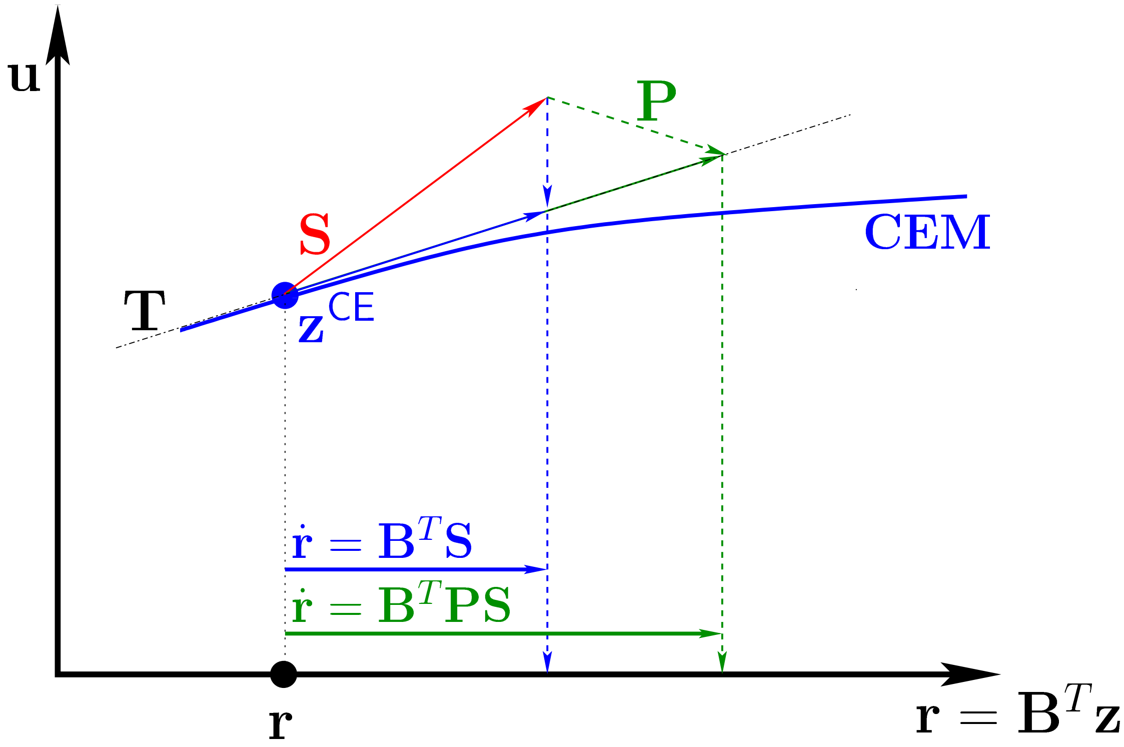

u
P
S
CEM
T
$\mathbf{z}^{CE}$
$\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{S}$
$\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{P}\mathbf{S}$
r
$\mathbf{r} = \mathbf{B}^T\mathbf{z}$

Figure 6.2: Sketch of the composition space (indicated by represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ subspaces) illustrating the projections involved in the RCCE method. Given a reduced composition denoted by $\mathbf{r}$, the classical RCCE implementation computes the source vector $\dot{\mathbf{r}}$ by computing the chemical source term $\mathbf{S}$ at the constrained-equilibrium composition $\mathbf{z}^{CE}$ (on the CEM) and then projecting it back to the represented subspace giving $\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{S}$. This implementation does not take into account the non-invariance of the CEM manifold. Alternatively, one could consider a projector denoted by $\mathbf{P}$, which first projects the source vector $\mathbf{S}$ onto the tangent plane of the CEM (denoted by $\mathbf{T}$) to account for the non-invariance, before projecting it back to the represented subspace to yield $\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{P}\mathbf{S}$.

pliers. This alternative implementation (also referred to as the *Rate-equations for the constraint potentials*) transforms the Rate-equations for the constraints (given by Eqn.6.9) into Rate-equations for the constraint potentials, thereby solving directly for the constraint potentials on the CEM and avoiding the need for computing the constrained-equilibrium composition at each sub-step of the integration. This implementation has been further developed, implemented and tested in [27, 38, 36].

It is important to note here that the above two implementations of RCCE are *mathematically equivalent* and both the implementations make use of the closure provided by Eqn.6.8 (see Eqn.(4.11), Eqn.(5.19) and Eqn.(5.20) in [40]). Numerically, however, the solution obtained by the two implementations may differ due to ODE integration errors. As the numerical integration sub-step time size approaches zero (i.e., for very small ODE integration error tolerance), the two implementations should yield the same solution and thus are consistent with each other. (This has been discussed in a previous work: refer to Fig.1 and the following discussion in [89].) As mentioned later in Sec.6.8.4, in this work we use DDASAC for ODE integration with a relatively small error tolerance of 10^{-8}. Hence we expect both the implementations to yield similar solutions (within the ODE integration error tolerance).

To assess the accuracy of the classical RCCE implementation, either of the aforementioned implementations can be used. In this work, we use the former implementation of RCCE, i.e., the implementation using the rate-equations for the constraints, which will henceforth be simply referred to as the RCCE implementation.

Unlike the RCCE/TIFS implementation, both the aforementioned implementations of RCCE (implicitly) attempt to accurately follow the CEM in the full composition space to obtain the reaction mapping $\mathbf{r}(t)$. However, as will be shown in the later sections, the projection (Eqn.6.8) used in the above implementations of RCCE yields an inaccurate reaction mapping.

In the next section we describe the CPIM method and then in the following section we describe the new Reaction-mixing Attracting Manifold Projector (RAMP), as an extension of the CPIM method, which provides a more accurate projection for implementing the RCCE method.

6.5 Accurate Projection for the RCCE Method

It is clear that the exact rate-equations for the constraints are given by Eqn.6.6, however we need a closure for the right-hand-side source vector denoted by $\dot{\mathbf{r}}$.

In the previous section we showed that the closure provided by the RCCE method is inaccurate because the CEM is not an invariant manifold. One way to account for the non-invariance of the CEM is to replace the chemical source term $\mathbf{S}$ in Eqn.6.8 by a projection of $\mathbf{S}$ onto the tangent plane of the CEM. Then, consistently, the composition remains on the CEM.

To this end, let us consider a general $n_s \times n_s$ projection matrix $\mathbf{P}$ such that the source vector $\dot{\mathbf{r}}$ is obtained (as shown in Fig.6.2) by

$$\dot{\mathbf{r}} = \mathbf{B}^T \mathbf{P} \mathbf{S}. \tag{6.10}$$

There are various choices available for the projection $\mathbf{P}$:

1. project in the unrepresented subspace, as in RCCE;

2. project in the CEM normal subspace;
3. project in the “fast” subspace, as given by ILDM [55] or CSP [44];
4. project using a thermodynamic projector [26];
5. project using a close-parallel manifold, as given by CPIM [90];

Projection (1) implied by RCCE is not accurate as will be shown in the later sections. Projection (2) in the CEM normal subspace depends on the scaling of represented variables and thus contains arbitrariness. Different projections can be obtained if species composition is represented using mass fractions instead of specific moles. Projections (3) and (4) provide more accurate projections, however their implementation is expensive and quite involved. Projection (5) provides a simple correction for the non-invariance of the CEM and an accurate projection, but has issues involving unrealizability and negative entropy production [90].

The projections given by ILDM, CSP and CPIM are based solely on the thermochemistry. This can become problematic when reactions are not fast (e.g., at low temperatures) leading to unrealizability and negative entropy production. The Reaction-mixing Attracting Manifold Projector (RAMP) method introduced below helps address some of these issues.

Here we first present a mathematical formulation for the CPIM method (similar to that provided in [90], however using our notation), and then introduce the RAMP method.

6.5.1 Subspaces and Projections

In the following sections we work with subspaces and projections and so before proceeding further we describe here the notation used for denoting subspaces.

The columns of the constraint matrix $\mathbf{B}$ are not necessarily orthogonal, and it is more convenient to work in terms of orthonormal basis vectors for computing projections. Using the QR or SVD factorization of $\mathbf{B}$ we obtain a set of n_r orthonormal basis vectors (for the represented subspace) denoted by matrix $\mathbf{R}$ of size $n_s \times n_r$ such that span($\mathbf{R}$) $=$ span($\mathbf{B}$); and a set of n_u orthonormal basis vectors (for the orthogonal complement unrepresented subspace) denoted by matrix $\mathbf{U}$ of size $n_s \times n_u$ such that span($\mathbf{U}$) $=$ span($\mathbf{B}$)$^\perp$.

Another important subspace is formed by the tangent vectors of the CEM. Consider an $n_s \times n_r$ matrix $\mathbf{T}^{CEM}$ whose columns are the tangent vectors of the CEM such that

$$\mathbf{T}^{CEM} = \frac{\partial \mathbf{z}^{CE}}{\partial \mathbf{r}}. \tag{6.11}$$

The column vectors of $\mathbf{T}^{CEM}$ span the tangent subspace of the CEM, however the column vectors need not be orthogonal. Using the QR or SVD factorization of $\mathbf{T}^{CEM}$ we obtain a set of n_r orthonormal basis vectors (for the CEM tangent subspace) denoted by matrix $\mathbf{T}$ of size $n_s \times n_r$ such that span($\mathbf{T}$) $=$ span($\mathbf{T}^{CEM}$), and a set of n_u orthonormal basis vectors (for the orthogonal complement CEM normal subspace) denoted by matrix $\mathbf{N}$ of size $n_s \times n_u$ such that span($\mathbf{N}$) $=$ span($\mathbf{T}^{CEM}$)$^\perp$.

We now have two sets of subspaces (or basis vectors) which together span the full n_s-dimensional composition space

1. the represented-unrepresented subspaces spanned by $\{\mathbf{R}, \mathbf{U}\}$, respectively;
2. the CEM tangent-normal subspaces spanned by $\{\mathbf{T}, \mathbf{N}\}$, respectively.

It should be noted that the represented-unrepresented subspaces are fixed once the matrix $\mathbf{B}$ is known, whereas the CEM tangent-normal subspaces are local to the CEM and vary as we move along the CEM.

Any n_s-vector $\mathbf{x}$ in the full space can be decomposed in the following two ways

$$\mathbf{x} = \mathbf{R}(\mathbf{R}^T\mathbf{x}) + \mathbf{U}(\mathbf{U}^T\mathbf{x}), \tag{6.12}$$

or

$$\mathbf{x} = \mathbf{T}(\mathbf{T}^T\mathbf{x}) + \mathbf{N}(\mathbf{N}^T\mathbf{x}). \tag{6.13}$$

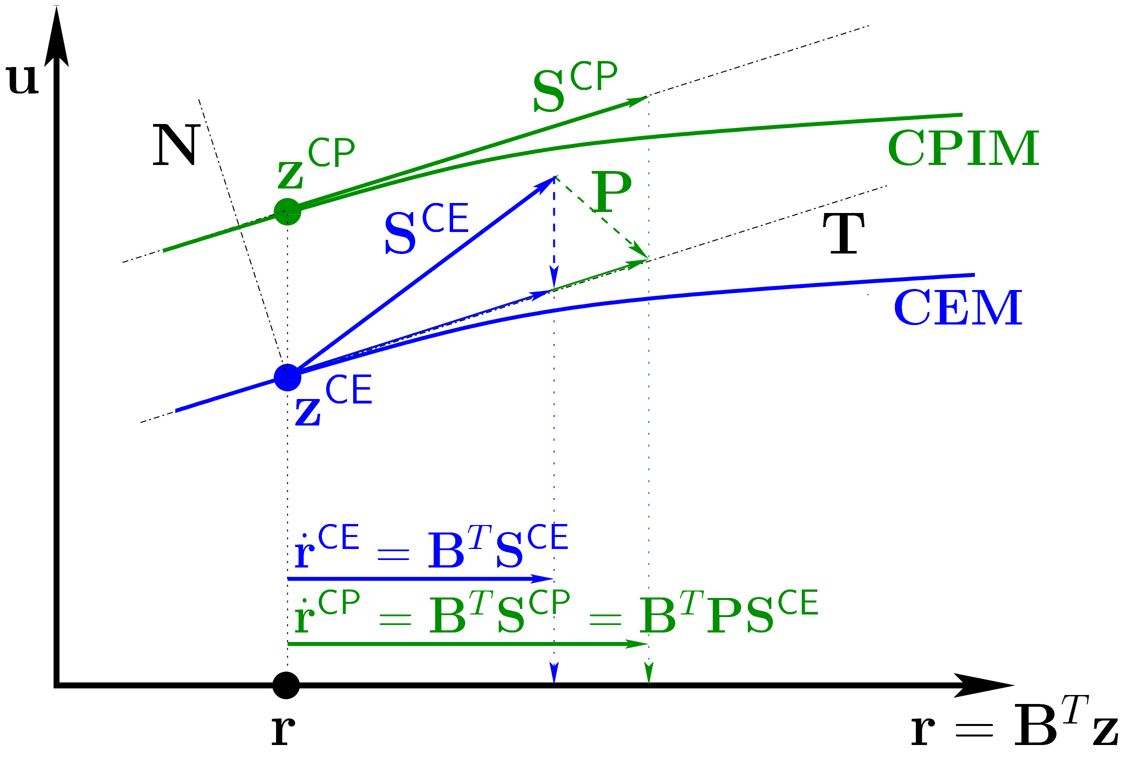
6.5.2 Close Parallel Inertial Manifold

In the CPIM method, it is hypothesized that there is a manifold close-and-parallel to the CEM which is invariant with respect to

$$\frac{d\mathbf{z}^{CP}(t)}{dt} = \mathbf{S}(\mathbf{z}^{CP}(t)), \tag{6.14}$$

where $\mathbf{z}^{CP}$ is a point on the manifold. This means that any reaction trajectory originating from a point on the manifold, by hypothesis, remains on the manifold.

The steps involved in the CPIM method are illustrated in Fig.6.3. For a given constraint $\mathbf{r}$, the constrained-equilibrium composition on the CEM is denoted by $\mathbf{z}^{CE} \equiv \mathbf{z}^{CE}(\mathbf{r})$. The chemical source term at $\mathbf{z}^{CE}$ is denoted by $\mathbf{S}^{CE} \equiv \mathbf{S}(\mathbf{z}^{CE})$.



$\mathbf{u}$
$\mathbf{N}$
$\mathbf{S}^{\mathsf{CP}}$
CPIM
$\mathbf{z}^{\mathsf{CP}}$
$\mathbf{P}$
$\mathbf{S}^{\mathsf{CE}}$
$\mathbf{T}$
CEM
$\mathbf{z}^{\mathsf{CE}}$
$\dot{\mathbf{r}}^{\mathsf{CE}} = \mathbf{B}^T\mathbf{S}^{\mathsf{CE}}$
$\dot{\mathbf{r}}^{\mathsf{CP}} = \mathbf{B}^T\mathbf{S}^{\mathsf{CP}} = \mathbf{B}^T\mathbf{P}\mathbf{S}^{\mathsf{CE}}$
$\mathbf{r}$
$\mathbf{r} = \mathbf{B}^T\mathbf{z}$

Figure 6.3: Sketch of the composition space (indicated by represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ subspaces) illustrating the projections involved in the CPIM method. Given a reduced composition $\mathbf{r}$, the RCCE implementation computes the source vector $\dot{\mathbf{r}}^{CE}$ by projecting the chemical source term $\mathbf{S}^{CE}$ computed at the constrained-equilibrium composition $\mathbf{z}^{CE}$ onto the represented space yielding $\dot{\mathbf{r}}^{CE} = \mathbf{B}^T\mathbf{S}^{CE}$. Alternatively, in the CPIM method, a manifold close-and-parallel to the CEM is considered to evaluate the chemical source term $\mathbf{S}^{CP}$ (which lies entirely in the CEM and CPIM tangent space denoted by $\mathbf{T}$). In the CPIM method, the source vector is given by $\dot{\mathbf{r}}^{CP} = \mathbf{B}^T\mathbf{S}^{CP}$, which can be rewritten in terms of a projector $\mathbf{P}$ such that $\dot{\mathbf{r}}^{CP} = \mathbf{B}^T\mathbf{P}\mathbf{S}^{CE}$.

Now we consider a point $\mathbf{z}^{CP}$ on the CPIM such that

$$\mathbf{B}^T\mathbf{z}^{CP} = \mathbf{B}^T\mathbf{z}^{CE} = \mathbf{r}, \tag{6.15}$$

and we denote

$$\delta\mathbf{z}^{CP} \equiv \mathbf{z}^{CP} - \mathbf{z}^{CE}. \tag{6.16}$$

Since $\mathbf{B}^T\delta\mathbf{z}^{CP} = 0$, this means that $\delta\mathbf{z}^{CP}$ lies entirely in the unrepresented subspace, and so we can express

$$\delta\mathbf{z}^{CP} = \mathbf{U}\delta\mathbf{u}^{CP}, \tag{6.17}$$

where $\delta\mathbf{u}^{CP}$ is an n_u-vector in the unrepresented subspace spanned by $\mathbf{U}$.

By the close-and-parallel assumption, $\mathbf{z}^{CP}$ is close to $\mathbf{z}^{CE}$, and so we can express $\mathbf{S}^{CP} \equiv \mathbf{S}(\mathbf{z}^{CP})$ using a linear approximation about $\mathbf{S}^{CE}$ as follows

$$\mathbf{S}^{CP} = \mathbf{S}^{CE} + \mathbf{J}\delta\mathbf{z}^{CP} = \mathbf{S}^{CE} + \mathbf{J}\mathbf{U}\delta\mathbf{u}^{CP}, \tag{6.18}$$

where $\mathbf{J}$ is the Jacobian evaluated at $\mathbf{z}^{CE}$

$$\mathbf{J} \equiv \frac{\partial\mathbf{S}(\mathbf{z}^{CE})}{\partial\mathbf{z}}. \tag{6.19}$$

Since the CPIM is invariant and parallel to CEM, at $\mathbf{z}^{CP}$ the chemical source term $\mathbf{S}^{CP} \equiv \mathbf{S}(\mathbf{z}^{CP})$ must be in the CEM tangent subspace. This enforces that $\mathbf{N}^T\mathbf{S}^{CP} = \mathbf{0}$, which gives

$$\mathbf{N}^T\mathbf{S}^{CE} + \mathbf{N}^T\mathbf{J}\mathbf{U}\delta\mathbf{u}^{CP} = \mathbf{0}, \tag{6.20}$$

and solving for $\delta\mathbf{u}^{CP}$ we obtain

$$\delta\mathbf{u}^{CP} = \mathbf{L}^{-1}\mathbf{N}^T\mathbf{S}^{CE}, \tag{6.21}$$

where we denote

$$\mathbf{L} \equiv -\mathbf{N}^T\mathbf{J}\mathbf{U}. \tag{6.22}$$

Substituting $\delta\mathbf{u}^{CP}$ in Eqn.6.17 we obtain

$$\mathbf{z}^{CP} = \mathbf{z}^{CE} + \mathbf{U}\mathbf{L}^{-1}\mathbf{N}^T\mathbf{S}^{CE}, \tag{6.23}$$

and using Eqn.6.18 we get

$$\mathbf{S}^{CP} = (\mathbf{I} + \mathbf{J}\mathbf{U}\mathbf{L}^{-1}\mathbf{N}^T)\mathbf{S}^{CE}. \tag{6.24}$$

We now use $\mathbf{S}^{CP}$ to compute the source vector $\dot{\mathbf{r}}$ in the rate-equations for the constraints Eqn.6.7 and denote it as

$$\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CP} \equiv \mathbf{B}^T\mathbf{S}^{CP}. \tag{6.25}$$

Thus the rate-equations for the constraints Eqn.6.7 using the CPIM approach are given as

$$\frac{d\mathbf{r}}{dt} = \mathbf{B}^T\mathbf{S}^{CP}, \tag{6.26}$$

which can be re-written as

$$\frac{d\mathbf{r}}{dt} = \mathbf{B}^T\mathbf{P}\mathbf{S}(\mathbf{z}^{CE}(\mathbf{r})), \tag{6.27}$$

where

$$\mathbf{P} \equiv \mathbf{I} + \mathbf{J}\mathbf{U}\mathbf{L}^{-1}\mathbf{N}^T, \tag{6.28}$$

provides a more accurate projection of the chemical source term evaluated on the CEM onto the tangent subspace.

The main issues involved in the CPIM method as highlighted in [90] are:

1. if the matrix $\mathbf{L}$ is ill-conditioned (especially at low temperatures) then the composition $\mathbf{z}^{CP}$ can be unrealizable;
2. the linear approximation of $\mathbf{S}^{CP}$ (for ill-conditioned matrix $\mathbf{L}$) may lead to negative entropy production.

These issues were handled in the CPIM method [90] by considering a linear combination of $\mathbf{S}^{CP}$ and $\mathbf{S}^{CE}$ to ensure positive entropy production and realizability.

Here we present the Reaction-mixing Attracting Manifold Projector (RAMP) [71] as an extension to the CPIM approach, which provides a much simpler way of handling the unrealizability and negative entropy production issues.

6.6 Reaction-mixing Attracting Manifold Projector

In the RAMP approach, similar to CPIM, a hypothetical manifold close-and-parallel to the CEM is considered, however now this manifold is assumed to be invariant with respect to the following evolution equation

$$\frac{d\mathbf{z}^{CP}(t)}{dt} = \mathbf{S}(\mathbf{z}^{CP}(t)) - \omega(\mathbf{z}^{CP}(t) - \mathbf{z}^{CE}(t)), \tag{6.29}$$

where $\mathbf{z}^{CP}(t)$ is a point on the CPIM; $\mathbf{z}^{CE}(t) \equiv \mathbf{z}^{CE}(\mathbf{r} = \mathbf{B}^T\mathbf{z}^{CP}(t))$ is a point on the CEM such that $\mathbf{B}^T\mathbf{z}^{CP} = \mathbf{B}^T\mathbf{z}^{CE}$; and ω is a specified mixing (relaxation) rate.

The inclusion of the additional mixing term in the evolution equation Eqn.6.29 is inspired by the general class of reaction-diffusion manifolds as described in [71], and in particular the REDIM method [13]. The inclusion of the additional mixing term helps address the realizability issues encountered in the CPIM approach, where the evolution equation Eqn.6.14 contains only the reaction term.

For the RAMP approach, the invariance condition for the evolution equation

Eqn.6.29 is given as

$$\mathbf{N}^T \left[\frac{d\mathbf{z}^{CP}(t)}{dt} \right] = \mathbf{0}. \tag{6.30}$$

Using the same notation as for the CPIM, we can express

$$\frac{d\mathbf{z}^{CP}(t)}{dt} = \mathbf{S}^{CP} - \omega \delta \mathbf{z}^{CP}. \tag{6.31}$$

Following the same steps as for the CPIM method we can express $\mathbf{S}^{CP}$ by a linear approximation given by Eqn.6.18, and $\delta \mathbf{z}^{CP} = \mathbf{U} \delta \mathbf{u}^{CP}$, which gives

$$\frac{d\mathbf{z}^{CP}(t)}{dt} = \mathbf{S}^{CE} + (\mathbf{J} - \omega \mathbf{I}) \, \mathbf{U} \delta \mathbf{u}^{CP}. \tag{6.32}$$

Using the invariance condition we obtain

$$\mathbf{N}^T \mathbf{S}^{CE} + \mathbf{N}^T (\mathbf{J} - \omega \mathbf{I}) \, \mathbf{U} \delta \mathbf{u}^{CP} = \mathbf{0}, \tag{6.33}$$

which gives

$$\delta \mathbf{u}^{CP} = \mathbf{L}^{-1} \mathbf{N}^T \mathbf{S}^{CE}, \tag{6.34}$$

where

$$\mathbf{L} \equiv \mathbf{L}(\omega) \equiv -\mathbf{N}^T (\mathbf{J} - \omega \mathbf{I}) \, \mathbf{U}. \tag{6.35}$$

So the only difference from the CPIM method is the additional $\omega\mathbf{I}$ term in the definition of $\mathbf{L}$. This term makes $\mathbf{L}$ better conditioned, because by assumption if the close-parallel manifold is an *attracting* manifold (as is implicitly assumed), then the eigenvalues of the Jacobian $\mathbf{J}$ have negative real parts, which are further decreased by ω due to the introduction of the relaxation term in the evolution equation.

The new projector with the RAMP method (which is a function of the relaxation parameter ω) is given as

$$\mathbf{P}(\omega) \equiv \mathbf{I} + \mathbf{J} \mathbf{U} \mathbf{L}^{-1} \mathbf{N}^T, \tag{6.36}$$

with $\mathbf{L}$ given by Eqn.6.35.

The source vector approximation given by the RAMP method is denoted as

$$\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CP}(\omega) \equiv \mathbf{B}^T \mathbf{P}(\omega) \mathbf{S}(\mathbf{z}^{CE}(\mathbf{r})). \tag{6.37}$$

The rate-equations for the constraints Eqn.6.7 using the RAMP approach are given as

$$\frac{d\mathbf{r}}{dt} = \mathbf{B}^T \mathbf{P}(\omega) \mathbf{S}(\mathbf{z}^{CE}(\mathbf{r})). \tag{6.38}$$

It is important to highlight here the properties of the RAMP method for the two limiting values of the relaxation rate parameter ω:

1. $\omega = 0$. From the definition of the evolution equation Eqn.6.29, it is obvious that $\omega = 0$ corresponds to the CPIM approach, with the projector $\mathbf{P}(\omega = 0) = \mathbf{P}$ provided by the CPIM method.
2. $\omega \to \infty$. From the definition of $\mathbf{L}$ (Eqn.6.35), we see that $\mathbf{L}^{-1}(\omega \to \infty) = \mathbf{0}$, i.e., inverse of $\mathbf{L}$ is singular. This yields from Eqn.6.36, that $\mathbf{P}(\omega \to \infty) = \mathbf{I}$, and the RAMP method corresponds to the classical RCCE implementation, since $\dot{\mathbf{r}}^{CP}(\omega \to \infty) = \mathbf{B}^T \mathbf{P}(\omega \to \infty) \mathbf{S}^{CE} = \mathbf{B}^T \mathbf{S}^{CE} = \dot{\mathbf{r}}^{CE}$.

In short this shows that the value of ω controls the closeness of the CPIM to the CEM in the RAMP approach. For $\omega = 0$, we obtain the CPIM method, and for $\omega \to \infty$, the CPIM collapses with the CEM and we get back the classical RCCE implementation. We will see in the results presented in the following sections, that by choosing an appropriate value of ω in the RAMP approach, we can address the unrealizability issues of the CPIM method and also account for the non-invariance of the CEM-based RCCE method.

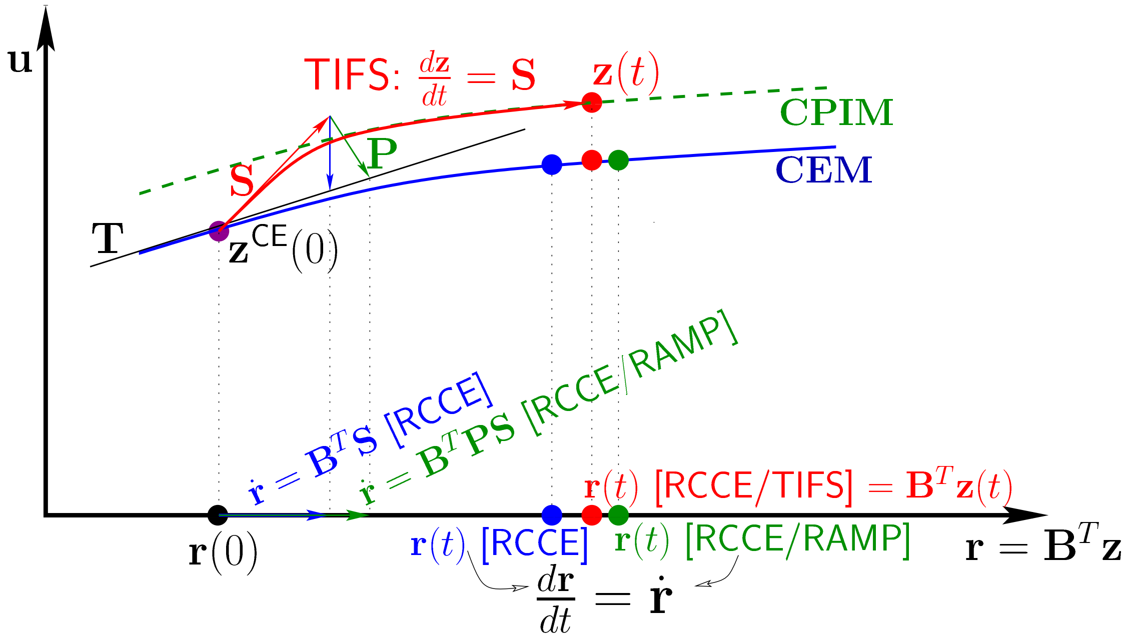

u
TIFS: $\frac{d\mathbf{z}}{dt} = \mathbf{S}$
$\mathbf{z}(t)$
CPIM
CEM
S
P
T
$\mathbf{z}^{\mathsf{CE}}(0)$
$\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{S}$ [RCCE]
$\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{PS}$ [RCCE/RAMP]
$\mathbf{r}(t)$ [RCCE/TIFS] $= \mathbf{B}^T\mathbf{z}(t)$
$\mathbf{r}(0)$
$\mathbf{r}(t)$ [RCCE]
$\mathbf{r}(t)$ [RCCE/RAMP]
$\mathbf{r} = \mathbf{B}^T\mathbf{z}$
$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}$

Figure 6.4: Sketch of the composition space (indicated by represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ subspaces) illustrating reaction mapping computation using the three implementations of the RCCE method: RCCE/TIFS, RCCE and RCCE/RAMP. Given the initial reduced composition $\mathbf{r}(0)$, (1) the RCCE/TIFS implementation computes the reaction mapping by following the trajectory in full space starting from $\mathbf{z}^{CE}(0)$ to obtain $\mathbf{z}(t)$ followed by reduction to yield the reaction mapping $\mathbf{r}(t)$ [RCCE/TIFS] $= \mathbf{B}^T\mathbf{z}(t)$; (2) the RCCE implementation solves a reduced system of ODEs with source vector $\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{S}$ to obtain $\mathbf{r}(t)$ [RCCE]; and (3) the RCCE/RAMP implementation solves the reduced system of ODEs (as in RCCE) using an alternative source vector $\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{PS}$ to obtain $\mathbf{r}(t)$ [RCCE/RAMP].

6.7 Partially-Stirred Reactor

In this section we describe the partially-stirred reactor (PaSR) test case used to study the three implementation of RCCE (illustrated in Fig.6.4) described in this work:

1. RCCE/TIFS: the implementation used in our previous works (described

in Sec.6.4.1);

2. RCCE: the classical RCCE implementation (described in Sec.6.4.2); and
3. RCCE/RAMP: the new implementation based on CPIM (described in Sec.6.6).

To compare these different implementations of the RCCE method, we consider methane/air premixed combustion in a partially-stirred reactor (PaSR). A detailed description of the PaSR is provided in [66]: here we briefly describe the details pertinent to the current study.

The PaSR can be used to study PDF particle implementation applied to a statistically homogeneous flow. In the PaSR, the thermochemical composition is represented by a fixed number of particles. The particle composition evolves due to *mixing* and *reaction* in fractional steps. The mixing is implemented using a pairwise mixing model [66]. Our implementation of PaSR allows the particle chemistry to be represented in any one of the following ways

1. detailed chemistry with n_s species, with reaction mapping computed using ODE integration (also referred to as *direct evaluation*);
2. detailed chemistry with n_s species, with reaction mapping computed using ISAT;
3. reduced chemistry with n_r represented variables using RCCE, with reaction mapping computed using the combined ISAT/RCCE approach [32].

For methane/air combustion in the PaSR, we use the same operating conditions as in [32]. The PaSR involves two inflowing streams: (1) a stoichiometric premixed stream of methane/air mixture at 600 K; and (2) a pilot stream of equilibrium products of composition of stream 1. The streams flow into the PaSR with a mass flow rate ratio of 0.95:0.05. Initially all the particles are set to the pilot stream composition. The pressure is atmospheric throughout. Other important parameters include: the number of particles, $N_p = 100$; the residence time, $\tau_{res} = 20$ ms*; the mixing time scale, $\tau_{mix} = 1$ ms; and the pairing time scale, $\tau_{pair} = 1$ ms. The chemistry is represented using the GRI-Mech 1.2 mechanism involving $n_s = 31$ species composed of $n_e = 4$ elements.

We perform a PaSR simulation with a reaction time step, $\Delta t = 0.033$ ms, and *reaction mapping* computed using ISAT (with error tolerance, $\epsilon_{tol} = 10^{-5}$) with the detailed mechanism. During the simulation, we save the compositions of the first N particles that result in an *add* in the ISAT table (and so are distinct) denoted by $\mathbf{z}^{(n)}(0)$ and their reaction mappings $\mathbf{z}^{(n)}(\Delta t)$ for $n = 1$ to N. We use these test compositions to study different implementations of the RCCE method. In this work, in all the tests, we use $N = 2,500$ test compositions. Henceforth, we use t to denote a general reaction time step, and Δt to denote the exact time step $\Delta t = 0.033$ ms used in the PaSR test to compute the reaction mappings.

It is important to note here that, in the PaSR, *reaction* causes the particle compositions to move towards to a low-dimensional attracting manifold and *mixing* causes the particle compositions to be pulled away from this manifold (as illustrated in Fig.6.5). Hence the test compositions at the beginning of the reaction fractional step $\mathbf{z}^{(n)}(0)$ are expected to be away from the attracting manifold, and the compositions at the end of the step $\mathbf{z}^{(n)}(\Delta t)$ are expected to be closer to a low-dimensional attracting manifold. Unless otherwise specified explicitly, in the following tests to study different implementations of RCCE we use the test

*the residence time was misreported in [32] to be $\tau_{res} = 10$ ms

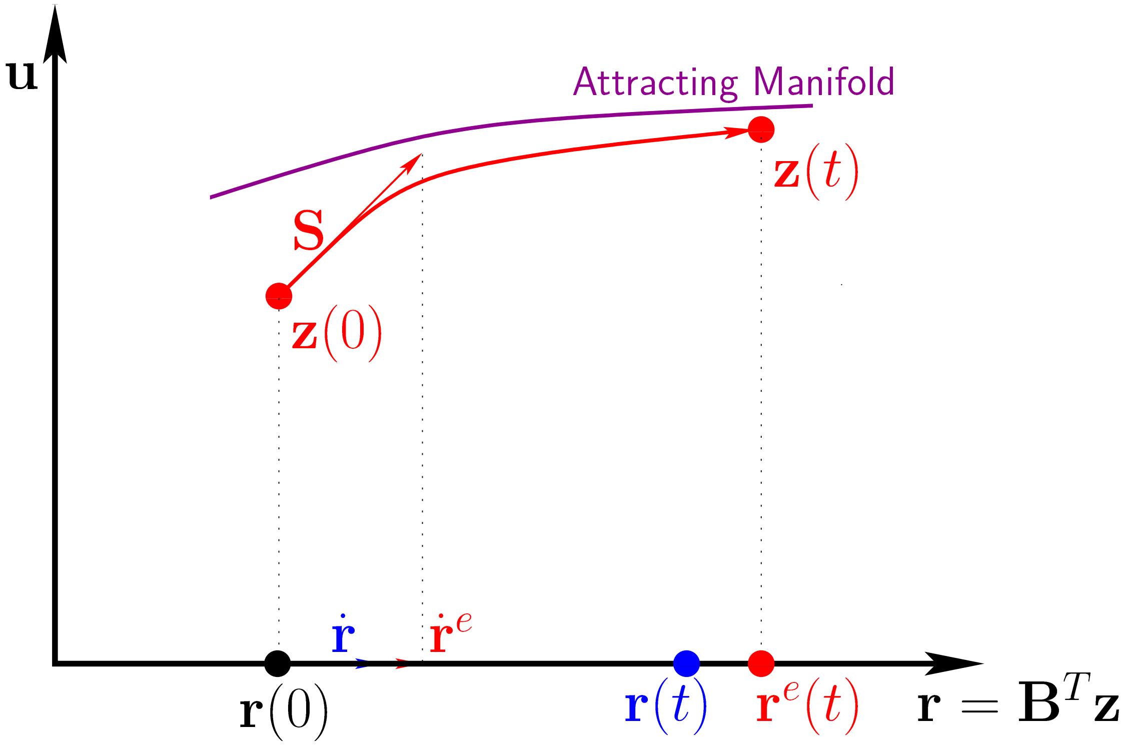


Figure 6.5: Sketch depicting a test composition $\mathbf{z}(0)$ and its reaction mapping $\mathbf{z}(t)$ in the full composition space (indicated by the represented $\mathbf{r}$ and unrepresented $\mathbf{u}$ subspaces). The reaction makes the test composition move closer to a low-dimensional attracting manifold. In the represented subspace, $\dot{\mathbf{r}}^e = \mathbf{B}^T\mathbf{S}$ denotes the exact source vector and $\dot{\mathbf{r}}$ denotes a source vector approximation obtained using one of the RCCE implementations. In addition, $\mathbf{r}^e(t) = \mathbf{B}^T\mathbf{z}(t)$ denotes the exact reaction mapping and $\mathbf{r}(t)$ denotes the reaction mapping obtained using one of the RCCE implementations.

compositions from the beginning of the reaction fractional step $\mathbf{z}^{(n)}(0)$, which typically will be encountered in real PDF computations. We use the test compositions $\mathbf{z}^{(n)}(\Delta t)$ only in a few cases to study the validity of the CPIM approximation.

Table 6.1: Sets of represented species obtained using GALI (with 31-species GRI-Mech 1.2 mechanism) for dimension reduction of methane/air premixed combustion with RCCE for $n_{rs} = 9$ to 15 (obtained from Table 3 in [32]).

n_{rs}	Represented Species
9	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3
10	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, C_2H_6, C_2H_4
11	CH_4, CO_2, H_2, O_2, H, OH, O, H_2O, CH_3, HO_2, CO
12	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O
13	CH_4, CO_2, H_2, O_2, H, OH, O, CH_3OH, CH_3, HO_2, CO, H_2O, CH_2CO
14	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O, CH_2CO, C_2H_5
15	CH_4, CO_2, H_2, O_2, H, OH, O, CH_2O, CH_3, HO_2, CO, H_2O, CH_2CO, C_2H_5, CH_2

6.7.1 PaSR Tests to Study RCCE Implementations

We use the test compositions saved from the PaSR run in the full dimension to study different implementations of the RCCE method. We perform RCCE tests over a range of values of n_{rs} from 9 to 15, corresponding to $n_r = n_{rs} + n_e$ in the range 13 to 19 (which yield less than 3% *reduction-tabulation error* [32]) with represented species selected using GALI. We use the same represented species as those listed in Table 3 in [32], which are obtained using GALI for the same test case – methane/air premixed combustion in PaSR with chemistry represented using the 31-species GRI-Mech 1.2 mechanism – as used in the current study. The relevant sets of represented species for $n_{rs} =$ 9 to 15 used in this study are listed again in Table 6.1.

Now for a specified set of represented species for performing dimension reduction with RCCE, we form the constraint matrix $\mathbf{B}$. At each selected particle composition $\mathbf{z} \equiv \mathbf{z}^{(n)}(0)$ (and in some cases $\mathbf{z} \equiv \mathbf{z}^{(n)}(\Delta t)$) for $n = 1$ to $N = 2,500$, we then compute

- the chemical source term, $\mathbf{S} \equiv \mathbf{S}(\mathbf{z})$;
- the exact source vector, $\dot{\mathbf{r}}^e = \mathbf{B}^T\mathbf{S}$ (as illustrated in Fig.6.5);
- the reduced composition, $\mathbf{r} = \mathbf{B}^T\mathbf{z}$;
- the constrained-equilibrium composition, $\mathbf{z}^{CE} \equiv \mathbf{z}^{CE}(\mathbf{r})$;
- the chemical source term, $\mathbf{S}^{CE} \equiv \mathbf{S}(\mathbf{z}^{CE})$;
- the orthogonal projections of $\mathbf{S}$ and $\mathbf{S}^{CE}$ onto the CEM tangent plane, denoted by $\mathbf{S}_t$ and $\mathbf{S}_t^{CE}$, respectively; and the angles between $\mathbf{S}$ and $\mathbf{S}_t$, and $\mathbf{S}^{CE}$ and $\mathbf{S}_t^{CE}$, denoted by $\angle(\mathbf{S}, \mathbf{S}_t)$ and $\angle(\mathbf{S}^{CE}, \mathbf{S}_t^{CE})$, respectively, to assess the non-invariance of the CEM manifold;
- the source vector given by RCCE, $\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CE} = \mathbf{B}^T\mathbf{S}^{CE}$;
- the source vector given by RAMP, $\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CP}(\omega) = \mathbf{B}^T\mathbf{P}(\omega)\mathbf{S}^{CE}$, for different values of the relaxation rate ω;
- the exact reaction mapping $\mathbf{r}^e(t) = \mathbf{B}^T\mathbf{z}(t)$ and reaction mappings $\mathbf{r}(t)$ using the three implementations of RCCE (as illustrated in Fig.6.4 and Fig.6.5).

In the following section we look at various scatter plots to analyze these data, and in addition quantify and compare the errors involved in the three implementations of RCCE.

6.8 Results

6.8.1 Non-Invariance

In this section we examine the “degree” of non-invariance of the CEM, by looking at the angle between the chemical source term $\mathbf{S}^{CE}$ and its orthogonal projection $\mathbf{S}_t^{CE}$ on the CEM (denoted by $\angle(\mathbf{S}^{CE}, \mathbf{S}_t^{CE})$). Fig.6.6 shows scatter plots of $\angle(\mathbf{S}^{CE}, \mathbf{S}_t^{CE})$ versus temperature T computed using the test compositions saved at $t = 0$ and $t = \Delta t$. If the CEM were an invariant manifold, then we would have $\angle(\mathbf{S}^{CE}, \mathbf{S}_t^{CE}) = 0$. However, in these scatter plots we see that $\angle(\mathbf{S}^{CE}, \mathbf{S}_t^{CE})$ is as large as 35^o at both $t = 0$ and $t = \Delta t$. This confirms that the CEM is not an invariant manifold, and in fact the reaction trajectories could be moving away from the manifold at large angles. This non-invariance introduces a large error in the RCCE implementation, in which $\mathbf{S}^{CE}$ is directly projected onto the constrained subspace without accounting for this non-invariance.

We now examine the angle between the chemical source term $\mathbf{S}$ (computed at the test composition) and its orthogonal projection onto the CEM $\mathbf{S}_t$ (denoted by $\angle(\mathbf{S}, \mathbf{S}_t)$), which gives a measure of the orientation of the chemical source term $\mathbf{S}$ relative to the CEM. Fig.6.7 shows scatter plot of $\angle(\mathbf{S}, \mathbf{S}_t)$ versus temperature T computed using the test compositions saved at $t = 0$ and $t = \Delta t$. Here we see that at $t = 0$, the angle $\angle(\mathbf{S}, \mathbf{S}_t)$ is relatively large (around 10^o) because the test compositions are pulled away from the attracting manifold due to mixing. However, at $t = \Delta t$, the angle $\angle(\mathbf{S}, \mathbf{S}_t)$ is very small (less than 2^o for $T > 1000$ K), which shows that the reaction takes the compositions closer to an attracting manifold, and this attracting manifold is nearly parallel to the CEM (because the angle is measured relative to the CEM). This observation confirms the CPIM

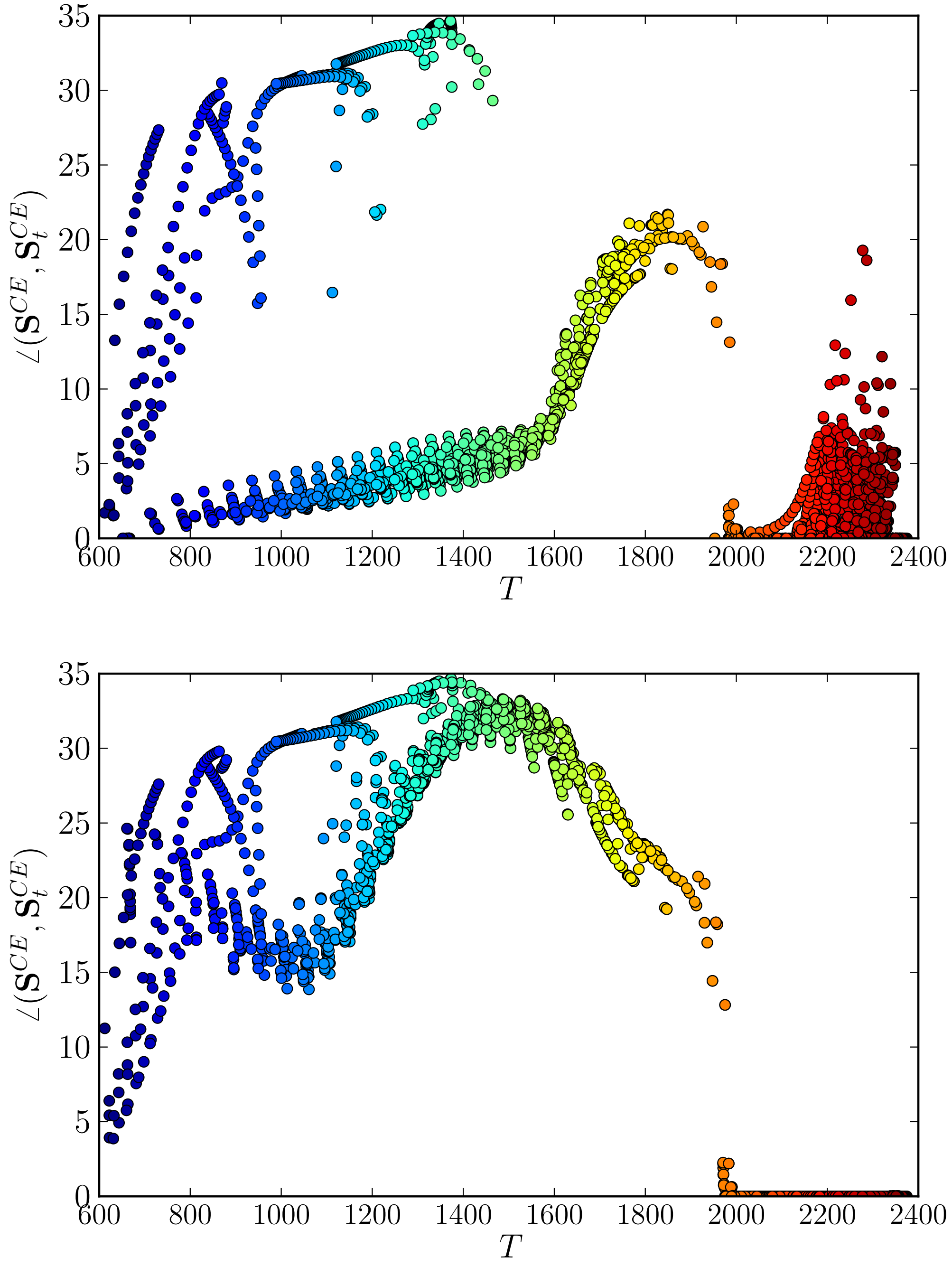


Figure 6.6: Scatter plots of angle (in degrees) between the chemical source term $\mathbf{S}^{CE}$ (evaluated on the CEM) and its orthogonal projection onto the CEM $\mathbf{S}_t^{CE}$ versus temperature T computed using the test compositions at $t = 0$ (top) and at $t = \Delta t$ (bottom).

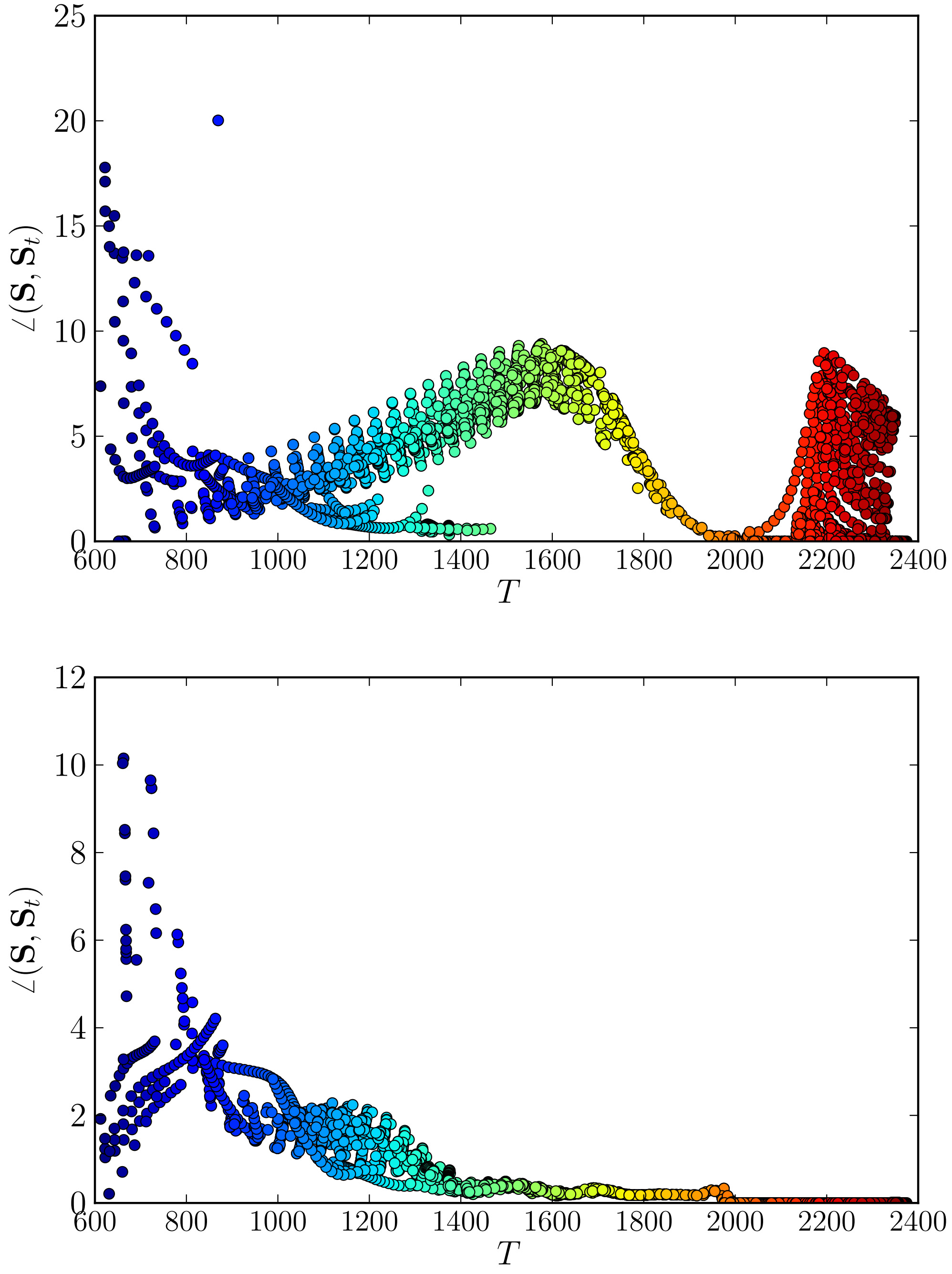


Figure 6.7: Scatter plot of angle (in degrees) between the chemical source term $\mathbf{S}$ and its orthogonal projection onto the CEM $\mathbf{S}_t$ versus temperature T computed using the test compositions at $t = 0$ (top) and at $t = \Delta t$ (bottom).

idea that there exists an invariant manifold close-and-parallel to the CEM. This is the reason why the CPIM and its extension RAMP are able to give a better approximation for the RCCE source vector (as quantified in the results included below).

6.8.2 Realizability and Entropy Production

Here we briefly examine the realizability and entropy production issues highlighted in the CPIM work [90] and mentioned in Sec.6.5.2, which are also pertinent to the extended RAMP approach.

The RCCE/TIFS implementation computes the reaction mapping following the reaction trajectory in the full space by integrating the full system of ODEs (Eqn.6.1) using the chemical source term $\mathbf{S}(\mathbf{z})$. Since there is no projection involved in this method, there are no realizability or negative entropy production issues in this implementation.

The RCCE implementation uses a reduced system of ODEs obtained by projecting the chemical source term $\mathbf{S}(\mathbf{z})$ directly onto the constrained subspace given as $\dot{\mathbf{r}} = \mathbf{B}^T\mathbf{S}(\mathbf{z})$. During the computation of the reaction mapping, the chemical composition evolves through a series of constrained-equilibrium compositions on the CEM, and it is shown in [40] that this implementation ensures (mathematically) non-negative entropy production and realizability.

The CPIM and RAMP approaches use an alternative projection $\mathbf{P}$ which need not necessarily ensure non-negative entropy production as described in [90]. There are two main concerns:

1. for an ill-conditioned matrix $\mathbf{L}(\omega)$, the composition $\mathbf{z}^{CP}(\omega)$ may not be realizable; and
2. the linear approximation $\mathbf{S}^{CP}(\omega)$ (for ill-conditioned matrix $\mathbf{L}(\omega)$) may lead to negative entropy production.

However, since $\mathbf{z}^{CP}(\omega)$ does not directly appear in the definition of the projector $\mathbf{P}(\omega)$ given by Eqn.6.36, the realizability of $\mathbf{z}^{CP}(\omega)$ is not a major concern. It is only the linear approximation to the chemical source term $\mathbf{S}^{CP}(\omega)$ that directly influences the projector, $\mathbf{P}(\omega)$.

To analyze these issues we compute the following quantities using the saved test compositions:

1. the minimum species composition in $\mathbf{z}^{CP}(\omega)$ denoted by $\min(\mathbf{z}^{CP})$; and
2. the ratio of the entropy production given by the RAMP approach denoted by $\dot{s}^{CP}$ to the actual entropy production at the test composition denoted by $\dot{s}$ (which are defined below).

The entropy production rate $\dot{s}$ at a test composition $\mathbf{z}$ is given as

$$\dot{s} = \eta^T(\mathbf{z})\mathbf{S}(\mathbf{z}), \tag{6.39}$$

where η is the entropy gradient vector (at constant enthalpy h and pressure p) given as

$$\eta = \left.\frac{\partial s}{\partial \mathbf{z}}\right|_{h,p} = \mathbf{s} - \frac{\mathbf{h}}{T}, \tag{6.40}$$

where s and $\mathbf{h}$ are molar entropies and enthalpies, respectively.

The entropy production given by the RAMP approach denoted by $\dot{s}^{CP}$ is given as

$$\dot{s}^{CP} = \eta^T(\mathbf{z}^{CE})\mathbf{PS}(\mathbf{z}^{CE}). \tag{6.41}$$

First we look at the realizability issue. We consider the test compositions saved at $t = 0$, and compute the minimum species composition in $\mathbf{z}^{CP}$ denoted by $\min(\mathbf{z}^{CP})$ at each test composition. For $\mathbf{z}^{CP}$ to be realizable, we must have $\min(\mathbf{z}^{CP}) \geq 0$. Fig.6.8 shows the value of -$\min(\mathbf{z}^{CP})$ for a range of values of ω from 0 to 10^9 s^{-1}. Each subplot shows only the test compositions for which $\mathbf{z}^{CP}$ is unrealizable, i.e. -$\min(\mathbf{z}^{CP}) > 0$. The title of each subplot indicates the percentage of test compositions (out of the overall 2,500) for which we have an unrealizable $\mathbf{z}^{CP}$, i.e. -$\min(\mathbf{z}^{CP}) > 0$. In addition, in parentheses, we indicate the percentage of these unrealizable compositions for which we have the value of -$\min(\mathbf{z}^{CP})$ above a reference value of 10^{-12} (indicative of round-off error). For this test case, we notice that at small values of ω, over 60% of the compositions are unrealizable and some of them have relatively large negative compositions in the order -10^{-4}. However as the value of ω increases, the percentage of unrealizable compositions decreases to less than 7% for values of $\omega > 10^5$, and less than 30% of these unrealizable compositions have $\min(\mathbf{z}^{CP}) < -10^{-12}$. The maximum magnitude of the negative species composition decreases to a value of less than 10^{-6} for $\omega \geq 10^5$ s^{-1} and a value less than 10^{-22} for $\omega \geq 10^8$ s^{-1}. As described earlier, larger values of ω pull the CPIM manifold closer to the CEM and improve the conditioning of the $\mathbf{L}(\omega)$ matrix, thereby making more compositions $\mathbf{z}^{CP}(\omega)$ realizable. Nonetheless, as mentioned earlier, since $\mathbf{z}^{CP}(\omega)$ does not directly appear in the definition of the projector $\mathbf{P}(\omega)$, realizability of $\mathbf{z}^{CP}(\omega)$ is not a major concern.

Next we look at the ratio of entropy productions, $\dot{s}^{CP}/\dot{s}$, as shown in Fig.6.9 for the same range of values of ω. We notice that for the current test case, none of the test compositions yields a negative entropy production using the RAMP approach (even at $\omega = 0$, which corresponds to the CPIM method). However, we

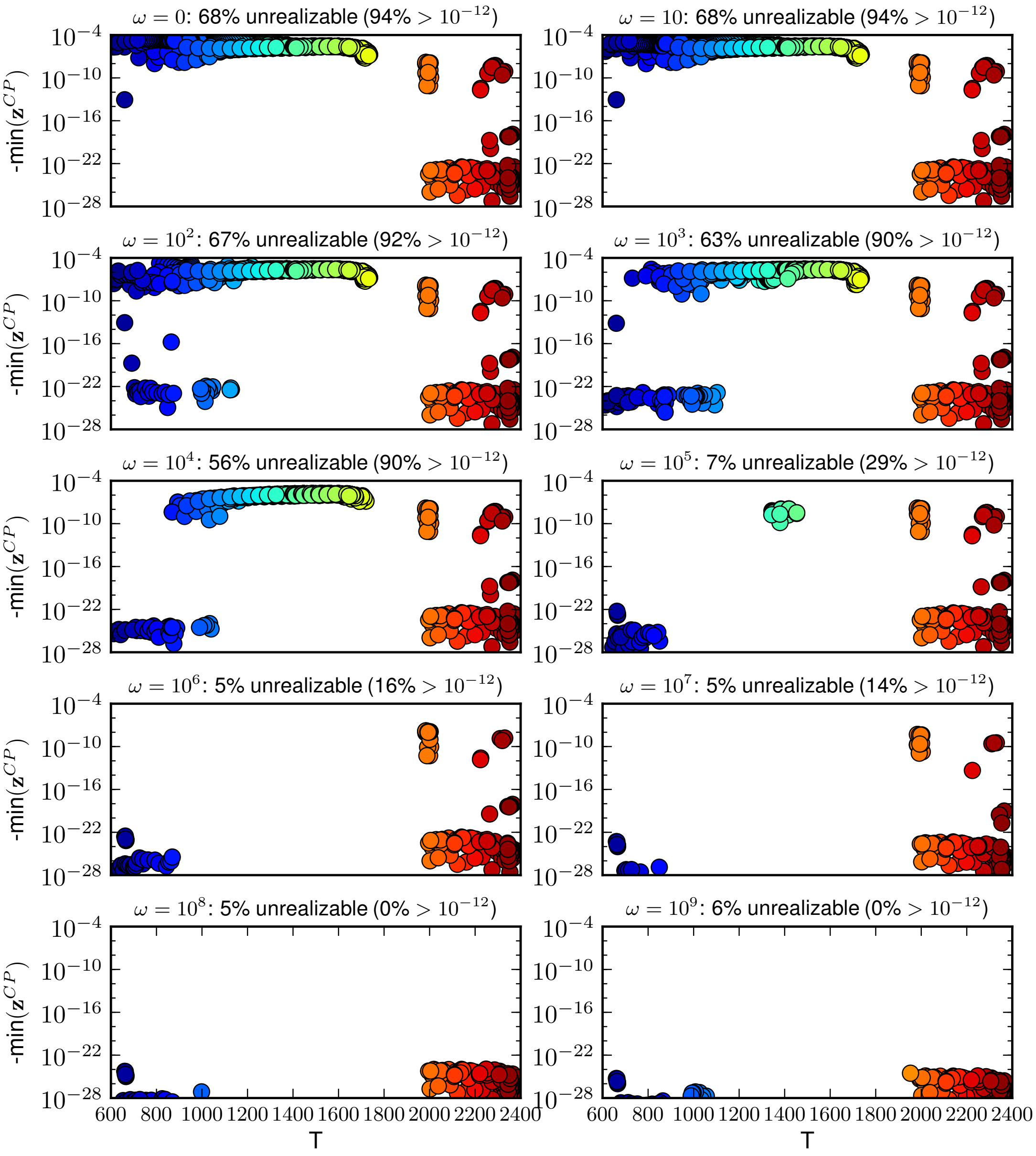


Figure 6.8: Scatter plots of the minimum species composition in $\mathbf{z}^{CP}$ (composition on the CPIM obtained using the RAMP approach) versus temperature T computed using the saved test compositions at $t = 0$ for a range of values of relaxation rate ω from 0 to 10^9 s^{-1}.

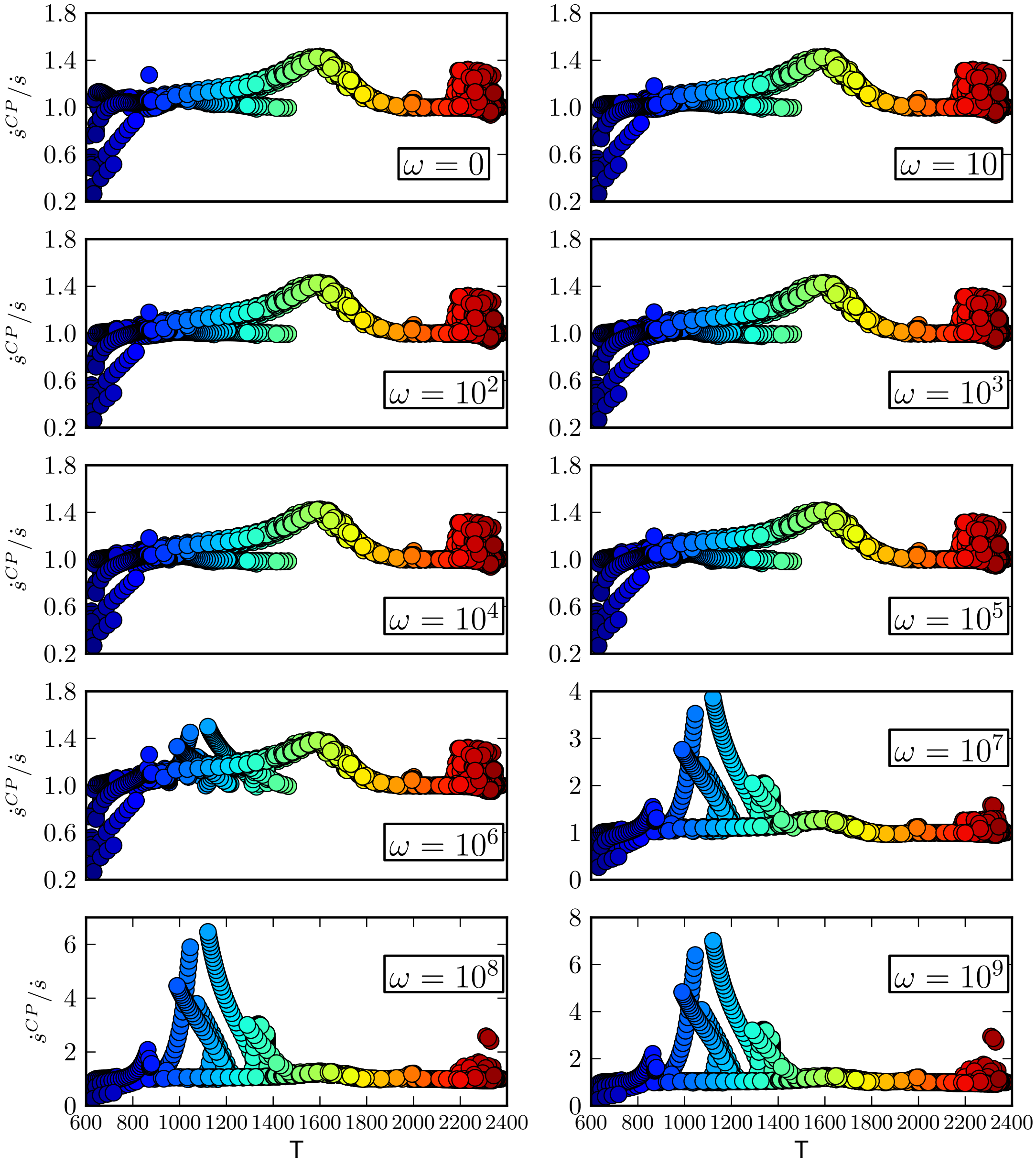


Figure 6.9: Scatter plots of ratio of entropy production rates $\dot{s}^{CP}$ (obtained using the RAMP approach) to $\dot{s}$ (at the test composition) versus temperature T computed using the saved test compositions at $t = 0$ for a range of values of relaxation rate ω from 0 to 10^9 s^{-1}.

do notice that at small values of ω and at low temperatures, the entropy production is significantly underpredicted (by about 80%) using the RAMP approach compared to the actual entropy production at the test composition. We also notice that for values of $\omega > 10^7$, the entropy production (predicted by the RAMP method) is overpredicted by 4 to 8 times in the temperature range around 1,000 K. At large values of ω, $\mathbf{z}^{CP}$ approaches $\mathbf{z}^{CE}$ and $\mathbf{S}^{CP}$ approaches $\mathbf{S}^{CE}$, which in the temperature range around 1,000 K is misaligned with the CEM and **S** (as inferred from Fig.6.6 and Fig.6.7), and this might be a possible reason for the overprediction of entropy in this temperature range. We analyzed these quantities using test compositions saved at $t = \Delta t$ as well (results not included), and observed a similar behavior.

In summary, compared to the CPIM method [90], the RAMP approach provides a good control over realizability and entropy production using the relaxation rate parameter ω.

6.8.3 Accuracy

In this section we look at three measures of error to compare the accuracy of the three implementations of RCCE. The three measures of error (which are described in more detail in the following sections) include:

1. **source vector error**: this is a measure of error in the source vector $\dot{\mathbf{r}}$ used in the RCCE and RCCE/RAMP implementations relative to the exact source vector $\dot{\mathbf{r}}^e$;
2. **reaction mapping error**: this is a measure of error in the reaction mapping $\mathbf{r}(t)$ obtained using the three implementations of RCCE, relative to the ex-

act reaction mapping $\mathbf{r}^e(t)$ obtained without using the RCCE dimension reduction;

3. **reduction-tabulation error**: this a measure of error in the reaction mapping obtained using the combined ISAT/RCCE methodology (as described in our previous work [29]) with the three implementations of RCCE.

Source Vector Error

We define the error in the source vector approximation $\dot{\mathbf{r}}(t)$ relative to the exact source vector $\dot{\mathbf{r}}^e(t)$ (as illustrated in Fig.6.5) by

$$\epsilon(\dot{\mathbf{r}}(t)) = \frac{[\dot{\mathbf{r}}(t) - \dot{\mathbf{r}}^e(t)]_{rms}}{[\dot{\mathbf{r}}^e(t)]_{rms}}, \tag{6.42}$$

where the operator $[\mathbf{x}]_{rms}$ is defined as

$$[\mathbf{x}]_{rms} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}^{(n)}\|^2}, \tag{6.43}$$

with $\|\mathbf{x}\|$ denoting the vector 2-norm. We measure the error $\epsilon(\dot{\mathbf{r}}(t))$ at two discrete times, $t = 0$ and $t = \Delta t$, using the saved test compositions.

Fig.6.10 shows the source vector error $\epsilon(\dot{\mathbf{r}}(t))$ computed using the saved test compositions at $t = 0$ and $t = \Delta t$ for the RCCE and RCCE/RAMP implementations using $n_{rs} = 11$ represented species (listed in Table 6.1). For the RAMP implementation, the error is computed for a range of values of relaxation rate ω from 0 to $10^{10}\ s^{-1}$.

At $t = 0$, we notice that both the RCCE and RCCE/RAMP implementations incur same error (around 20%) for small values of ω. We see a slight reduction in

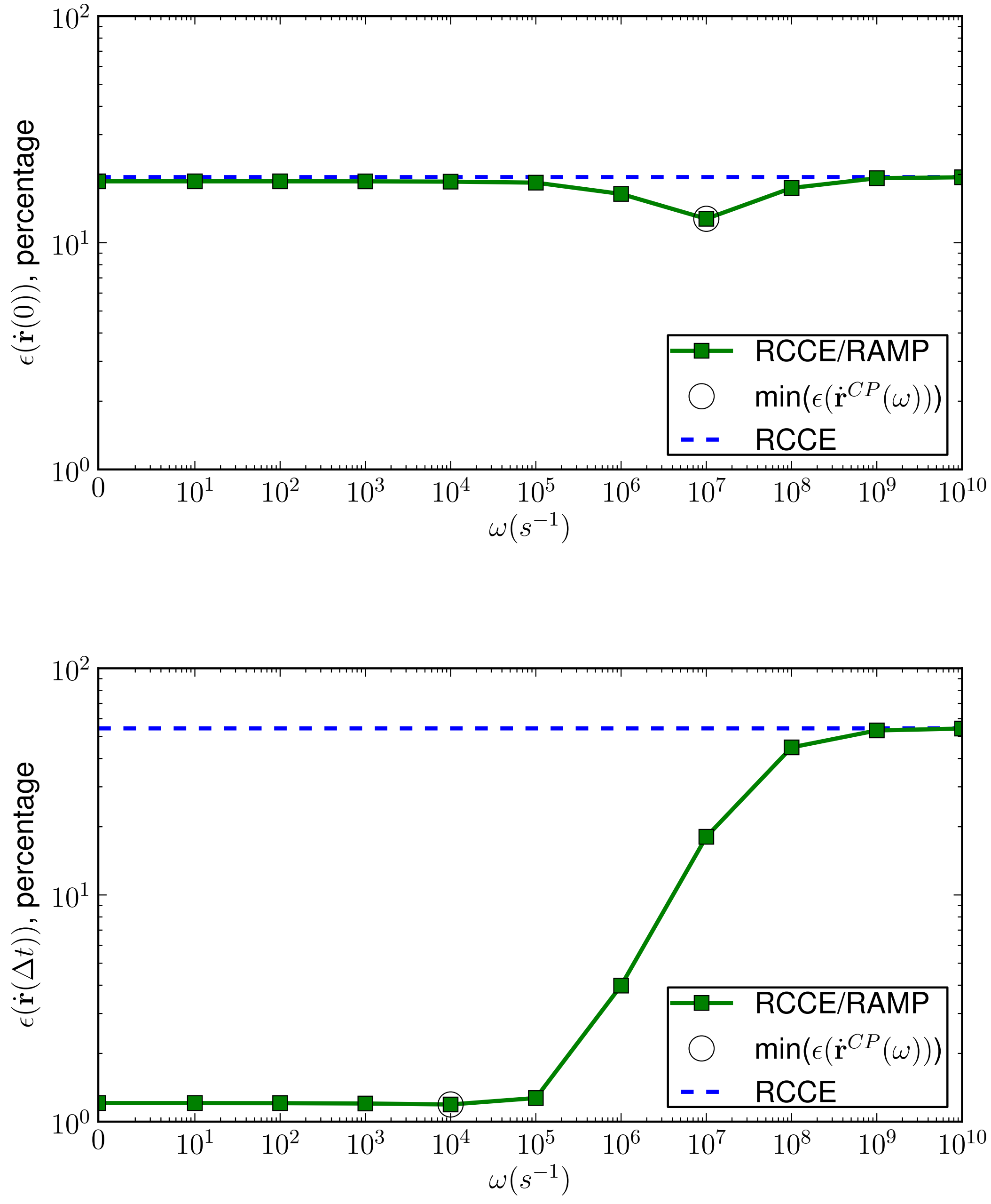
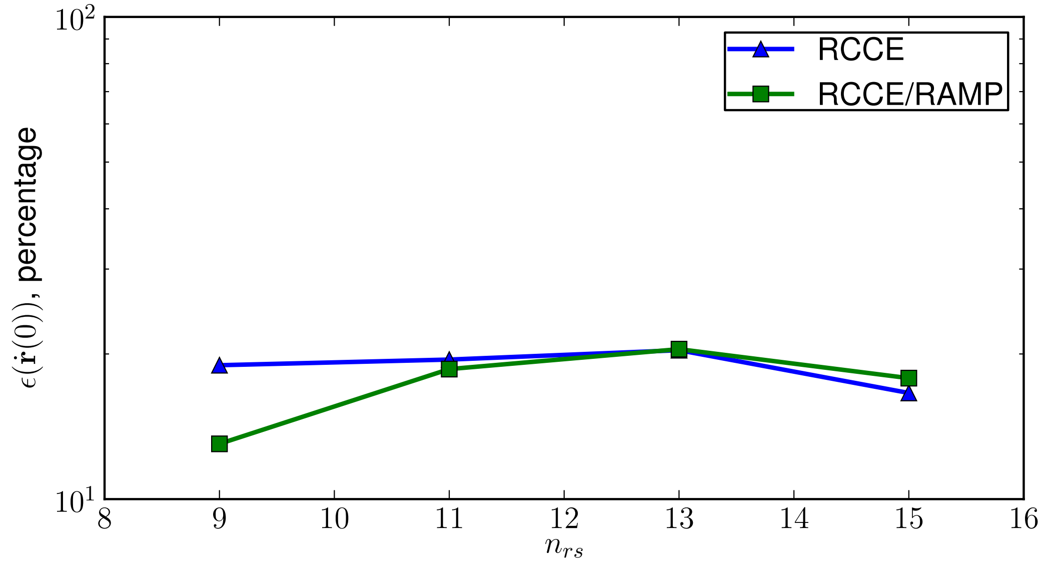


Figure 6.10: Source vector error in the RCCE and RCCE/RAMP (for various values of ω) implementations at $n_{rs} = 11$ computed using test compositions saved at $t = 0$ (top) and $t = \Delta t$ (bottom).

the source vector error using the RAMP approach (to about 11%) near $\omega = 10^7$ s^{-1}. Presumably this value of ω provides the best approximation for the test compositions at $t = 0$ using the RAMP approach. As the value of ω further increases, the error using the RAMP implementation again increases and reaches the same value as with RCCE at $\omega = 10^9$ s^{-1}.

At $t = 0$, the test compositions are away from the attracting manifold (due to mixing) and thus both RCCE and RCCE/RAMP yield a similar level of accuracy. However, at $t = \Delta t$, we notice that for small values of ω the source vector error using RCCE/RAMP is considerably smaller (around 2%) than the error incurred using RCCE (over 50%). The test compositions at $t = \Delta t$ are closer to an attracting manifold which is well approximated by the CPIM used in the RAMP approach. The RCCE method incurs large errors due to the inaccurate projection used to compute the source vector. The RAMP approach however yields a more accurate projection (using the CPIM), which in turn provides a more accurate source vector approximation as confirmed by these error measurements. We do notice that as the value of ω increases beyond 10^4 s^{-1}, the error incurred by the RCCE/RAMP approach starts increasing and approaches the same level as RCCE at $\omega = 10^9$ s^{-1}. This, as explained earlier, is because for larger values of ω, the CPIM is pulled closer to the CEM, and the RAMP approach yields the same source vector as the RCCE approach. In this case the minimum error is attained at $\omega = 10^4$ s^{-1}. We computed the source vector error for a few other values of n_{rs} (not shown), and observed a similar behavior. The RCCE/RAMP and RCCE approaches yield similar levels of error at $t = 0$, however RCCE/RAMP yields significantly lower errors at $t = \Delta t$, and for this test the minimum error is always achieved around $\omega = 10^4$ s^{-1}. For this reason, in the following tests, we use a fixed value of $\omega = 10^4$ s^{-1} in the RCCE/RAMP



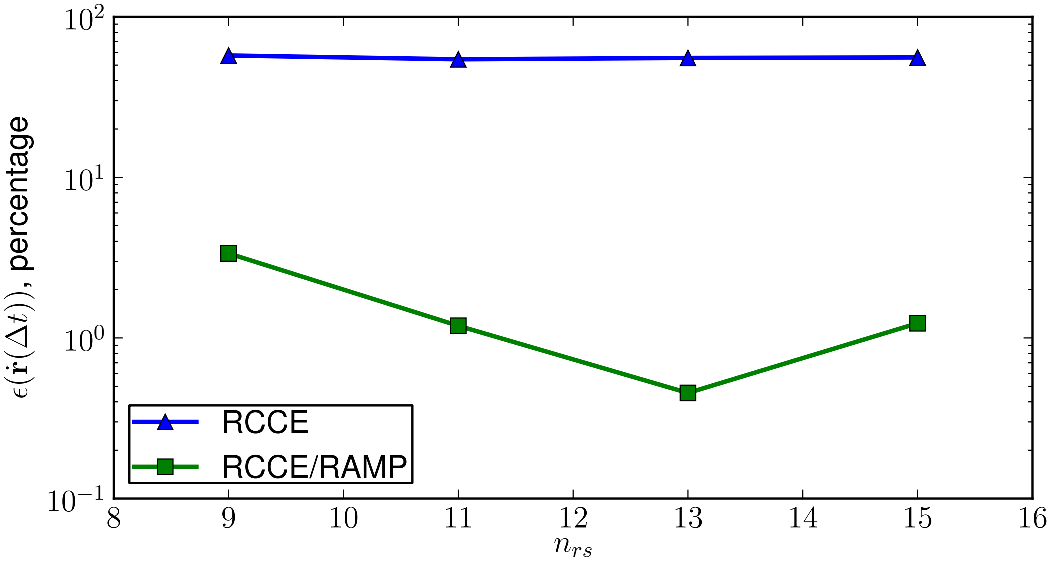


Figure 6.11: Source vector error in the RCCE and RCCE/RAMP (for fixed $\omega = 10^4$) implementations at various values of n_{rs} computed using test compositions saved at $t = 0$ (top) and $t = \Delta t$ (bottom).

implementation.

We think the value of ω which provides the best source vector approximation should be indirectly related to the flow mixing time scale, τ_{mix}. The mixing time scale determines the extent to which particle compositions can move away from the attracting manifold in the composition space. Similarly the ω value in the RAMP approach controls the position of the CPIM relative to the CEM. Hence, we think, a value of ω in the range $0.1/\tau_{mix}$ to $10/\tau_{mix}$ should provide good source vector approximation. In this work we have $\tau_{mix} = 1$ ms, which yields the good range of values of ω to be from 100 to 10,000 s^{-1}.

Fig.6.11 shows the source vector error $\epsilon(\dot{\mathbf{r}}(t))$ computed for a range of values of n_{rs} (using represented species listed in Table 6.1) at $t = 0$ and $t = \Delta t$. For the RCCE/RAMP implementation, we use a fixed value of $\omega = 10^4$ s^{-1}. Here again we see that at $t = 0$ the errors using the two implementations are comparable, however at $t = \Delta t$, RCCE/RAMP yields an order of magnitude smaller error than the RCCE.

Reaction Mapping Error

Here we compare the relative error in the reaction mapping $\mathbf{r}(t)$ obtained using different implementations of RCCE relative to the exact reaction mapping $\mathbf{r}^e(t)$ (as illustrated in Fig.6.5). For this, at each test composition $\mathbf{z} \equiv \mathbf{z}^{(n)}(0)$ we define $\mathbf{r}(0) = \mathbf{B}^T\mathbf{z}$ and then compute the reaction mapping using the following methods (as illustrated in Fig.6.4 and Fig.6.5)

- **Direct Evaluation**: the exact reaction mapping, $\mathbf{r}^e(t) = \mathbf{B}^T\mathbf{z}(t)$, where $\mathbf{z}(t)$ is obtained using ODE integration in the full space starting from $\mathbf{z}$;

- **RCCE/TIFS**: $\mathbf{r}(t) = \mathbf{B}^T\mathbf{z}(t)$, where $\mathbf{z}(t)$ is computed using ODE integration in the full space starting from $\mathbf{z}^{CE}(\mathbf{r}(0))$;
- **RCCE**: $\mathbf{r}(t)$ obtained by integrating the reduced system of ODEs (Eqn.6.7) for the constraints starting from $\mathbf{r}(0)$ with the source vector $\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CE}$; and
- **RCCE/RAMP**: $\mathbf{r}(t)$ obtained by integrating the reduced system of ODEs (Eqn.6.7) for the constraints starting from $\mathbf{r}(0)$ with the source vector $\dot{\mathbf{r}} \equiv \dot{\mathbf{r}}^{CP}$.

We define the error in the reaction mapping $\mathbf{r}(t)$ as

$$\epsilon(\mathbf{r}(t)) = \frac{[\mathbf{r}(t) - \mathbf{r}^e(t)]_{rms}}{[\mathbf{r}^e(t) - \mathbf{r}(0)]_{rms}}, \tag{6.44}$$

where the rms error is computed as before using Eqn.6.43.

Fig.6.12 shows the reaction mapping error $\epsilon(\mathbf{r}(t))$ for varying reaction time step from $t = 10^{-12}$ s to $t = 10^{-2}$ s using the three implementations of RCCE with $n_{rs} = 11$ and $n_{rs} = 15$ represented species (listed in Table 6.1). The RAMP implementation uses a fixed value of $\omega = 10^4\ s^{-1}$. For small reaction time steps, we notice that all the three implementations yield similar reaction mapping error.

From the definition of the reaction mapping error Eqn.6.44, we find that as $t \to 0$ we get

$$\lim_{t\to 0} \mathbf{r}(t) = \mathbf{r}(0) + t\dot{\mathbf{r}}(0), \quad \text{and} \quad \lim_{t\to 0} \mathbf{r}^e(t) = \mathbf{r}(0) + t\dot{\mathbf{r}}^e(0), \tag{6.45}$$

which gives

$$\lim_{t\to 0} \epsilon(\mathbf{r}(t)) = \frac{[\dot{\mathbf{r}}(0) - \dot{\mathbf{r}}^e(0)]_{rms}}{[\dot{\mathbf{r}}^e(0)]_{rms}}, \tag{6.46}$$

and hence

$$\lim_{t\to 0} \epsilon(\mathbf{r}(t)) = \epsilon(\dot{\mathbf{r}}(0)). \tag{6.47}$$

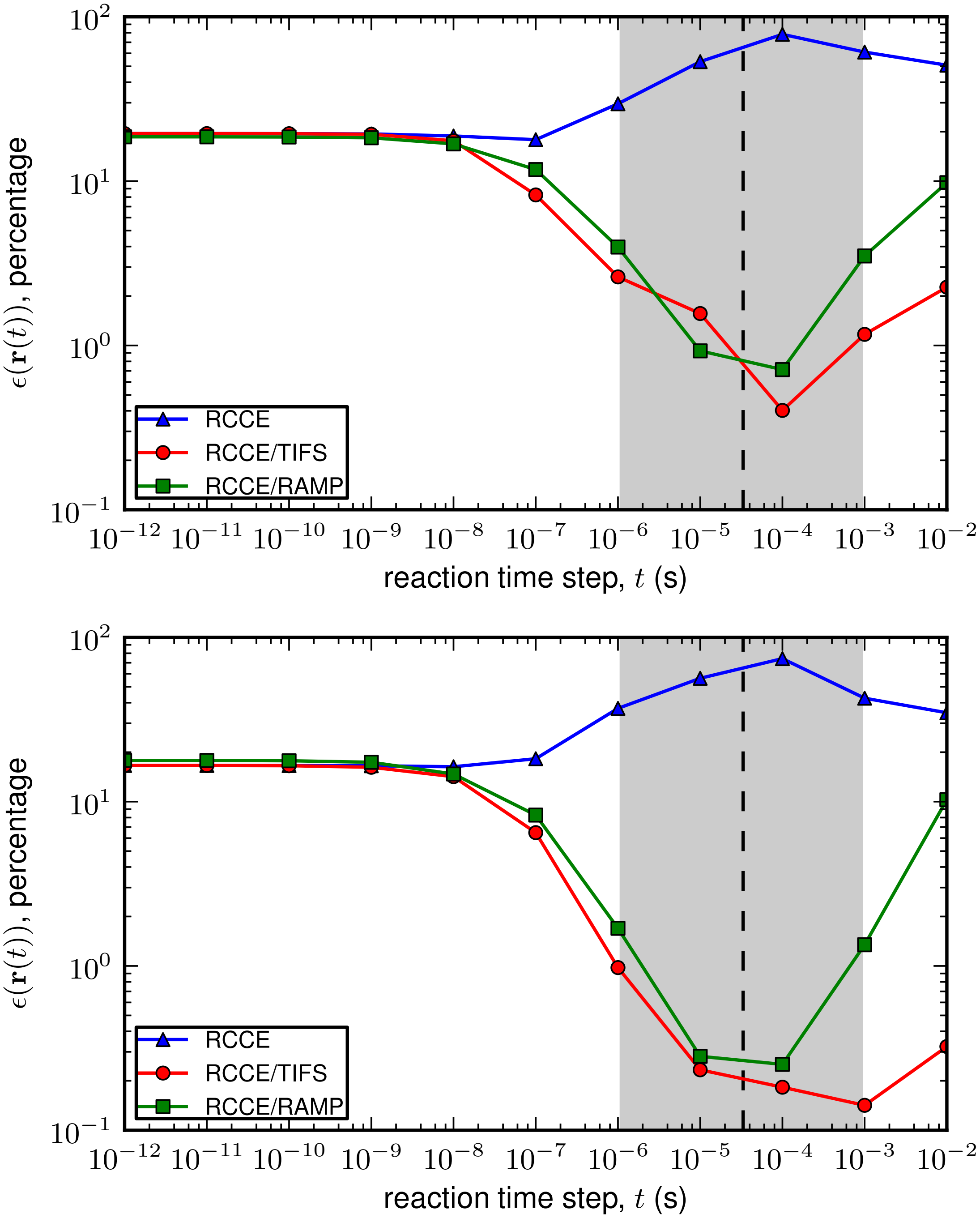


Figure 6.12: Reaction mapping error for varying reaction time step t at $n_{rs} = 11$ (top) and $n_{rs} = 15$ (bottom) using the three implementations: RCCE/TIFS, RCCE and RCCE/RAMP. The gray colored highlighted region shows the typical range of values of t (from 1 μs to 1 ms) used in real LES/PDF computations. The dashed line indicates $t = \Delta t$.

So for small reaction time steps, the reaction mapping error $\epsilon(\mathbf{r}(t))$ for the RCCE and RCCE/RAMP implementations is the same as the corresponding source vector error $\epsilon(\dot{\mathbf{r}}(0))$ at the same value of n_{rs} as seen in Fig.6.11. Also, the reaction mapping error for the RCCE/TIFS implementation is the same as the error for the RCCE implementation, since both use the chemical source term $\mathbf{S}^{CE}$ computed at $\mathbf{z}^{CE}(\mathbf{r}(0))$ for ODE integration.

At both $n_{rs} = 11$ and $n_{rs} = 15$, we notice that as the reaction time step increases, the error in the RCCE/TIFS and RCCE/RAMP implementations starts decreasing. This is because in the RCCE/TIFS implementation we solve the full system of (unconstrained) n_s ODEs which provides an accurate reaction mapping; and in the RCCE/RAMP implementation the RAMP approach provides an accurate approximation for the source vector at large time steps (as seen in Fig.6.10 and Fig.6.11). However, we notice that the error in the RCCE implementation grows or remains constant for larger reaction time steps. This is because the RCCE approach does not provide a good approximation for the source vector (as seen in Fig.6.11), and so the error in the reaction mapping grows for large reaction time steps. These results show that both RCCE/TIFS and RCCE/RAMP approaches are accurate and yield similar levels of error: however, the RCCE approach yields significantly larger errors at large reaction time steps. Fig.6.12 highlights the typical range of values of reaction time step t from 1 μs to 1 ms that we may use in real LES/PDF computations. We notice that in this range the RCCE/TIFS and RCCE/RAMP implementations both yield less than 4% and 2% errors at $n_{rs} = 11$ and $n_{rs} = 15$, respectively. However, the RCCE implementation yields over 30% error.

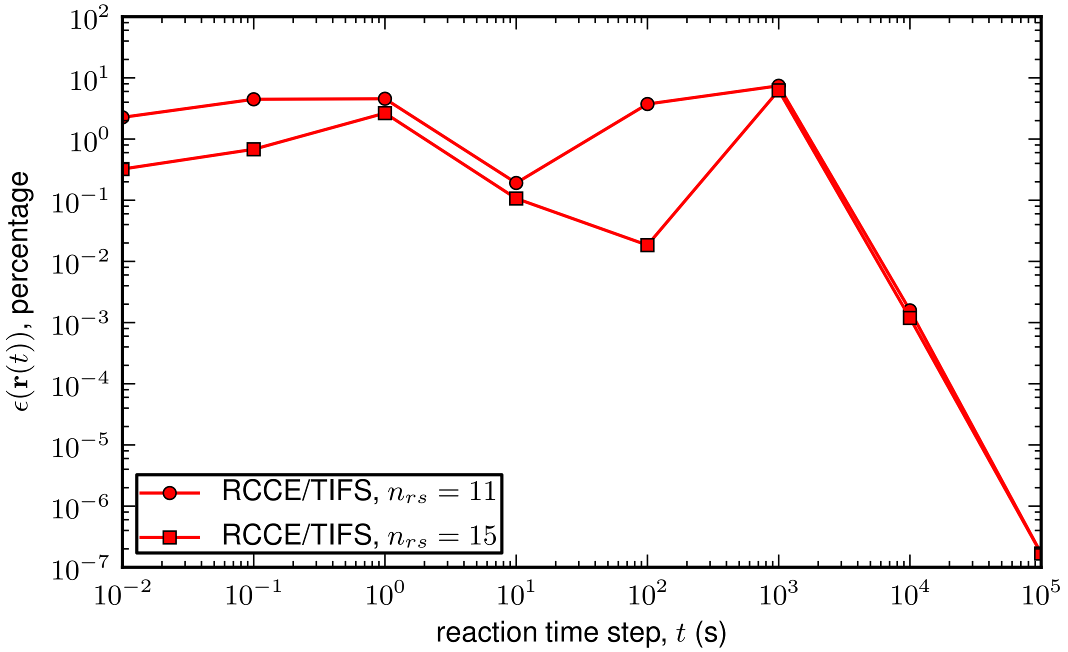


Figure 6.13: Reaction mapping error for large reaction time steps t at $n_{rs} = 11$ and $n_{rs} = 15$ using the RCCE/TIFS implementation.

We notice that the error in the RCCE/TIFS and RCCE/RAMP implementations increases for time steps beyond $t = 10^{-4}$ s. Ideally, the reaction mapping error should approach zero for large reaction time t because both $\mathbf{r}(t)$ and $\mathbf{r}^{e}(t)$ in the definition of $\epsilon(\mathbf{r}(t))$ Eqn.6.44 should reach the equilibrium composition as $t \to \infty$. The ODE integration using DDASAC becomes expensive at large reaction time steps because the solution is computed by taking many smaller substeps (as explained later in Sec.6.8.4). Thus we examine the reaction mapping error at very large reaction time steps using the RCCE/TIFS implementation alone and the results are presented in Fig.6.13. We find that, as expected, the reaction mapping error approaches zero at very large reaction time steps over 10^{3} s. The relatively large variations in the reaction mapping error for time steps beyond $t = 10^{-4}$ and the long time required for the error to reach zero can be attributed to different ignition delay times for different test compositions. Since the test compositions obtained from PaSR have a wide range of initial temperatures from 600 K to equilibrium temperature (around 2,400 K), the ignition delay time for different test compositions can vary by orders of magnitude thereby requiring a very long time for all the test compositions to reach equilibrium.

In summary these results show that both the RCCE/TIFS and RCCE/RAMP implementations yield significantly smaller error than the RCCE implementation.

Reduction-Tabulation Error

In [32], we describe our combined dimension reduction and tabulation ISAT/RCCE methodology. In this combined methodology, the reaction mapping computation using RCCE is tabulated using the ISAT algorithm to save computational time in large scale LES/PDF computations [30]. Here we measure the combined reduction-tabulation error incurred using this ISAT/RCCE methodology with RCCE implemented using the three methods described. We use the same definition of error as in our previous work [32], given as

$$\epsilon_{RT} = \frac{[\mathbf{z}^r(\Delta t) - \mathbf{z}^r_{RT}(\Delta t)]_{rms}}{[\mathbf{z}^r(\Delta t) - \mathbf{z}^r(0)]_{rms}}, \tag{6.48}$$

where $\mathbf{z}^r(0)$ and $\mathbf{z}^r(\Delta t)$ denote the composition of the represented species obtained from $\mathbf{z}(0)$ and $\mathbf{z}(\Delta t)$, respectively; and $\mathbf{z}^r_{RT}$ denotes the reaction mapping obtained using ISAT/RCCE. This error is measured for a fixed reaction time step, $t = \Delta t = 0.033$ ms, at different values of n_{rs}. The rms error is computed by considering $N = 10^5$ test compositions from a PaSR simulation (same as in [32]).

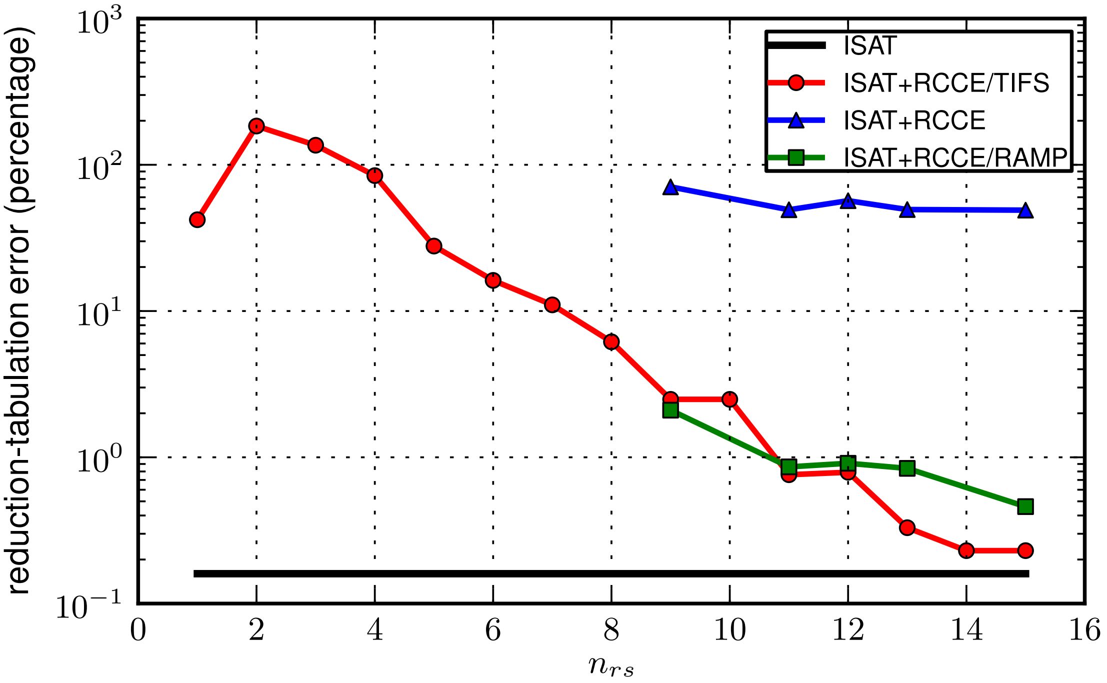


Figure 6.14: Combined reduction-tabulation error for a fixed reaction time step, $t = \Delta t$, at various values of n_{rs} using the three implementations of RCCE: RCCE/TIFS, RCCE and RCCE/RAMP with ISAT. The tabulation error (without reduction) is indicated by the solid line labeled ISAT.

Fig.6.14 shows the combined reduction-tabulation error using a fixed reaction time step $t = \Delta t$ for different values of n_{rs} for the three implementations of RCCE. (The RAMP implementation uses a fixed value of $\omega = 10^4\ s^{-1}$.) A fixed ISAT error tolerance of $\epsilon_{tol} = 10^{-5}$ is used which yields less than 1% tabulation error as shown in Fig.6.14. As we observed in the reaction mapping error results, here again we notice that the error incurred by the RCCE/TIFS and RCCE/RAMP implementations is comparable and significantly smaller than the RCCE implementation. For $n_{rs} > 9$, the RCCE/TIFS and RCCE/RAMP implementations yield less than 2% error, however the RCCE implementation yields over 50% error.

6.8.4 Computational Efficiency

In this section we compare the computational performance of the three implementations of RCCE. Each of the three implementations of RCCE involve solving a system of ODEs to compute the reaction mapping. We use DDASAC [15] to perform ODE integration with an absolute error tolerance of 10^{-8}.

Given a system of ODEs

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \tag{6.49}$$

with initial condition $\mathbf{x}(0)$ and a time duration t, DDASAC returns the solution $\mathbf{x}(t)$ within a specified error tolerance. DDASAC uses a variable time step predictor-corrector algorithm (involving backward-difference formulas (BDF) for predictor and modified Newton iterations for the corrector) to compute the solution. To solve a system of ODEs of the form Eqn.6.49 using DDASAC, we need to provide functions to compute the right-hand-side source vector $\mathbf{f}(\mathbf{x})$ and the Jacobian $\mathbf{J}(\mathbf{x})$ defined as

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}. \tag{6.50}$$

For the Jacobian, we also have the option to use DDASAC's in-built finite difference approximation for the Jacobian.

For a given initial condition $\mathbf{x}(0)$ and time duration t, DDASAC takes multiple variable time steps to compute the solution, $\mathbf{x}(t)$. At each step the source vector $\mathbf{f}(\mathbf{x})$ is evaluated, however the Jacobian is reevaluated only when needed (based on an error estimate). The number of sub-steps and Jacobian evaluations depends on the stiffness of the ODE equations and the specified error tolerance.

Considering the three implementations of RCCE, in the RCCE/TIFS implementation, we solve a system of n_s ODEs given as

$$\frac{d\mathbf{z}}{dt} = \mathbf{S}(\mathbf{z}), \tag{6.51}$$

and in the RCCE and RCCE/RAMP implementations we solve a reduced system of n_r ODEs given as

$$\frac{d\mathbf{r}}{dt} = \dot{\mathbf{r}}(\mathbf{r}), \tag{6.52}$$

where in the RCCE implementation $\dot{\mathbf{r}} = \dot{\mathbf{r}}^{CE}$ and in the RCCE/RAMP implementation $\dot{\mathbf{r}} = \dot{\mathbf{r}}^{CP}$.

For the RCCE/TIFS implementation, we provide a function generated using ADIFOR [1] to evaluate the Jacobian. However, due to the relatively complex steps involved in the evaluation of $\dot{\mathbf{r}}^{CE}$ and $\dot{\mathbf{r}}^{CP}$ in the RCCE and RCCE/RAMP implementations, respectively, using ADIFOR to generate functions for their Jacobian is not the best approach (as it results in the generation of many subfunctions). Hence we use DDASAC's in-built finite difference approximation for the Jacobian in the RCCE and RCCE/RAMP implementations. (The finite difference approximation provides a slightly less accurate Jacobian compared to ADIFOR.)

Fig.6.15 compares (on a log scale) the average cost of evaluating the source vector for the three implementations; and the overall cost of evaluating the reaction mapping using the three implementations. We see that the cost of evaluating the chemical source term **S** (source vector for the RCCE/TIFS implementation) is only around 30 μs. Compared to this the cost of evaluating the source vector for the RCCE implementation, $\dot{\mathbf{r}}^{CE}$, is around 200 μs as it involves computing the constrained-equilibrium composition $\mathbf{z}^{CE}$ which dominates the cost. The cost of evaluating the source vector for the RCCE/RAMP implementation, $\dot{\mathbf{r}}^{CP}$, goes further up to about 1,000 μs. This is because in the RAMP approach: to

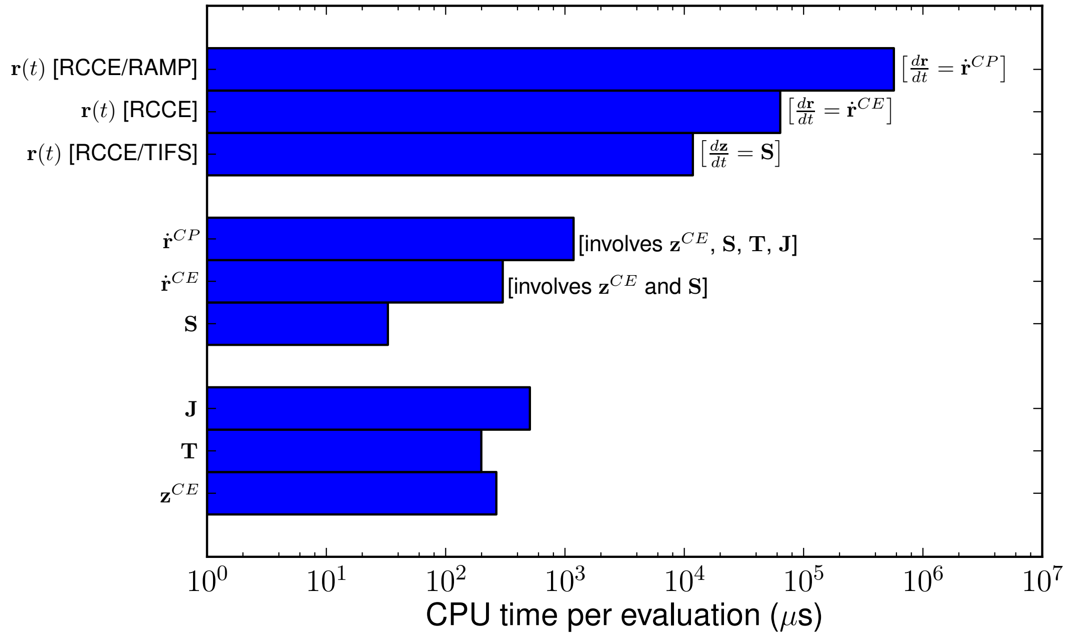


Figure 6.15: Average CPU time required to compute the reaction mapping and other quantities involved in the RCCE/TIFS, RCCE and RCCE/RAMP implementations at $n_{rs} = 11$ with the reaction time step $t = \Delta t$. The quantities shown (from bottom) include: constrained-equilibrium composition, $\mathbf{z}^{CE}$; the CEM tangent space, $\mathbf{T}$; Jacobian, $\mathbf{J}$; chemical source term, $\mathbf{S}$; source vector in RCCE implementation, $\dot{\mathbf{r}}^{CE}$; source vector in RCCE/RAMP implementation, $\dot{\mathbf{r}}^{CP}$; and finally at the top the reaction mapping $\mathbf{r}(t)$ using the three implementations.

compute the projector we need to compute the Jacobian $\mathbf{J}$ and the CEM tangent subspace $\mathbf{T}$ which are expensive to evaluate.

The cost of evaluating the reaction mapping is directly related to the cost of evaluating the source vector, and for this reason, we see that the RCCE/TIFS implementation is cheapest (around 10^4 μs) and the RCCE/RAMP implementation is the costliest (around 10^6 μs, two orders of magnitude more than RCCE/TIFS). The RCCE implementation is around five times more expensive than the RCCE/TIFS implementation.

For the RCCE implementation, there exists a mathematically equivalent alternative implementation in terms of the constraint potentials as described in [40, 27, 38, 36]. This implementation reduces the cost of evaluating the source vector by avoiding the constrained-equilibrium calculations by transforming the ODEs in terms of the constraint potentials. This implementation, however, poses some numerical issues as reported in [36], because constraint potentials can attain very large values (constraint potential value corresponding to a species with zero concentration must be infinite). Special transformation and pre-conditioning methods [36] are used to resolve these problems. This alternative implementation may be computationally less expensive, however since this implementation is mathematically equivalent to our current implementation of RCCE, the accuracy is not improved by this alternative implementation.

In summary, these results show that the RCCE/TIFS implementation is the most efficient among the three implementations of RCCE.

6.8.5 Robustness

In addition for a method to be accurate and efficient, it is also important for the method to be robust and fail-safe. In large scale LES/PDF computations, we compute the reaction mapping in the order $\mathcal{O}(10^{12})$ times, and we want our ISAT/RCCE implementation to return an accurate, realizable reaction mapping every time without failing.

In our tests with the three implementations of RCCE, we find the

RCCE/TIFS approach to be most robust. We have tested this implementation using the partially-stirred reactor for a wide range of chemical mechanisms, represented species and testing conditions, and the method has never failed [31, 32]. It has also been recently tested for performing large scale LES/PDF simulations of the Sandia Flame D [30], and again the method worked without any issues.

However, in the current study we encountered some test cases (at certain values of n_{rs} e.g., $n_{rs} = 10$ for which results are not included) where both the classical RCCE and RCCE/RAMP implementations failed to provide a realizable reaction mapping or failed to converge within DDASAC. Both these implementations solve a reduced system of ODEs for the constraints by projecting the chemical source term onto the constrained subspace. The projected source vector in some occasions is found to yield negative unrealizable constraint compositions during the DDASAC substeps. We were able to resolve some of these cases by using a smaller error tolerance in DDASAC (which in turn forces DDASAC to take smaller substeps), however, the overall implementation still failed for a few test points.

6.8.6 Comparison with Previous Works

Here we have shown that the classical RCCE implementation yields significant errors in the reaction mapping at large reaction time steps. However, many of the previous works using the classical RCCE implementation report good accuracy [27, 38, 35, 36]. There are two key differences in the implementation of RCCE used in these previous works compared to our classical RCCE implementation used in this work. In the previous works:

1. the rate-equations for the constraint potentials are solved to compute the reaction mapping; and
2. general linear combination of species compositions are used as constraints.

As pointed out earlier, the rate-equations for the constraint potentials are mathematically equivalent to the rate-equations for the constraints, and thus the two approaches should yield the same reaction mapping for similar constraints. The accuracy of the RCCE method is very sensitive to the choice of the constraints [38, 36, 31], and thus one possible reason for achieving good accuracy in these previous works could be attributed to better selection of linear constraints. We understand that the selection of linear constraints may in some cases provide more accurate results with RCCE. Ideally, it would be very insightful to test the three implementations of RCCE described in this work for linear constraints. However, our current implementation of ISAT/RCCE [32] would require significant changes to incorporate linear constraints, and hence we are unable to perform these tests at this stage. Nonetheless, it should be noted that in the previous CPIM work [90], it is shown that even for the constraint potentials based implementation using general linear constraints, the CPIM method yields better accuracy than the RCCE implementation.

Moreover, it should be noted that most of the previous works [27, 35, 36] rely more on qualitative comparison of temperature and species profiles against time to assess the accuracy of the RCCE implementation versus the detailed mechanism, which is a relatively weak test for determining the overall accuracy of the

RCCE dimension reduction method. Furthermore, to the authors' knowledge, the sensitivity of the accuracy of the RCCE implementation to reaction time step has not been studied carefully in any of the previous works.

6.9 Conclusions

In this study, we looked at three different implementation of the RCCE dimension reduction method:

1. RCCE/TIFS: involving the solution of the full system of n_s ODEs;
2. RCCE: involving the solution of a reduced system of n_r ODEs for the constraints, with source vector evaluated by projecting the chemical source term evaluated on the CEM directly onto the represented subspace;
3. RCCE/RAMP: involving an implementation similar to RCCE, however the source vector is evaluated using a more accurate RAMP approach (based on the CPIM method).

From the results presented in this work we can draw the following conclusions:

- the RCCE/TIFS implementation is the most accurate, robust and efficient among the three implementations of RCCE;
- the RCCE implementation is based on an inaccurate projection of the chemical source term onto the represented subspace, which does not take into account the non-invariance of the CEM manifold and thus yields large errors;

- the RAMP approach provides a more accurate projection for the source vector evaluation (based on the CPIM method) which significantly reduces the error;
- for the methane/air test case considered in this work, both RCCE/TIFS and RCCE/RAMP implementations yield less than 2% error in the reaction mapping compared to over 50% using the RCCE implementation;
- computationally, however, the RCCE/RAMP is an order of magnitude more expensive than the RCCE/TIFS and RCCE implementations due to the need for expensive Jacobian evaluations in the RAMP approach.

The RAMP approach, even though being expensive

- elucidates the inaccuracies in the projection employed in the classical RCCE implementation;
- demonstrates the accuracy of the CPIM approximation;
- provides an alternative framework for the implementation of RCCE (and possibly other related dimension reduction methods) based on the invariant manifold concepts.

6.10 Acknowledgements

This work is supported by Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division of the US Department of Energy (DOE) under Grant No. DE-FG02-90ER. This research was also supported in part by the National Science Foundation through TeraGrid resources provided by the Texas Advanced Computing Center under Grant No. TG-CTS090020.

CHAPTER 7

CONCLUSIONS

The main contributions of this work are the following:

1. We have developed a combined dimension reduction and tabulation methodology for the accurate and computationally-efficient representation of combustion chemistry in reacting flow computations. In this combined methodology, the dimension reduction is performed using the Rate-Controlled Constrained-Equilibrium (RCCE) method followed by tabulation using the *In Situ* Adaptive Tabulation (ISAT) algorithm. (Refer to Chapter 3.)
2. We have developed an automated Greedy Algorithm with Local Improvement (GALI) for selecting good represented species for use in the RCCE method. (Refer to Chapters 2 and 3.)
3. We have shown that our implementation of RCCE using the Trajectory In Full Space (TIFS) approach is the most accurate, efficient and robust implementation compared to some of the previous implementations of RCCE. (Refer to Chapter 6.)
4. We have developed a new Partitioned Uniform Random (P-URAN) parallel strategy using the *x2f_mpi* Fortran library for the efficient and scalable implementation of chemistry in large-scale parallel LES/PDF simulations of turbulent combustion. (Refer to Chapter 4.)
5. We have extended our LES/PDF solver with the capability of representing chemistry using our combined ISAT/RCCE approach, and have shown

that this combined algorithm works accurately and efficiently and is scalable to large number of cores using the P-URAN parallel strategy. (Refer to Chapter 5.)

The above combination enables us to perform accurate and computationally-efficient large-scale LES/PDF simulations of turbulent reacting flows with detailed chemistry involving hundreds of species. In this study we have shown that

1. the combined ISAT/RCCE methodology yields order $10^3 - 10^4$-fold speed-up relative to direct evaluation with very good error control (refer to results presented in Chapters 3, 4 and 5);
2. for performing Sandia Flame D simulations, the P-URAN strategy yields over 85% relative weak scaling efficiency and over 60% relative strong scaling efficiency on up to 9,216 cores (refer to Chapter 4);
3. compared to an LES/PDF simulation of Sandia Flame D using the simple single-scalar based Flamelet representation, the simulation using ISAT/RCCE with 15 represented variables (for a 38-species skeletal mechanism) is only 3.2 times more expensive (based on results presented in Chapters 4 and 5).

The future challenges and opportunities lie in making the above algorithms more adaptive and self-aware for performing simulations of real combustors and complex devices, where there is limited *a priori* knowledge about the flame structure and chemical reactivity. In such problems, decisions will need to be made on-the-fly to achieve good accuracy and computational performance. To this end, the following aspects of our chemistry implementation can be made more adaptive:

1. Adaptive species selection: In the current implementation we choose the represented species *a priori* using the GALI algorithm. However, for more complex problems, it will be more suitable to pick the required number of represented species on-the-fly based on a local measure of the dimension reduction error.
2. Adaptive chemistry: In addition to adaptive species selection, it is also desirable to have the flexibility of choosing (either *a priori* or on-the-fly) different sets of represented species and different chemical mechanisms to represent chemistry in different regions of the computational domain based on the local chemical reactivity. For instance, regions in coflow/air can be represented accurately using relatively fewer species and smaller mechanisms than regions in the flame front.
3. Adaptive partitioning strategy: The current simple partitioning strategy used in the P-URAN strategy uses the *a priori* knowledge about the flame structure and direction of load imbalance to form the partitions. In more complex problems we may not have sufficient knowledge about the load imbalance to form the partitions *a priori*. In such problems, it is more desirable to have an on-the-fly partitioning strategy based on the local load imbalance.

APPENDIX A

APPROXIMATION OF TEMPERATURE AND DENSITY

In this section we describe how the temperature and density are approximated using the reduced representation in the reduction-tabulation algorithm.

Approximation of Temperature

In the full composition space, given the composition $\mathbf{z} = \{\mathbf{z}^r, \mathbf{z}^u\}$ and temperature T, the enthalpy h is given as

$$\begin{aligned} h(\mathbf{z}, T) &= \mathbf{h}^T(T)\mathbf{z}, \\ &= \mathbf{h}^{r^T}(T)\mathbf{z}^r + \mathbf{h}^{u^T}(T)\mathbf{z}^u, \\ &= h^r + h^u, \end{aligned} \tag{A.1}$$

where $\mathbf{h}$ denotes molar enthalpies of species, and the superscripts r and u denote the represented and unrepresented components.

Given the full composition $\mathbf{z}$ and the total enthalpy h, the temperature T is computed using Newton's method which satisfies the following equation

$$h = h(\mathbf{z}, T). \tag{A.2}$$

However, with dimension reduction, only the composition of represented species are stored in the reduced representation, $\mathbf{r} = \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$. So, given the reduced representation $\mathbf{r}$ and temperature T, the represented part of enthalpy, h^r, can be computed exactly, but h^u needs to be approximated. Let the approximated total enthalpy, h^a, be given as

$$\begin{aligned} h^a(\mathbf{r}, T) &= \mathbf{h}^{r^T}(T)\mathbf{z}^r + \left(\mathbf{h}^{u^T}(T)\mathbf{P}\right)\left(\mathbf{E}^{u^T}\mathbf{z}^u\right) \\ &= h^r + \mathbf{h}^{u^T}(T)\mathbf{P}\mathbf{z}^{u,e}, \end{aligned} \tag{A.3}$$

where $\mathbf{E}^u$ is the constant $n_{us} \times n_e$ element matrix for unrepresented species such that $\mathbf{z}^{u,e} = \mathbf{E}^{u^T}\mathbf{z}^u$, and $\mathbf{P}$ is a specified constant $n_{us} \times n_e$ matrix. The question now to be addressed is: how best to specify $\mathbf{P}$? In the above approximation for enthalpy, we are implicitly approximating $\mathbf{z}^u$ and h^u as

$$\mathbf{z}^{u,a} \equiv \mathbf{P}\mathbf{z}^{u,e}, \tag{A.4}$$

$$h^{u,a} \equiv \mathbf{h}^{u^T}(T)\mathbf{z}^{u,a}, \tag{A.5}$$

which gives

$$h^a = h^r + h^{u,a}. \tag{A.6}$$

In the above approximation, it is important to note that

- $\mathbf{z}^{u,a}$ may not be realizable.
- $\mathbf{z}^{u,a}$ is used only to estimate h, ρ and T, and not to approximate $\mathbf{z}^u$ directly.

Note that one possible way of computing $\mathbf{z}^{u,a}$ is by performing a constrained-equilibrium calculation, but this is relatively expensive and takes $\mathcal{O}(10^3)$ μs compared to a typical query time of $\mathcal{O}(10)$ μs, and hence is avoided.

We define the approximation error in $\mathbf{z}^{u,a}$ as

$$\delta\mathbf{z}^{u,a} = \mathbf{z}^{u,a} - \mathbf{z}^u, \tag{A.7}$$

and the approximation error in enthalpy as

$$\delta h = h^a - h = h^{u,a} - h^u = \mathbf{h}^{u^T}(T)\left(\mathbf{P}\mathbf{E}^{u^T} - \mathbf{I}\right)\mathbf{z}^u. \tag{A.8}$$

For a given reduced representation, $\mathbf{r} = \{\mathbf{z}^r, \mathbf{z}^{u,e}\}$, and enthalpy, h, the approximate temperature, T^a is defined as

$$h = h^a(\mathbf{r}, T^a), \tag{A.9}$$

which is computed using Newton's method.

In the next two subsections we describe two different methods for computing the matrix $\mathbf{P}$ for approximating the enthalpy, h^a using Eqn.A.3.

Method 1

If $\mathbf{z}^u$ is not know, then the best approximation for $\mathbf{P}$ that minimizes the error (Eqn.A.8) in the approximated enthalpy, h^a, is obtained by setting $\mathbf{P}$ equal to the pseudo-inverse of $\mathbf{E}^{u^T}$. Let $\mathbf{P} = \mathbf{P}_1$ = pseudo-inverse($\mathbf{E}^{u^T}$), computed using the SVD of $\mathbf{E}^{u^T}$.

Method 2

Assuming $\mathbf{z}^u$ is known at N test points, an improved approximation for enthalpy can be computed using this known information. Let the values of $\mathbf{z}^u$ computed at these N test points be stored in an $n_{us} \times N$ matrix, $\mathbf{Z}^u$. The error in the approximated enthalpy, h^a computed at the i^{th} test point is given (in an obvious notation) as

$$\delta h_i = \mathbf{h}_i^{u^T}(T) \left(\mathbf{P}\mathbf{E}^{u^T} - \mathbf{I} \right) \mathbf{z}_i^u. \tag{A.10}$$

The vector $\mathbf{h}_i^{u^T}(T)$ is different for each test point and depends on the temperature at that point. Minimizing the overall error in the approximated enthalpy at all the test points using the above equation is not easy, and so we instead minimize the error in approximating $\mathbf{z}^u$ given as

$$\begin{aligned}\delta\mathbf{z}_i^{u,a} &= \mathbf{z}_i^{u,a} - \mathbf{z}_i^u, \\ &= \left(\mathbf{P}\mathbf{E}^{u^T} - \mathbf{I}\right)\mathbf{z}_i^u. \end{aligned} \tag{A.11}$$

Based on this error, the error in the approximated $\mathbf{z}^u$ at all the N test points is denoted by the $n_{us} \times N$ matrix $\delta\mathbf{Z}^{u,a}$ and is given as

$$\delta\mathbf{Z}^{u,a} = \mathbf{P}\mathbf{E}^{u^T}\mathbf{Z}^u - \mathbf{Z}^u \tag{A.12}$$

We compute $\mathbf{P}$ which minimizes the error, $||\delta\mathbf{Z}^{u,a}||_2$ as follows. Let the SVD of $\mathbf{Z}^u$ be given as

$$\mathbf{Z}^u = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{A.13}$$

and let the first n_e vectors of $\mathbf{U}$ be denoted by the $n_{us} \times n_e$ matrix $\mathbf{X}$. Then the matrix $\mathbf{P} \equiv \mathbf{P}_2$, which minimizes the error $||\delta\mathbf{Z}^{u,a}||_2$ (Eqn.A.12) is given as

$$\mathbf{P}_2 = \mathbf{X}(\mathbf{E}^{u^T}\mathbf{X})^{-1}. \tag{A.14}$$

So, we have two methods for computing the matrix $\mathbf{P}$ to approximate the enthalpy and thus temperature:

1. $\mathbf{P} = \mathbf{P}_1$ - easily computable, but not very accurate
2. $\mathbf{P} = \mathbf{P}_2$ - accurate but requires some stored values of $\mathbf{z}^u$

In our current implementation, we start the computations by setting $\mathbf{P} = \mathbf{P}_1$ and start storing $\mathbf{z}^u$ values as they are computed. After a certain number of $\mathbf{z}^u$ values have been stored, based on some error criterion, the matrix $\mathbf{P}$ is reset to $\mathbf{P}_2$.

Approximation of Density

Given the full composition $\mathbf{z} = \{\mathbf{z}^r, \mathbf{z}^u\}$, the density, ρ, is computed using the ideal gas law as follows:

$$\rho = \frac{p}{R_u T \sum_{i=1}^{n_s} z_i}, \tag{A.15}$$

where p is the pressure.

However, with dimension reduction, given only the reduced representation, $\mathbf{r}$, the sum $\sum_{i=1}^{n_s} z_i$ is approximated as

$$\sum_{i=1}^{n_s} z_i^a \equiv \sum_{i=1}^{n_{rs}} z_i^r + \sum_{i=1}^{n_{us}} z_i^{u,a}, \tag{A.16}$$

and the approximated density, ρ^a is given as

$$\rho^a = \frac{p}{R_u T^a \sum_{i=1}^{n_s} z_i^a}. \tag{A.17}$$

Approximation Errors

The approximation errors in temperature and density are computed by considering 10^5 test compositions from a methane/air premixed combustion in the PaSR. The results are shown in Fig.A.1 and Fig.A.2. We see that less than 1% root-mean-square relative error is measured in the approximated density and temperature relative to the exact values and those computed using the constrained-equilibrium reconstruction, which is more than two orders of magnitude more expensive than a typical ISAT query time and hence is avoided. The time spent in approximating the temperature and density using the aforementioned method is negligible compared to the ISAT query time. We also note that in general the approximation error reduces as the number of represented species, n_{rs}, is increased.

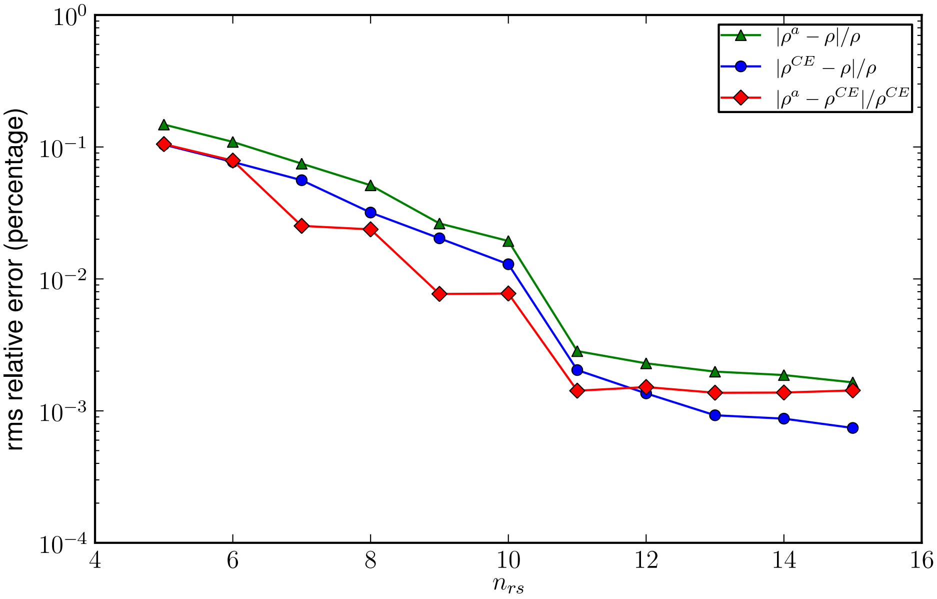


Figure A.1: The root-mean-square relative error in the approximated density, ρ^a, (computed using the RCCE reduced representation for methane/air premixed combustion at various values of n_{rs}) relative to the exact density, ρ, and the density computed using a (relatively expensive) constrained-equilibrium reconstruction, ρ^{CE}. The errors are computed by considering 10^5 test compositions in the full composition space from a methane/air premixed combustion in PaSR.

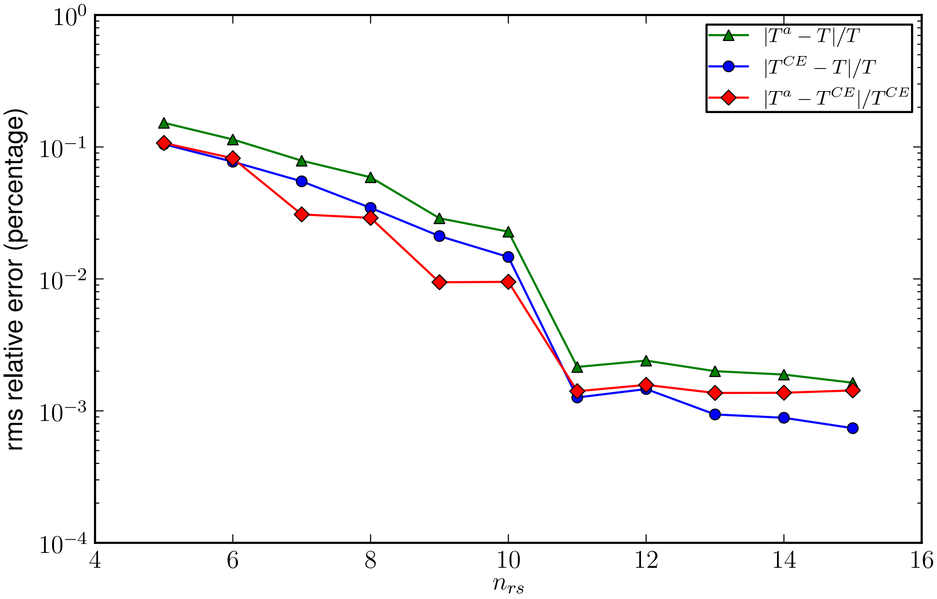


Figure A.2: The root-mean-square relative error in the approximated temperature, T^a, (computed using the RCCE reduced representation for methane/air premixed combustion at various values of n_{rs}) relative to the exact temperature, T, and the temperature computed using a (relatively expensive) constrained-equilibrium reconstruction, T^{CE}. The errors are computed by considering 10^5 test compositions in the full composition space from a methane/air premixed combustion in PaSR.

APPENDIX B

WALL CLOCK TIME STATISTICS AND ESTIMATES

In our combined LES/HPDF solver, we collect various wall clock time statistics. Here we describe the method used to estimate the breakdown of the wall clock time spent in LES, HPDF (outside reaction), Reaction and Waiting, as presented in some of the results (figures) in this work.

In our LES/HPDF solver, at each time step $n = 1$ to N_t of a simulation performed on N_c cores ranked from $c = 0$ to $N_c - 1$, we compute: the overall wall clock time spent in that time step, $t_{n,c}$; the part of the time spent within HPDF, $t^{\mathrm{P}}_{n,c}$; and, within HPDF, the part of the time spent on Reaction (including *x2f_mpi* communication), $t^{\mathrm{R}}_{n,c}$. Cumulative wall clock time statistics can then be collected on each core by summing over all the time steps as follows

$$T_c = \sum_{n=1}^{N_t} t_{n,c}, \tag{B.1}$$

and similarly for T^{P}_c and T^{R}_c. For instance, Fig.B.1 shows the cumulative wall clock time statistics collected from a Sandia Flame D simulation performed for $N_t = 2,000$ time steps on $N_c = 1,024$ cores with the chemistry represented using the 16-species mechanism with the P-URAN parallel strategy.

Now using these time statistics collected on each core, we estimate the global wall clock time statistics for the entire run as follows. The LES and HPDF solvers are synchronized (among all the cores) at the end of each time step, and so the wall clock time spent in each time step, $t_{n,c}$, and the wall clock time spent in HPDF, $t^{\mathrm{P}}_{n,c}$, are approximately the same on all the cores, and so are the cumulative times T and T^{P} (as seen in Fig.B.1). Hence, we take the time statistics from the core ranked $c = 0$ and estimate the overall wall clock time spent for the

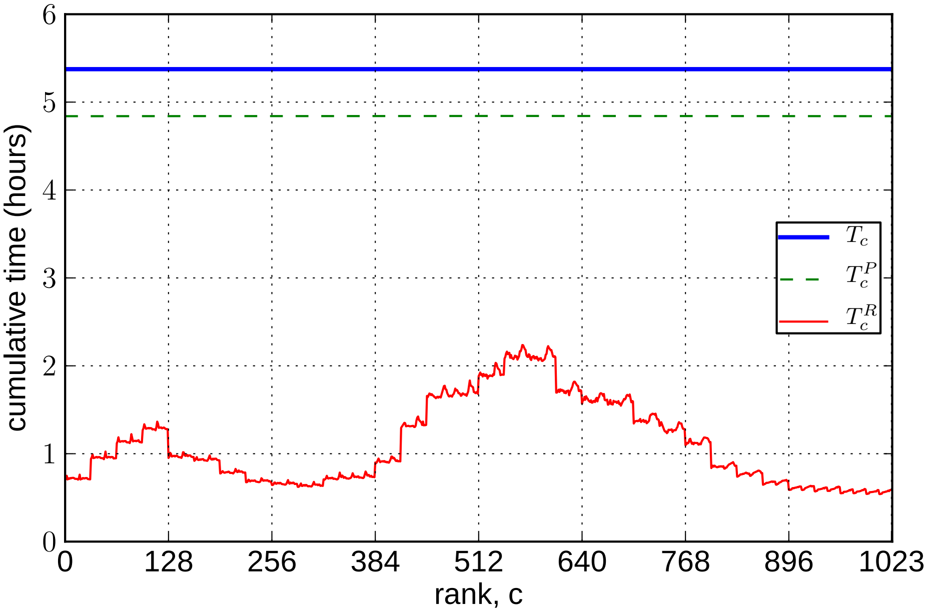


Figure B.1: LES/PDF simulation of the Sandia Flame D for $N_t = 2,000$ time steps on $N_c = 1,024$ cores with chemistry represented using the 16-species mechanism with the P-URAN[0.2h,32] parallel strategy. On each core ranked, $c = 0$ to 1,023, plotted are the cumulative wall clock time spent on the computations, T_c; the part of the time spent within HPDF, T_c^{P}; and within HPDF the part of the time spent on Reaction (including *x2f_mpi* communication), T_c^{R}.

computations as

$$T = T_0 = \sum_{n=1}^{N_t} t_{n,0}, \tag{B.2}$$

and the wall clock time spent within HPDF as

$$T^{\mathrm{P}} = T_0^{\mathrm{P}} = \sum_{n=1}^{N_t} t_{n,0}^{\mathrm{P}}, \tag{B.3}$$

which gives the wall clock time spent within LES as

$$T^{\mathrm{L}} = T - T^{\mathrm{P}}. \tag{B.4}$$

However, the wall clock time spent in Reaction, $t_{n,c}^{\mathrm{R}}$, is found to vary significantly across the cores (as seen in Fig.B.1), depending on the strategy used for implementing chemistry. So we estimate the overall wall clock time spent in Reaction (including *x2f_mpi* communication) to be the cumulative sum of the maximum reaction time (at each time step, over all the cores) as follows

$$T^{\mathrm{R}} = \sum_{n=1}^{N_t} \max_{c} \left(t^{\mathrm{R}}_{n,c} \right). \tag{B.5}$$

The time spent in HPDF (outside reaction) is then given as

$$T^{\mathrm{H}} = T^{\mathrm{P}} - T^{\mathrm{R}}, \tag{B.6}$$

and consequently we have

$$T = T^{\mathrm{L}} + T^{\mathrm{H}} + T^{\mathrm{R}}. \tag{B.7}$$

In addition, we also estimate the average (idle) Waiting time as follows. On a given time step, the “slowest” core is defined to be that which takes the greatest time for reaction. The Waiting time is the average idle time of the other cores spent waiting for the slowest core to complete, and is computed as follows

$$T^{\mathrm{W}} = \sum_{n=1}^{N_t} \frac{1}{(N_c - 1)} \sum_{c=0}^{N_c-1} \left(\max_{c'} \left(t^{\mathrm{R}}_{n,c'} \right) - t^{\mathrm{R}}_{n,c} \right). \tag{B.8}$$

Note that the Waiting time is in parallel with the Reaction time, and is indicative of the extent of reaction load imbalance. The Waiting time has a lower bound of zero indicating perfect reaction load balancing, and an upper bound equal to the Reaction time for the extreme case where the complete reaction load is concentrated on a single core at each time step.

In summary, in the figures, we plot the overall wall clock time, T; the LES time, T^{L}; the HPDF (outside reaction) time, T^{H}; the Reaction (including *x2f_mpi* communication) time, T^{R}; and the Waiting time, T^{W}.

It should be noted that typically we observe about 5% variation in the computed wall clock times (on repeated runs of our solver with identical initial conditions) due to load variations on the TACC Ranger cluster.

APPENDIX C

BEST PERFORMANCE ESTIMATES

In the results presented in Fig.4.5 and Fig.4.6, we make estimates for the lowest theoretically achievable wall clock time. These estimates are made using the following method.

For both the mechanisms, we consider the LES/PDF simulation performed on 1,024 cores using the URAN strategy (which achieves near-ideal load balancing), and find the core rank, c, with the maximum cumulative reaction time, T_c^{R}. On this core, we compute the total number of ISAT queries performed (i.e., particles resolved), N_q; the average ISAT query time, t_q, (i.e., the average time taken to resolve a particle using ISAT); and the average ISAT retrieve time, t_r, (i.e., the average time taken to retrieve a particle's reaction mapping using the ISAT table). For the 16-species mechanism, we find $t_q = 9\,\mu s$ and $t_r = 4\,\mu s$; and for the 38-species mechanism, we find $t_q = 32\,\mu s$ and $t_r = 12\,\mu s$.

Now using these data, we make two estimates for the best wall clock time for reaction:

1. Estimate (only retrieves) - estimate based on performing only local retrieves using pre-built ISAT tables. In this we estimate the reaction wall clock time on all the cores to be the same, $T^{\mathrm{R}} = T_c^{\mathrm{R}} = N_q \times t_r$.
2. Estimate (No Commun.) - estimate based on perfect load balancing with no *x2f_mpi* communication cost, while allowing for a typical fraction of direct evaluations to be performed in addition to retrieves. In this we estimate the reaction wall clock time on all the cores to be the same, $T^{\mathrm{R}} = T_c^{\mathrm{R}} = N_q \times t_q$.

APPENDIX D

OPTION TO CHECKPOINT/RE-START ISAT TABLES

In our initial implementation of the LES/PDF solver (integrated with ISAT/RCCE) [29, 30], on every re-start of a parallel simulation, the ISAT tables on all the cores are built from scratch. For small mechanisms (involving less than 30 species), the time required to build the ISAT table is often small (typically less than 1 hour). However, for larger mechanisms, the ISAT table build time can be significantly larger (e.g., 24 hours) [32].

To reduce the time spent on building ISAT tables on every re-start of a simulation, we have implemented an option in our LES/PDF solver to checkpoint the ISAT table on each core at the end of a simulation and then use these saved ISAT tables to reinitialize the ISAT tables on the next re-start of the simulation. Here we show that this method of checkpointing all the ISAT tables at the end of a simulation and using them to re-start the simulation works effectively at reducing the reaction and overall simulation time.

We demonstrate the effectiveness of checkpointing ISAT tables by performing LES/PDF simulations of the Sandia Flame D [9] with a 38-species C_1-C_4 skeletal mechanism [23]. The computational setup is the same as that described in [29], i.e., the domain size is $80D \times 30D \times 2\pi$; the LES grid size is 192×192×96; and the number of PDF particles per LES cell is $N_{pc} = 40$, which corresponds to over 141×10^6 particles in the computational domain. We perform these simulations on two core counts:

1. on 1,024 cores with a domain decomposition of 64×16 using the P-URAN[0.2h,32] parallel strategy; and

2. on 4,608 cores with a domain decomposition of 96×48 using the P-URAN[0.2h,48] parallel strategy.

(A detailed description of the P-URAN parallel strategy can be found in [29].)

On both the core counts, we first perform a simulation starting from empty ISAT tables for some N_T time steps. At the end of the simulation, we checkpoint all the ISAT tables. We then compare the performance of running the simulation for another N_T time steps by re-starting the simulation using the following two options:

1. using empty ISAT tables; and
2. using the ISAT tables saved at the end of the previous run.

On 1,024 cores we perform $N_T = 500$ time steps, which corresponds to around 6.9×10^7 ISAT queries per core; and on 4,608 cores we perform 2,000 time steps, which corresponds around 6.1×10^7 ISAT queries per core. In these simulations, we use a maximum ISAT table size per core of 500 MB (which is found to work efficiently for ISAT based on tests performed in a partially-stirred reactor) with an error tolerance of $\epsilon_{tol} = 10^{-4}$.

All the simulations are performed on the TACC Ranger cluster. A major challenge in checkpointing all the ISAT tables when performing simulation on large core counts is managing the input/output (I/O) and the storage space. In our current implementation, we checkpoint the ISAT tables only once at the end of the simulation and read the tables only once at the beginning of the simulation. This reduces the I/O time and the frequency of read/write. (Frequent I/O on large core counts can cause failures due to the limitations imposed by Metadata

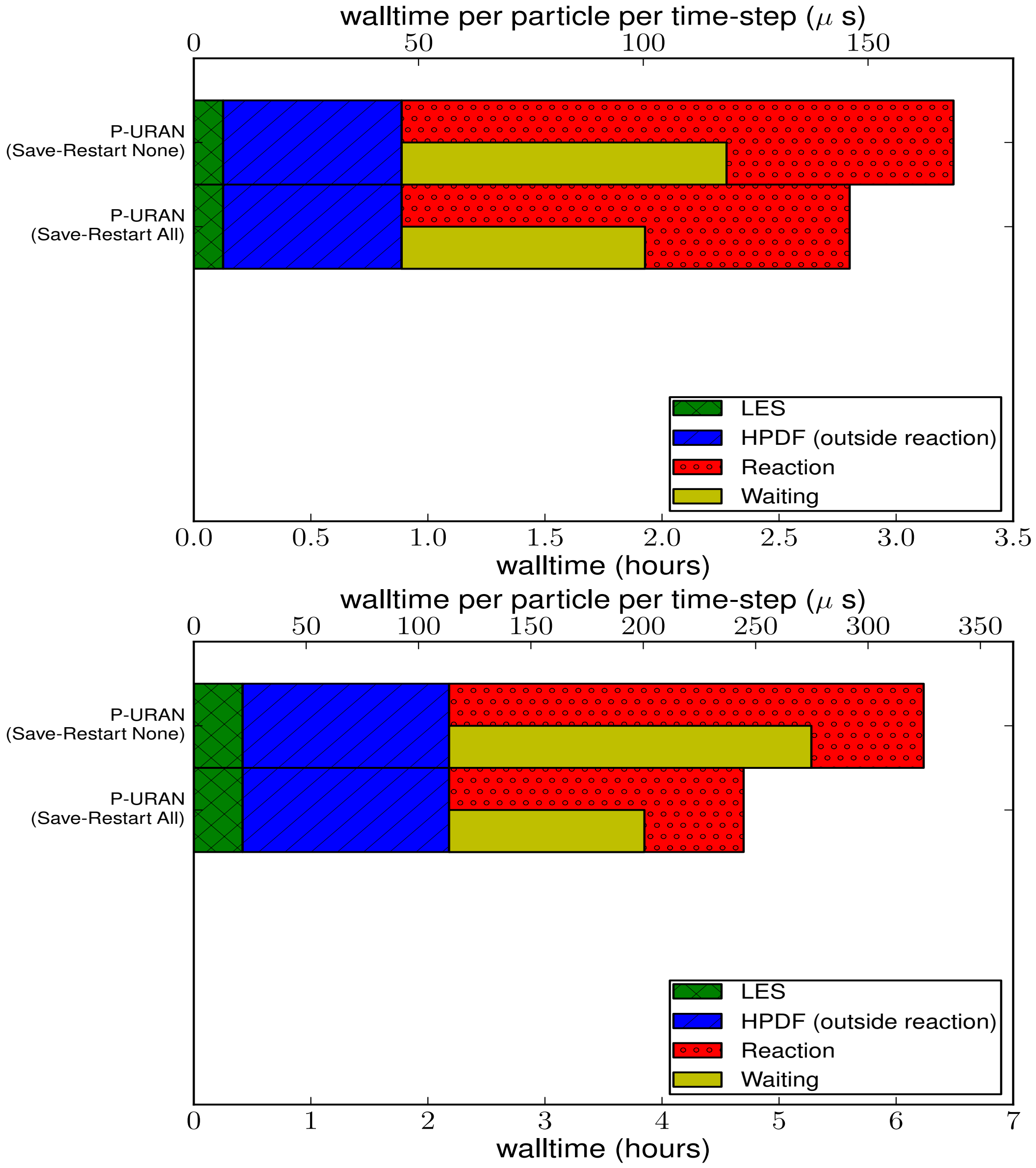


Figure D.1: Wall clock time for performing Sandia Flame D simulation on 1,024 cores for $N_T = 500$ time steps (top); and on 4,608 cores for $N_T = 2,000$ time steps (bottom) using the following two options: (i) Save-Restart None, i.e., starting the simulation using empty ISAT tables; and (ii) Save-Restart All, i.e., starting the simulation using saved ISAT tables from the previous run. In each case, the breakdown of the time spent in LES, HPDF (outside reaction), Reaction (including communication) and Waiting is shown.

Server (MDS) used on parallel clusters like TACC Ranger.) Using our current checkpoint implementation, we have not encountered any I/O or storage issues in our simulations performed on the Ranger compute cluster.

Fig.D.1 shows the overall wall clock time for performing $N_T = 500$ time steps on 1,024 cores and $N_T = 2,000$ time steps on 4,608 cores using the aforementioned two re-start strategies. On both the core counts we see a decrease in the reaction time when the simulation is started using the saved ISAT tables from the previous run. On 1,024 cores we see around 19% reduction in the reaction time, and on 4,608 cores we see around 38% reduction in the reaction time. On both the core counts, we also see a decrease in the Waiting time due to a higher percentage of ISAT queries resulting in a successful retrieve and a better load balancing achieved in the PLP stage of P-URAN. (It takes less than a couple of minutes to checkpoint and read an ISAT table of around 500 MB size, and this time is negligible in these large-scale computations.)

More reduction in the reaction time on 4,608 core count may be attributed to an indirect effect of achieving a better load balancing by the use of 96×48 domain decomposition with 96 partitions of 48 cores each, compared to a domain decomposition of 64×16 with 32 partitions of 32 cores each on 1,024 cores. In the former, more cores are handling regions near the center-line and the reaction workload is more uniformly distributed. The increase in walltime per particle per time-step when going from 1,024 cores to 4,608 cores is attributed to achieving only 60% relative strong scaling efficiency as measured in our previous scaling tests [29].

Fig.D.2 shows the percentage ISAT table filled (500 MB size) for performing $N_T = 500$ and $N_T = 2,000$ simulation time steps on 1,024 and 4,608 cores, re-

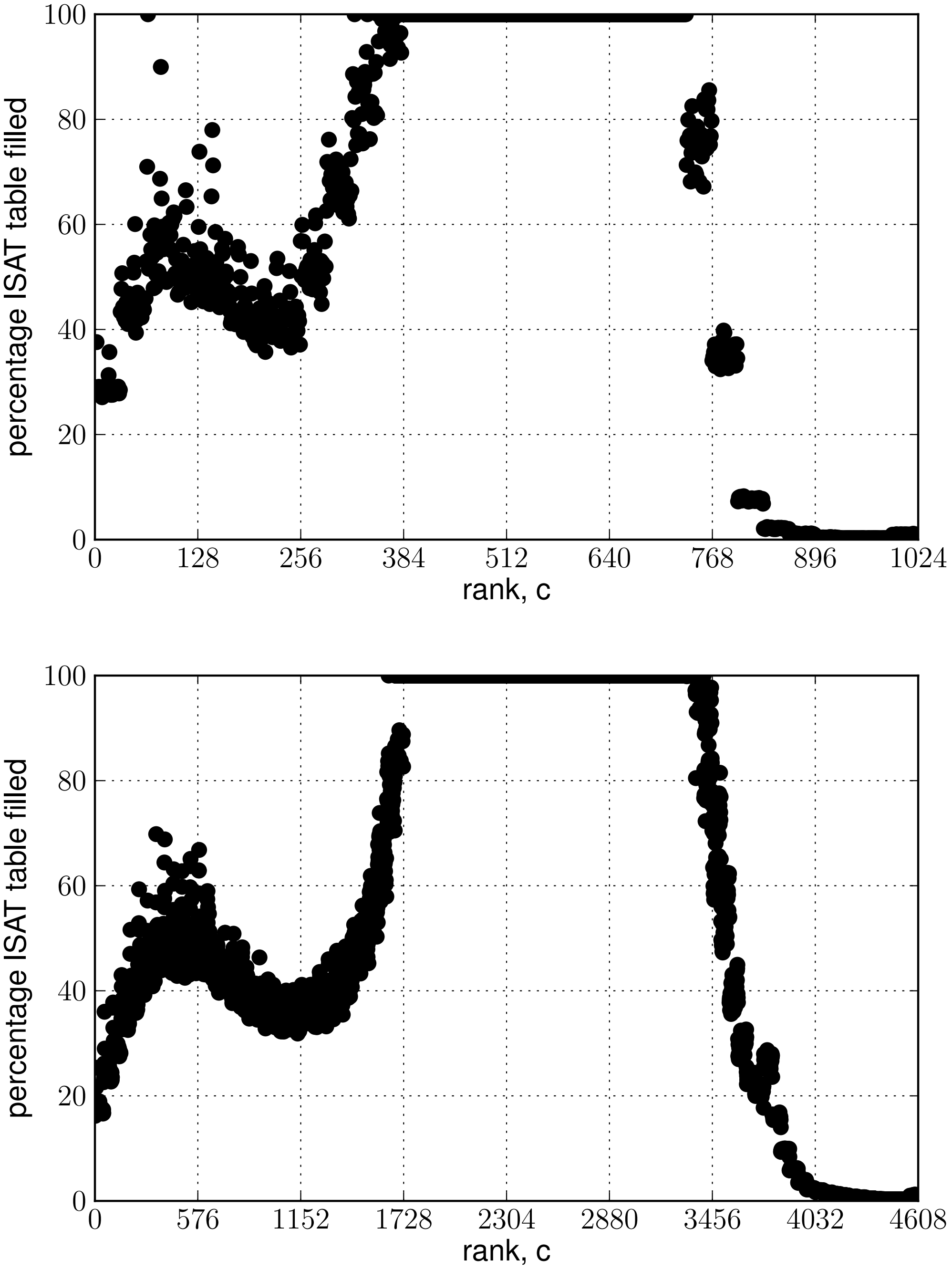


Figure D.2: Percentage ISAT table filled (500 MB size) for performing Sandia Flame D simulation on 1,024 cores for $N_T = 500$ time steps (top); and on 4,608 cores for $N_T = 2,000$ time steps (bottom).

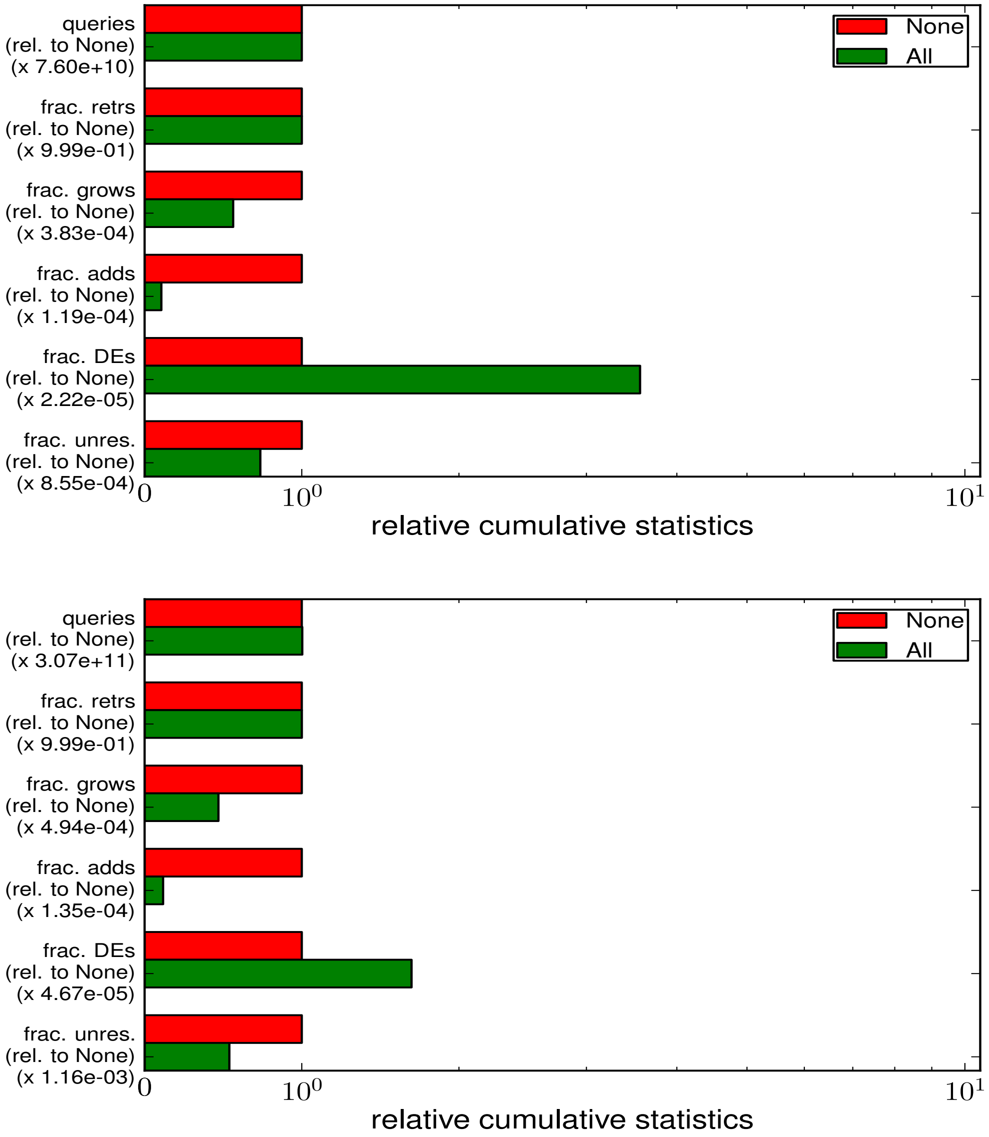


Figure D.3: Various ISAT operations performed (cumulative over all the cores) during the Sandia Flame D simulation on 1,024 cores for $N_T = 500$ time steps (top); and on 4,608 cores for $N_T = 2,000$ time steps (bottom) using the following two options: (i) None, i.e., starting the simulation using empty ISAT tables; and (ii) All, i.e., starting the simulation using saved ISAT tables from the previous run. The ISAT operations shown include (from the top): number of queries; fraction of retrieves; fraction of grows; fraction of adds; fraction of direct evaluations (DEs); and fraction of unresolved queries (during the *quick try* stage of P-URAN as explained in [29]).

spectively. We see that 100% table is filled only on the cores handling the regions in the middle of the computational domain where the fluid is most reactive due to the mixing of fuel, pilot and oxidizer streams. In the other regions, the ISAT tables are not yet completely filled.

Fig.D.3 shows a comparison of the various ISAT operations resulting from the two simulations performed using different re-start strategies on 1,024 and 4,608 cores. On both the core counts, we see that the simulation re-started using the saved ISAT tables results in significantly fewer grows and adds in ISAT and also fewer unresolved queries. This is because most ISAT queries result in a successful retrieve due to the use of ISAT tables built in the previous run. We do notice a relative increase in the direct evaluations (DEs) in the simulation re-started using the saved tables due to the use of completely filled ISAT tables (as seen in Fig.D.2) on the cores handling the regions in the middle of the domain. However, typically a grow or an add in ISAT can be three to four times more expensive than a direct evaluation [49]. Thus, the significant decrease in the fraction of adds and grows easily compensates for the cost involved in the additional DEs.

In conclusion, we have shown that re-starting an LES/PDF parallel simulation using the saved ISAT tables from the previous run can effectively help reduce the reaction and the overall simulation time. We have shown here that for performing LES/PDF simulations of Sandia Flame D with a 38-species skeletal mechanism, we have been able to reduce the reaction time by 20-40% by re-starting simulation using the saved ISAT tables from the previous run. We have also shown that our simple ISAT checkpoint implementation works without any I/O or storage issues on up to 4,608 cores.

BIBLIOGRAPHY

[1] ADIFOR 2.0 Automatic Differentiation of Fortran. http://www.mcs.anl.gov/research/projects/adifor/.

[2] BP Energy Outlook 2030, London, United Kingdom, 2012.

[3] Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change (IPCC), Geneva, Switzerland, 2007.

[4] Energy Information Administration (EIA), International Energy Outlook, September 2011.

[5] International Energy Agency (IEA), World Energy Outlook 2011, Paris, France, 2011.

[6] R. Aglave, U. Riedel, and J. Warnatz. Turbulence-chemistry interactions in cfd modelling of diesel engines. *Combustion Theory and Modelling*, 12(2):305–325, 2008.

[7] A. N. Al-Khateeb, J. M. Powers, S. Paolucci, A. J. Sommese, J. A. Diller, J. D. Hauenstein, and J. D. Mengers. One-dimensional slow invariant manifolds for spatially homogenous reactive systems. *Journal of Chemical Physics*, 131:024118 (1–19), 2009.

[8] A. Arsenlis, N. Barton, R. Becker, and R. Rudd. Generalized in situ adaptive tabulation for constitutive model evaluation in plasticity. *Computer Methods in Applied Mechanics and Engineering*, 196:1–13, 2006.

[9] R. S. Barlow and J. H. Frank. Sandia/TUD piloted CH4/Air jet flames. *Proc. Combust. Inst.*, 27:1087–1095. `http://www.sandia.gov/TNF/DataArch/FlameD.html`, 1998.

[10] G. P. Beretta, J. C. Keck, M. Janbozorgi, and H. Metghalchi. The rate-controlled constrained-equilibrium approach to far-from-local-equilibrium thermodynamics. *Entropy*, 14(2):92–130, 2012.

[11] M. Bodenstein and S. C. Lind. Geschwingigkeitder bildung des bromwasserstoffs aus seinen elementen. *Z. Phys. Chem.*, 57:168, 1906.

[12] V. Bykov and U. Maas. Extension of the ILDM method to the domain of slow chemistry. *Proceedings of the Combustion Institute*, 31:465–472, 2006.

[13] V. Bykov and U. Maas. Problem adapted reduced models based on reaction-diffusion manifolds (redims). *Proc. Combust. Inst.*, 32:561–568, 2009.

[14] R. Cao and S. Pope. The influence of chemical mechanisms on pdf calculations of nonpremixed piloted jet flames. *Combustion and Flame*, 43:450–470, 2005.

[15] M. Caracotsios and W. E. Stewart. Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations. *Computers & Chemical Engineering*, 9:359–365, 1985.

[16] J. H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C. S. Yoo. Terascale direct numerical simulations of turbulent combustion using s3d. *Computational Science & Discovery*, 2:015001, 2009.

[17] J.-Y. Chen, W. Kollmann, and R. W. Dibble. PDF modeling of turbulent methane-air nonpremixed jet flames. *Combustion Science and Technology*, 64:315–346, 1989.

[18] F. C. Christo, A. R. Masri, E. M. Nebot, and S. B. Pope. An integrated PDF/neural network approach for simulating turbulent reacting systems. *Proceedings of the Combustion Institute*, 26:43–48, 1996.

[19] F. Contino, H. Jeanmart, T. Lucchini, and G. DErrico. Coupling of in situ adaptive tabulation and dynamic adaptive chemistry: An eective method for solving combustion in engine simulations. *Proceedings of the Combustion Institute*, 33:30573064, 2011.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms. *MIT Press and McGraw-Hill*, 2001.

[21] B. I. Davidov. On the statistical dynamics of an incompressible turbulent fluid. *Dokl. Akad. Nauk SSSR*, 136:47–50, 1961.

[22] T. Echekki and E. Mastorakos, editors. *Turbulent Combustion Modeling: Advances, New Trends and Perspectives*. Springer, 2010.

[23] G. E. Esposito and H. K. Chelliah. Skeletal reaction models based on principal component analysis: Application to ethylene-air ignition, propagation, and extinction phenomena. *Combustion and Flame*, 158(3):477–489, 2011.

[24] M. Frenklach, H. Wang, C.-L. Yu, M. Goldenberg, C. Bowman, R. Hanson, D. Davidson, E. Chang, G. Smith, D. Golden, W. Gardiner, and V. Lissianski. GRI-Mech – an optimized detailed chemical reaction mechanism for methane combustion. 1995. Url: http://www.me.berkeley.edu/gri_mech/.

[25] M. Germano. Turbulence : the filtering approach. *Journal of Fluid Mechanics*, 238:325–336, 1992.

[26] A. N. Gorban and I. V. Karlin. Uniqueness of thermodynamic projector and kinetic basis of molecular individualism. *Physica A*, 336:391432, 2004.

[27] D. Hamiroune, P. Bishnu, M. Metghalchi, and J. C. Keck. Rate-controlled constrained-equilibrium method using constraint potentials. *Combustion Theory and Modelling*, 2:81–94, 1998.

[28] J. D. Hedengren and T. F. Edgar. Approximate nonlinear model predictive control with in situ adaptive tabulation. *Computers and Chemical Engineering*, 32:706–714, 2008.

[29] 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(12):3096–3109, 2012.

[30] 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. *Proceedings of the Combustion Institute*, 34:doi:10.1016/j.proci.2012.06.004, 2013.

[31] 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.

[32] V. Hiremath, Z. Ren, and S. B. Pope. Combined dimension reduction and tabulation strategy using ISAT-RCCE-GALI for the efficient implementation of combustion chemistry. *Combustion and Flame*, 158(11):2113–2127, 2011.

[33] M. Ihme and H. Pitsch. Prediction of extinction and reignition in nonpremixed turbulent flames using a flamelet/progress variable model: 2. application in les of sandia flames d and e. *Combustion and Flame*, 155:90–107, 2008.

[34] T. Ishihara, Y. Kaneda, M. Yokokawa, K. Itakura, and A. Uno. Small-scale statistics in high-resolution direct numerical simulation of turbulence: Reynolds number dependence of one-point velocity gradient statistics. *Journal of Fluid Mechanics*, 592:335–366, 2007.

[35] M. Janbozorgi and H. Metghalchi. Rate-controlled constrained-equilibrium theory applied to the expansion of combustion products in the power stroke of an internal combustion engine. *Int. J. of Thermodynamics*, 12:44–50, 2009.

[36] M. Janbozorgi, S. Ugarte, H. Metghalchi, and J. C. Keck. Combustion modeling of mono-carbon fuels using the rate-controlled constrained-equilibrium method. *Combustion and Flame*, 156:1871–1885, 2009.

[37] J. Janicka, W. Kolbe, and W. Kollmann. Closure of the transport equation for the PDF of turbulent scalar fields. *Journal of Non-Equilibrium Thermodynamics*, 4:47–66, 1970.

[38] W. Jones and S. Rigopoulos. Rate-controlled constrained equilibrium: Formulation and application to nonpremixed laminar flames. *Combustion and Flame*, 142(3):223 – 234, 2005.

[39] W. P. Jones and B. E. Launder. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 15:301–314, 1972.

[40] J. C. Keck. Rate-controlled constrained equilibrium theory of chemical reactions in complex systems. *Prog. Energy Combust. Sci.*, 16:125–154, 1990.

[41] J. C. Keck and D. Gillespie. Rate-controlled partial equilibrium method for treating reacting gas-mixtures. *Combust. Flame*, 17:237–241, 1971.

[42] P. Kloeden and E. Platen. Numerical solution of stochastic differential equations. *Springer Verlag Berlin*, 1992.

[43] A. Kumar and S. Mazumder. Adaptation and application of the in situ adaptive tabulation (ISAT) procedure to reacting flow calculations with

complex surface chemistry. *Computers & Chemical Engineering*, 35(7):1317–1327, 2011.

[44] S. H. Lam and D. A. Goussis. The CSP method for simplifying kinetics. *International Journal of Chemical Kinetics*, 26:461–486, 1994.

[45] M. Lesieur and O. Metais. New trends in large-eddy simulations of turbulence. *Annu. Rev. Fluid Mech*, 28:4582, 1996.

[46] L. Liang, J. Stevens, and J. Farrel. A dynamic adaptive chemistry scheme for reactive ow computations. *Proceedings of the Combustion Institute*, 32:527534, 2009.

[47] D. K. Lilly. A proposed modification of the germano subgridscale closure method. *Physics of Fluids*, 4(3):633, 1992.

[48] L. Lu, S. R. Lantz, Z. Ren, and S. B. Pope. Computationally efficient implementation of combustion chemistry in parallel PDF calculations. *Journal of Computational Physics*, 228(15):5490–5525, 2009.

[49] L. Lu and S. B. Pope. An improved algorithm for in situ adaptive tabulation. *Journal of Computational Physics*, 228(2):361–386, 2009.

[50] L. Lu, Z. Ren, S. R. Lantz, V. Raman, S. B. Pope, and H. Pitsch. Investigation of strategies for the parallel implementation of ISAT in LES/FDF/ISAT computations. *Proceedings of the 4th Joint Meeting of US Sections of the Combustion Institute, Philadelphia, PA*, 2005.

[51] L. Lu, Z. Ren, V. Raman, S. B. Pope, and H. Pitsch. LES/FDF/ISAT computations of turbulent flames. *Proceedings of the CTR Summer Program*, pages 283–294, 2004.

[52] T. Lu, Y. Ju, and C. K. Law. Complex CSP for chemistry reduction and analysis. *Combustion and Flame*, 126:1445–1455, 2001.

[53] T. Lu and C. K. Law. A directed relation graph method for mechanism reduction. *Proc. Combust. Inst.*, 30:1333–1341, 2005.

[54] T. Lvs, S. Navarro-Martinez, and S. Rigopoulos. On adaptively reduced chemistry in large eddy simulations. *Proceedings of the Combustion Institute*, 33(1):1339–1346, 2011.

[55] U. Maas and S. B. Pope. Simplifying chemical-kinetics: intrinsic low-dimensional manifolds in composition space. *Combust. Flame*, 88:239–264, 1992.

[56] 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*, 154:342371, 1999.

[57] R. Mustata, L. Valio, C. Jimnez, W. Jones, and S. Bondi. A probability density function eulerian monte carlo field method for large eddy simulations: Application to a turbulent piloted methane/air diffusion flame (sandia d). *Combustion and Flame*, 145:88–104, 2006.

[58] T. Nagy and T. Turanyi. Reduction of very large reaction mechanisms using methods based on simulation error minimization. *Combustion and Flame*, 156:417–428, 2009.

[59] M. B. Nik, S. L. Yilmaz, P. Givi, M. R. H. Sheikhi, and S. B. Pope. Simulation of sandia flame d using velocityscalar filtered density function. *American Institute of Aeronautics and Astronautics*, 48:1513–1522, 2010.

[60] P. Pepiot-Desjardins and H. Pitsch. An efficient error-propagation-based reduction method for large chemical kinetic mechanisms. *Combustion and Flame*, 154:67–81, 2008.

[61] N. Peters. *Turbulent Combustion*. Cambridge University Press, 2000.

[62] C. D. Pierce. *Progress-variable approach for Large-Eddy Simulation of turbulent combustion*. PhD thesis, Stanford University, 2001.

[63] 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.

[64] W. J. Pitz, N. P. Cernansky, F. L. Dryer, F. N. Egolfopoulos, J. T. Farrell, D. G. Friend, and H. Pitsch. Development of an experimental database and kinetic models for surrogate gasoline fuels. *Society of Automotive Engineers*, SAE-2007-01-0175, 2007.

[65] S. B. Pope. PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science*, 11:119–192, 1985.

[66] S. B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combust. Theory Model.*, 1:41–63, 1997.

[67] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[68] S. B. Pope. The computation of constrained and unconstrained equilibrium compositions of ideal gas mixtures using Gibbs function continuation. *Cornell University report FDA 03-02*, 2003.

[69] S. B. Pope. Gibbs function continuation for the stable computation of chemical equilibrium. *Combustion and Flame*, 139:222–226, 2004.

[70] S. B. Pope. Simple models of turbulent flows. *Physics of Fluids*, 23(1):011301, 2011.

[71] S. B. Pope. Small scales, many species and the manifold challenges of turbulent combustion. *Proceedings of the Combustion Institute*, 34:doi:10.1016/j.proci.2012.09.009, 2013.

[72] S. B. Pope and U. Mass. Simplifying chemical kinetics: trajectory-generated low-dimensional manifolds. *Mechanical and Aerospace Engineering Report*, FDA 93-11. Cornell University, Ithaca, 1993.

[73] R. Rabenseifner, G. Hager, and G. Jost. Hybrid mpi/openmp parallel programming on clusters of multi-core smp nodes. *Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2009, Weimar, Germany, 18-20 Febuary 2009, IEEE Computer Society*, pages 427–436, 2009.

[74] V. Raman, R. O. Fox, and A. D. Harvey. Hybrid finite-volume/transported pdf simulations of a partially premixed methane-air flame. *Combustion and Flame*, 136:327–350, 2004.

[75] Z. Ren and G. M. Goldin. An efficient time scale model with tabulation of chemical equilibrium. *Combustion and Flame*, 158(10):1977–1979, 2011.

[76] Z. Ren, G. M. Goldin, V. Hiremath, and S. B. Pope. Reduced description of reactive flows with tabulation of chemistry. *Combustion Theory and Modelling*, 15(6):827–848, 2011.

[77] Z. Ren, V. Hiremath, and S. B. Pope. Dimension reduction and tabulation of combustion chemistry using ICE-PIC and ISAT. *6th US National Combustion Meeting, May 17-20, 2009, Ann Arbor, Michigan.*, 2009.

[78] Z. Ren, V. Hiremath, and S. B. Pope. ice-pic: "a library of Fortran 90 routines for using the dimension reduction methods RCCE and ICE-PIC with ISAT". pages http://eccentric.mae.cornell.edu/∼tcg/ice–pic, 2009.

[79] Z. Ren, S. B. Pope, A. Vladimirsky, and J. M. Guckenheimer. The invariant constrained equilibrium edge pre-image curve method for the dimension reduction of chemical kinetics. *J. Chem. Phys.*, 124:Art. No. 114111, 2006.

[80] S. Rigopoulos. The rate-controlled constrained equilibrium (RCCE) method for reducing chemical kinetics in systems with time-scale separation. *International Journal for Multiscale Computational Engineering*, 5:11–19, 2007.

[81] M. Sandip. Modeling full-scale monolithic catalytic converters: Challenges and possible solutions. *Journal of Heat Transfer*, 129:526–535, 2007.

[82] S. Sarathy, C. Westbrook, M. Mehl, W. Pitz, C. Togbe, P. Dagaut, H. Wang, M. Oehlschlaeger, U. Niemann, K. Seshadri, P. Veloo, C. Ji, F. Egolfopoulos, and T. Lu. Comprehensive chemical kinetic modeling of the oxidation of 2-methylalkanes from c7 to c20. *Combustion and Flame*, 158(12):2338–2357, 2011.

[83] S. Shafiee and E. Topal. When will fossil fuel reserves be diminished? *Energy Policy*, 37(1):181–189, 2009.

[84] D. A. Sheen, X. You, H. Wang, and T. Lovas. Spectral uncertainty quantification, propagation and optimization of a detailed kinetic model for ethylene combustion. *Proceedings of the Combustion Institute*, 32:535–542, 2009.

[85] M. Sheikhi, T. Drozda, P. Givi, F. Jaberi, and S. Pope. Large eddy simulation of a turbulent nonpremixed piloted methane jet flame (sandia flame d). *Proceedings of the Combustion Institute*, 30:549–556, 2005.

[86] 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.

[87] M. D. Smooke. Reduced kinetic mechanisms and asymptotic approximations for methane-air flames. *Springer-Verlag, Berlin.*, 1991.

[88] C. J. Sung, C. K. Law, and J.-Y. Chen. An augmented reduced mechanism for methane oxidation with comprehensive global parametric validation. *Symposium (International) on Combustion*, 27:295–304, 1998. Url: http://www.princeton.edu/∼cklaw/kinetics/12-step/index.html.

[89] Q. Tang and S. B. Pope. Implementation of combustion chemistry by in situ adaptive tabulation of rate-controlled constrained equilibrium manifolds. *Proceedings of the Combustion Institute*, 29:1411–1417, 2002.

[90] Q. Tang and S. B. Pope. A more accurate projection in the rate-controlled constrained-equilibrium method for dimension reduction of combustion chemistry. *Combust. Theory Modelling*, 8:255–279, 2004.

[91] Q. Tang, J. Xu, and S. B. Pope. PDF calculations of local extinction and NO production in piloted-jet turbulent methane/air flames. *Proceedings of the Combustion Institute*, 28:133–139, 2000.

[92] S. R. Tonse, N. W. Moriarity, N. J. Brown, and M. Frenklach. PRISM: Piecewise Reusable Implementation of Solution Mapping. An economical strategy for chemical kinetics. *Israel J. Chem.*, 39:97–106, 1999.

[93] T. Turanyi. Parameterization of reaction mechanisms using orthonormal polynomials. *Computers Chem.*, 18:45–54, 1994.

[94] 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*, 146:29–51, 2006.

[95] S. Viswanathan, H. Wang, and S. 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.

[96] H. Wang, M. Juddoo, S. Starner, A. Masri, and S. Pope. A novel transient turbulent jet flame for studying turbulent combustion. *Proceedings of the Combustion Institute*, 34, 2013.

[97] H. Wang and S. B. Pope. Large Eddy Simulation/Probability Density Function modeling of a turbulent CH4/H2/N2 jet flame. *Proceedings of the Combustion Institute*, 33:1319–1330, 2011.

[98] H. Wang, P. P. Popov, and S. B. Pope. Weak second order splitting schemes for lagrangian monte carlo particle methods for the composition PDF/FDF transport equations. *Journal of Computational Physics*, 229:1852–1878, 2010.

[99] H. Wang, X. You, A. V. Joshi, S. G. Davis, A. Laskin, F. Egolfopoulos, and C. K. Law. USC Mech Version II. high-temperature combustion reaction model of H2/CO/C1-C4 compounds. 2007. Url: http://ignis.usc.edu/USC_Mech_II.htm.

[100] 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. *Combust. Flame*, 156:181–199, 2009.

[101] J. Xu and S. B. Pope. Pdf calculations of turbulent nonpremixed flames with local extinction. *Combustion and Flame*, 123:281–307, 2000.

[102] Y. Yang, H. Wang, S. Pope, and J. Chen. Large-eddy simulation/probability density function modeling of a nonpremixed co/h2 temporally evolving jet flame. *Proceedings of the Combustion Institute*, 34, 2013.

[103] A. C. Zambon and H. K. Chelliah. Explicit reduced reaction models for ignition, flame propagation, and extinction of C2H4/CH4/H2 and air systems. *Combustion and Flame*, 150:71–91, 2007.