Coupled Time Integration and Globalized Newton's Method for Faster Solution of Chemical Reactor Networks

Pratik Sutar

Master Thesis Project

Delft University of Technology

Faculty of Aerospace Engineering

COUPLED TIME INTEGRATION AND GLOBALIZED NEWTON'S METHOD FOR FASTER SOLUTION OF CHEMICAL REACTOR NETWORKS

by

Pratik Sutar

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Thursday October 26, 2021 at 14:00.

Supervisor:	Dr. Arvind Gangoli Rao Dr. Domenico Lahaye Rishikesh Sampat, MSc
Thesis committee:	Dr. Arvind Gangoli Rao, TU Delft Dr. Domenico Lahaye, TU Delft Dr. Dirk Roekaerts, TU Delft

CONTENTS

Abstract **vii**

List of Figures **ix**

List of Tables **xv**

Nomenclature **xvii**

1 Introduction **1**

1.1 Background 1
1.2 Chemical Kinetics 2
1.3 Reaction Mechanisms 3
  1.3.1 GRI-Mech 3.0 3
1.4 Emissions 3
  1.4.1 Carbon Monoxide (CO) 4
  1.4.2 Nitrogen Oxides (NO_X) 4
1.5 Transport Processes 6
1.6 Combustion Modeling 7
  1.6.1 Computational Flow Dynamics (CFD) 8
  1.6.2 Chemical Reactor Network (CRN) 9
  1.6.3 Hybrid CFD-CRN 10
1.7 Past Developments in CFD-CRN 11
1.8 AGNES 16
  1.8.1 Research Possibilities in AGNES 20
1.9 Research Outline 21
1.10 Overview 22

2 Chemical Reactor Network Solver **23**

2.1 Building Reactor Network: Cantera 23
  2.1.1 Boundary Conditions 25
  2.1.2 Mass Imbalance Correction 26
2.2 Governing Equations 26
2.3 Jacobian Calculation 30
  2.3.1 Only Transport Contributions with Energy OFF 30
  2.3.2 Only Reaction Contributions with Energy OFF 31
  2.3.3 Energy ON 33

2.4 PETSc: ODE and SNES Solver . . . 34
  2.4.1 Index Sets . . . 37
  2.4.2 Vectors . . . 37
  2.4.3 Matrices . . . 38
  2.4.4 Linear System Solvers . . . 39
  2.4.5 Nonlinear System Solvers . . . 42
  2.4.6 Time-stepping Schemes . . . 44
2.5 Solver Implementation . . . 47
  2.5.1 AGNES v1.1 Solver Implementation . . . 49
  2.5.2 AGNES v1.2.1: KPPSMOKE Approach . . . 51
  2.5.3 AGNES v1.2.2: Only Time Integration (TS) . . . 53
  2.5.4 AGNES v1.2.3: Time Integration and Globalised Newton's Method . . . 55

3 CFD Test Case: Sandia Flame D **57**
3.1 Sandia Flames . . . 57
3.2 Experimental Results . . . 59
3.3 CFD Simulation . . . 60
3.4 CRN Setup . . . 61

4 Results and Discussion **63**
4.1 Single Reactor . . . 63
4.2 Multiple Reactors . . . 66
4.3 Sandia Flame D . . . 68
  4.3.1 Reactors Independence Study . . . 69
  4.3.2 Major Species Concentration . . . 74
  4.3.3 Minor Species Concentration . . . 85
  4.3.4 Solver Performance . . . 91

5 Conclusion and Recommendations **97**
5.1 Conclusion . . . 97
5.2 AGNES Improvement Recommendation . . . 99

Bibliography **101**

ABSTRACT

In the present era of environment-friendly and clean combustion systems, there is an increasing demand for fast and accurate tools for emission predictions. The best choice is the CFD-CRN method which is a combination of computational fluid dynamics (CFD) and chemical reactor network (CRN) for decoupled simulation of fluid flow and detailed chemical kinetics. This thesis describes an improved solver implementation for resolving a constructed CRN using only global resolution methods and debunks the notion of needing any form of sequential resolution.

This research focuses on further improving the Python based computational tool AGNES, developed at the Delft University of Technology. AGNES can automatically cluster CFD cells into reactors, solve the network and visualise the results [**?**][**?**]. This project aims at boosting the performance by reducing computation time and selecting a PETSc [1] based global resolution approach using time integration and Newton's method. Sandia flame D, a piloted methane-air jet flame (Re=22400) [2], is chosen as the test case. The CRN results for species concentration, mainly NO and CO, are validated with the experimental data and CFD simulation results to ensure no compromise on the accuracy. The solution time and convergence rate were compared for pre-research AGNES (AGNES v1.1) and the current version (AGNES v1.2).

Results show that opting for an entirely global resolution approach is computationally feasible at higher reactor count (more than 500 reactors) and proves superior to the pre-research version. The achieved speedup is around 13% and with smart Jacobian evaluation, this is risen up to 21%. The potential reason for the improved performance is identified as the capability of the global time integration method (in place of local sequential resolution approach) to provide sufficient convergence with an increasing number of reactors and increased complexity. Moreover, the solver can be further augmented by selecting an efficient way for Jacobian calculations either by using automatic differentiation or simplifying the current Python loop approach.

LIST OF FIGURES

1.1 The Monfalcone burner used by Benedetto et al. [3] 12
1.2 Schematic representation of the KPP solution procedure for non-linear system of equations [4] 14
1.3 Comparison between KPP numerical results (lines) and experimental measurements (symbol) at several locations in (a) axial direction and (b) radial direction [4] 14
1.4 KPPSMOKE residuals norm 1 trends, normalized with respect to their initial value. Continuous lines: resolution through the sequential approach (local solver). Dotted lines: resolution through the global time integration. Dashed lines: resolution using global Newton's method [5] 15
1.5 Schematic description of AGNES CFD-CRN approach 18

2.1 Schematic representation of a PSR model in Cantera [6] 25
2.2 Three reactors system 29
2.3 Transport Jacobian (J_S) sample structure accounting for reactor interconnectivity, based on Figure 2.2 configuration and energy (temperature recalculation) OFF 33
2.4 Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Figure 2.2 configuration and energy (temperature recalculation) OFF 33
2.5 Transport Jacobian (J_S) sample structure accounting for reactor interconnectivity, based on Figure 2.2 configuration and energy (temperature recalculation) ON. Gray blocks are additional Jacobian contributions due to solving the energy equation for each reactor . . . 34
2.6 Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Figure 2.2 configuration and energy (temperature recalculation) ON. Gray blocks are additional Jacobian contributions due to solving the energy equation for each reactor 34
2.7 Numerical libraries of PETSc [7] 36
2.8 Schematic to visualise AGNES v1.1 resolution methodology 49
2.9 Schematic to visualise AGNES v1.2.1 resolution methodology 52

2.10 Schematic to visualise AGNES v1.2.2 resolution methodology 54
2.11 Schematic to visualise AGNES v1.2.3 resolution methodology 55

3.1 Close up of Sandia flame [2] 58
3.2 Dimensions of Sandia flame test setup [2] 58
3.3 Axial profiles of measured mixture fraction and temperature (Favre averaged) in piloted flames C, D, E, and F [8] 59
3.4 Visualisation of all axial locations where temperature and species concentration are measured 60

4.1 Comparison of solution time to resolve a single reactor system using three solvers 66
4.2 Number of linear solve iterations based on different choices of solving scheme and preconditioner for solving a single reactor system . 66
4.3 Three reactors system 67
4.4 Analytical Jacobian for the three reactors system 67
4.5 Comparison of solution time to resolve a three reactors system using three solvers 68
4.6 Transport Jacobian (J_S) sample structure accounting for reactor interconnectivity, based on Sandia flame D CFD simulation and energy (temperature recalculation) OFF 69
4.7 Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Sandia flame D CFD simulation and energy (temperature recalculation) OFF 69
4.8 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of CO_2 mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75) 70
4.9 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of H_2O mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75) 71
4.10 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of CO mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75) 71

4.11 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of NO mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75) 72
4.12 Variation in volume integral averaged CO_2 mass fraction at D75 axial location with increasing number of reactors 73
4.13 Variation in volume integral averaged H_2O mass fraction at D75 axial location with increasing number of reactors 73
4.14 width=.8 73
4.15 Variation in volume integral averaged NO mass fraction at D75 axial location with increasing number of reactors 73
4.16 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D15 (x = 108 mm & x/d_{jet} = 15) 74
4.17 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30) 75
4.18 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45) 75
4.19 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D60 (x = 432 mm & x/d$_{jet}$ = 60) 76
4.20 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75) 76
4.21 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D15 (x = 108 mm & x/d$_{jet}$ = 15) 77
4.22 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30) 78

4.23 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of O_2 mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$) 78

4.24 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of O_2 mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$) 79

4.25 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of O_2 mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$) 79

4.26 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of H_2O mass fraction at axial location D15 (x = 108 mm & $x/d_{jet} = 15$) 80

4.27 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of H_2O mass fraction at axial location D30 (x = 216 mm & $x/d_{jet} = 30$) 80

4.28 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of H_2O mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$) 81

4.29 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of H_2O mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$) 81

4.30 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of H_2O mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$) 82

4.31 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of CO_2 mass fraction at axial location D15 (x = 108 mm & $x/d_{jet} = 15$) 82

4.32 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (◆), and volume integral averages (right) of CO_2 mass fraction at axial location D30 (x = 216 mm & $x/d_{jet} = 30$) 83

4.33 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$) 83
4.34 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$) 84
4.35 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$) 84
4.36 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D15 (x = 108 mm & $x/d_{jet} = 15$) 85
4.37 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D30 (x = 216 mm & $x/d_{jet} = 30$) 86
4.38 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$) 86
4.39 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$) 87
4.40 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$) 87
4.41 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D15 (x = 108 mm & $x/d_{jet} = 15$) 88
4.42 Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D30 (x = 216 mm & $x/d_{jet} = 30$) 89

4.43 Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$) 89
4.44 Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$) 90
4.45 Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$) 90
4.46 Solving time for various reactor size with energy calculation switch on/off using different solver implementations 91
4.47 Solution time in hr:min:secs for solving a 1000 reactors network using different solver implementations 92
4.48 Frequency of Jacobian evaluations with increasing number of reactors 93
4.49 Improvement in solving time after using the improved Jacobian evaluation approach 93
4.50 Relative importance of three resolution methods in terms of time for different solver implementations 94
4.51 Residual function norm 2 trends for AGNES v1.2.1 and AGNES v1.2.3, normalized with respect to the initial value. Dash dot line: resolution through the sequential approach (Cantera based local solver). Solid line: global resolution using time integration. Dashed lines: resolution using globalised non-linear solver 95

LIST OF TABLES

2.1 List of PETSc linear solvers 40
2.2 List of PETSc preconditioners 41
2.3 List of PETSc nonlinear solvers 43
2.4 List of PETSc time integration methods 45

3.1 Axial locations and their respective radial range of Sandia flame D experimental results (d=0.0072 m) 60
3.2 Zone limits and reactor criteria as specified by Monaghan et al. [9] . 62

4.1 Product species concentration for methane-air combustion at $\phi =$ 1.0 inside a single PSR 64
4.2 Product species concentration for methane-air combustion at $\phi =$ 1.5 inside a single PSR 65
4.3 Product species concentration for methane-air combustion at $\phi =$ 0.5 inside a single PSR 65
4.4 Product concentration for methane-air combustion at $\phi = 1.0$ inside a three PSR network 68

NOMENCLATURE

AGNES	Automatic Generation of Networks for Emission Simulation
BDF	Backward Differentiation Formula
BFS	Breadth First Search
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CRN	Chemical Reactor Network
DNS	Direct Numerical Simulation
FGM	Flamelet Generated Manifold
KPP	Kinetic Post-Processor
LES	Large Eddy Simulation
ODE	Ordinary Differential Equation
PaSR	Partially Stirred Reactor
PFR	Plug-Flow Reactor
PSR	Perfectly Stirred Reactor
RANS	Reynolds Average Navier Stokes
RSM	Reynolds Stress equation Model
SFD	Sandia Flame D
SFIRN	Simplified Fluid dynamics by Ideal Reactor Networks
SLES	Scalable Linear Equations Solvers
SNES	Scalable Nonlinear Equations Solvers
SUNDIALS	SUite of Non-linear DIfferential ALgebraic equations Solver

1

INTRODUCTION

Combustion is the central driving force of today's transportation and power generation technologies. They rely heavily on the energy generated from the combustion of non-renewable fossil fuels. However, this energy comes at the cost of inevitable pollutant emissions, which are detrimental to both the environment and all living organisms. One of the most significant contributors to the ever-growing problem of pollution is the aviation industry. For the past two decades, several researchers are developing clean combustion technologies to curb emission production in the combustion chamber [10][11]. This research creates a demand for combustion simulation tools to generate emission predictions because such novel technologies' physical testing proves to be difficult and quite expensive. This thesis focuses on developing an improved reactor network solver to generate accurate emission predictions at a reasonable computational cost.

1.1. BACKGROUND

Combustion is a high temperature exothermic chemical process where chemical energy stored in reactants is released in the form of heat. It is usually a chemical reaction between a fuel and an oxidizer (generally air). Gas turbine engines utilize combustion to power the turbines and produce useful work. A simple chemical reaction for the combustion of a hydrocarbon ($C_x H_y$) in air is shown below.

$$\mathrm{C}_x\mathrm{H}_y + \mathrm{a}(\mathrm{O}_2 + 3.76\,\mathrm{N}_2) \longrightarrow x\,\mathrm{CO}_2 + \frac{\mathrm{y}}{2}\mathrm{H}_2\mathrm{O} + 3.76\,a\,\mathrm{N}_2 \tag{1.1}$$

Equation 1.1 shows a stoichiometric combustion of $C_x H_y$ to create products. Stoichiometric combustion occurs when just enough oxidizer is present to burn the fuel completely. This is often characterized by equivalence ratio (ϕ). It is simply a ratio of actual fuel-to air ratio to stoichiometric fuel-to-air ratio. The products can vary depending on equivalence ratio of the mixture. Excess fuel (rich mixture, $\phi > 1$) or excess air (lean mixture, $\phi < 1$) results in by-products in the form of unused reactants.

1.2. CHEMICAL KINETICS

Combustion process can be described easily with a chemical reaction. Chemical reactions occur due to collision between molecules and chemical kinetics deals with understanding their rates. There are three basic types of elementary reactions:

1. Unimolecular reaction: This type of reaction consist of one molecule and products are formed by rearrangement or dissociation of that molecule.

$$\mathrm{A} \longrightarrow \text{products} \tag{1.2}$$

2. Bimolecular reaction: This refers to reaction of two molecule, either similar or different, to form products.

$$\mathrm{A} + \mathrm{B} \longrightarrow \text{products} \tag{1.3}$$

$$\mathrm{A} + \mathrm{A} \longrightarrow \text{products} \tag{1.4}$$

3. Termolecular reaction: This is a three body reaction usually a recombination reaction, as shown below. The third molecule M's main purpose is to absorb internal energy of the newly formed product C. Otherwise, C will dissociate back into its parent reactants A and B.

$$\mathrm{A} + \mathrm{B} + \mathrm{C} \longrightarrow \text{products} \tag{1.5}$$

$$\mathrm{A} + \mathrm{B} + \mathrm{M} \longrightarrow \mathrm{D} + \mathrm{M} \tag{1.6}$$

Lets consider a bimolecular reaction of A and B to from molecule C and D as shown in Equation 1.7. ν_i is the stochiometric coefficient for each molecule involved in the reaction. Rate of reaction, which is also known as rate of depletion of reactant A or rate of production of product C, is defined as the product of species concentration raised to the power of their respective stochiometric coefficients. This is shown in Equation 1.8.

$$\nu_\mathrm{A}\mathrm{A} + \nu_\mathrm{B}\mathrm{B} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \nu_\mathrm{C}\mathrm{C} + \nu_\mathrm{D}\mathrm{D} \tag{1.7}$$

$$-\frac{1}{\nu_A}\frac{d[A]}{dt} = \frac{1}{\nu_C}\frac{d[C]}{dt} = k_1[A]^{\nu_A}[B]^{\nu_B} = -k_2[C]^{\nu_C}[D]^{\nu_D} \tag{1.8}$$

$$k = A'e^{\frac{-E_a}{RT}} \tag{1.9}$$

Equation 1.9 is known as the Arrhenius equation. Here, k is the reaction rate coefficient or rate constant, A' is the pre-exponent factor, E_a is the activation energy, T is the temperature and R is the universal gas constant. Therefore, rate of reaction is dependent on the temperature and pressure, which affect molecular space density in turn affecting species concentrations.

1.3. REACTION MECHANISMS

Equation 1.1 seems to suggest a simple one step reaction for combustion of hydrocarbons. but in reality, hydrocarbon combustion is governed by multi-step complicated set of reactions. A reaction mechanism is step-by-step collection of elementary reactions by which the entire global reaction occurs. The slowest step in the reaction mechanism is known as rate-defining step and hence determines the rate law for the global reaction. Reaction mechanism also consists of several intermediate species that are absent from the global reaction. Based on the participating species and elementary steps, a correct choice of reaction mechanisms is an critical step to ensure the accurate emission predictions.

1.3.1. GRI-MECH 3.0

GRI-Mech 3.0 [12] is an optimised reaction mechanism developed for natural gas combustion, including NO_X formation and re-burn chemistry. This mechanism consists of 53 species,including nitric species, and 325 elementary reactions. This is specifically optimised for the higher temperature range of 1000 - 2500 K and 10 torr - 10 atm pressure range. Based on its accuracy and application, this mechanism is selected for this project.

1.4. EMISSIONS

Combustion is the main source of energy in most of sectors. However, combustion process has its own disadvantage in the form of pollutants or emissions. Some of these pollutants are inevitable such as carbon-dioxide which is guaranteed product of combustion using any fossil fuel or hydrocarbon. Other emission include minor species like CO, NO_X and unburnt hydrocarbons (UHC). For the conclusion of this project, only NO and CO are studied out of other minor species and hence their formation processes are elaborated below.

1.4.1. CARBON MONOXIDE (CO)

CO formation mechanism starts with the combustion of hydrocarbon. Lets consider combustion of methane CH_4 at a high temperature (> 1000 K). Combustion of CH_4 requires free radicals like O, H, OH and HO_2 to react with the fuel molecule as shown below in Equation 1.10. These radicals are formed in the pre-heat zone or before ignition. Below reactions end with the formation CH_3 intermediate species [13].

$$\begin{gathered} \mathrm{O + CH_4 \rightleftharpoons CH_3 + OH} \\ \mathrm{OH + CH_4 \rightleftharpoons CH_3 + H_2O} \\ \mathrm{H + CH_4 \rightleftharpoons CH_3 + H_2} \\ \mathrm{HO_2 + CH_4 \rightleftharpoons CH_3 + H_2O_2} \end{gathered} \tag{1.10}$$

The intermediate CH_3 species with further oxidizes to form CH_2O.

$$\begin{gathered} \mathrm{CH_3 + O_2 \rightleftharpoons CH_2O + OH \rightleftharpoons CH_3O + O} \\ \mathrm{CH_3 + O \rightleftharpoons CH_2O + H} \end{gathered} \tag{1.11}$$

CH_2O undergoes oxidation to form CO as shown in Equation 1.12.

$$\mathrm{CH_2O \xrightarrow{oxidation} CO + H_2O} \tag{1.12}$$

CO further oxides through several reactions, as shown in Equation 1.13 to form CO_2 which is in-fact the final end product of complete combustion.

$$\begin{gathered} \mathrm{CO + O_2 \rightleftharpoons CO_2 + O} \\ \mathrm{CO + OH \rightleftharpoons CO_2 + H} \\ \mathrm{CO + O + M \rightleftharpoons CO_2 + M} \end{gathered} \tag{1.13}$$

Formation of CO consists of relatively faster reactions whereas oxidation of CO into CO_2 is a slow reaction. Therefore, at low temperature, fuel rich area and low residence time contribute towards high CO concentration.

1.4.2. NITROGEN OXIDES (NO_X)

NO_X consists of NO, NO_2 and N_2O. Formation of NO_X is highly complex as it is formed through several pathways: Thermal NO_X mechanism, Prompt mechanism, NNH mechanism, N_2O pathway and Fuel NO_X pathway.

THERMAL MECHANISM

This mechanism is also known as Zeldovich mechanism [14], as described below. This mechanism begins with a stable molecule M of high energy which is necessary to break the bounds of O_2 into O radicals. The liberated O radical then reacts with N_2. However, this reaction requires high activation energy to break nitrogen molecule's triple bond. Hence, this mechanism is only dominant at high temperature (> 2000 K), but still occurs at low temperatures.

$$\begin{aligned} O_2 + M &\rightleftharpoons 2O + M \\ O + N_2 &\rightleftharpoons NO + N \\ N + O_2 &\rightleftharpoons NO + O \\ N + OH &\rightleftharpoons NO + H \end{aligned} \tag{1.14}$$

PROMPT MECHANISM

The HCN radical is formed in fuel rich condition using the below mechanism [15]. The liberated N radical then oxidizes to form NO. This reaction has a significantly lower activation energy, hence this mechanism is the biggest contributor to the formation of NO mainly in fuel rich condition.

$$\begin{aligned} CH_2 + N_2 &\rightleftharpoons HCN + NH \\ C + N_2 &\rightleftharpoons CN + N \\ CH + N_2 &\rightleftharpoons HCN + N \\ N + O_2 &\rightleftharpoons NO + O \end{aligned} \tag{1.15}$$

NNH MECHANISM

This pathway for NO formation [15] is achieved by oxidation of NNH radicals.This mechanism is a dominant source of NO production at low temperature fuel-rich conditions, where thermal NO_X pathway is suppressed.

$$\begin{aligned} N_2 + H &\rightleftharpoons NNH \\ NNH + O &\rightleftharpoons NO + NH \end{aligned} \tag{1.16}$$

N_2O MECHANISM

This mechanism is only important in case of higher pressure because it requires completion of a three body reaction. This mechanism is shown in Equation 1.17 [15]. In presence of a third molecule M, N_2 reacts with O radical to form N_2O which later oxidises to form NO.

$$\begin{aligned} N_2 + O + M &\rightleftharpoons N_2O + M \\ N_2O + O &\rightleftharpoons NO + NO \end{aligned} \tag{1.17}$$

FUEL NO_X PATHWAY

This mechanism is only applicable if there is fuel bound nitrogen such as in case of coal burning. Therefore, this is not relevant for this project.

1.5. TRANSPORT PROCESSES

Transport equations describe how scalar quantities such as temperature and species concentration are transported in space. Modeling of the combustion zone requires accurately solving of these equations. All the transport equations are presented below.

Mass conservation

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho \vec{v})}{\partial x} = 0 \tag{1.18}$$

where ρ is the mixture density and $\vec{v}$ is the mixture velocity.

Species conservation

$$\frac{\partial \rho_i}{\partial t} + \vec{\nabla}.\rho_i \vec{v} = -\vec{\nabla}.\rho_i \vec{V_i} + M_i \Omega_i \tag{1.19}$$

Each term in the above equation describes a different transport phenomenon or source/sink term. For example, $\vec{\nabla}.\rho_i \vec{v}$ describes the convective flux (convection), $-\vec{\nabla}.\rho_i \vec{V_i}$ is the diffusive flux (diffusion) and finally, $M_i \Omega_i$ is the species production term (source). Here, ρ_i is the partial density of species i, $\vec{V}$ is the diffusion velocity vector, M_i and Ω_i are the molar mass and production rate of species i.

Momentum conservation

$$\frac{\partial \rho \vec{v}}{\partial t} + \vec{\nabla}.\rho \vec{v} \otimes \vec{v} = -\vec{\nabla}.\sigma + \rho \vec{g} \tag{1.20}$$

$$\sigma_{ij} = -p\delta_{ij} + 2\mu(\frac{1}{2}(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}) - \frac{1}{3}(\frac{\partial v_k}{\partial x_k})\delta_{ij}) \tag{1.21}$$

where σ is the stress tensor, given in equation 1.21, μ is the dynamic viscosity, p is the pressure, δ_{ij} is the Kronecker delta and ρg is the body force term. This momentum transport equation is also known as Navier-Stokes equation.

Energy conservation

$$\frac{\partial \rho h}{\partial t} + \vec{\nabla}.\rho h \vec{v} = -\vec{\nabla}.\vec{J_q} + \Phi_{visc} + \frac{Dp}{Dt} + s_e \tag{1.22}$$

$$\frac{DP}{Dt} = (\frac{\partial}{\partial t} + \vec{v}.\vec{\nabla})p \tag{1.23}$$

$$\Phi_{visc} = \tau_{ij} \frac{\partial v_i}{\partial x_i} \tag{1.24}$$

The above equation shows the transport process of specific enthalpy. Here, h is the specific enthalpy and $\vec{J}_q$ is the diffusive vector. On breakdown of the equation, $\vec{\nabla}.\rho h \vec{v}$ is convection term, $-\vec{\nabla}.\vec{J}_q$ is diffusion, Φ_{visc} describes viscous dissipation (Equation 1.24), $\frac{Dp}{Dt}$ presents enthalpy transport due to pressure change (material derivative as shown in Equation 1.23) and finally, s_e is the source/sink term.

1.6. COMBUSTION MODELING

Most of the combustion systems involve the burning of hydrocarbons, which has major byproducts in the form of carbon dioxide (CO_2) and water vapours (H_2O). Along with nitrogen (N_2) and unused oxygen (O_2), this constitutes the major species present in the exhaust. However, nitrogen oxides (NO_X) and carbon monoxide (CO), that forms the bulk of minor species, are far more harmful to environment and human health. NO_X, which is mixture of NO, NO_2 and N_2O, causes depletion of ozone in the Stratosphere thus leaving the Earth's surface vulnerable to Sun's ultraviolet radiation. In the Troposphere, NO_X is the main constituent of photo-chemical smog, acid rain and generates ozone by chemical reactions, which in turn increases the temperature near the Earth's surface. CO, on the other hand, is detrimental for living organisms. If inhaled, it produces a complex compound with the hemoglobin in blood and inhibits blood's capability of oxygen transport for metabolism. This could lead to unconsciousness, nausea and even death. Hence, the legislation has far stricter rules to curb the problem of these emissions. This requires efficient and accurate emission prediction tools.

Reliable prediction of both major and minor species comprises accurate modeling of the flow field and chemical kinetics involved in the combustion process. This approach results in a coupled and large system of nonlinear equations, thus significantly increasing the computational time and resources. Computational modeling of combustion systems can either prioritize the solution of the flow field with simplified chemistry or focus primarily on the chemical kinetics. This way reduces the computational load but results in unreliable results.

Additionally, flow is highly turbulent in most combustion systems due to the high Reynolds number, resulting in fluctuations in state properties and flow velocity. The presence of eddies characterizes turbulence in a flow. Energy is transferred from larger eddies to smaller and smaller turbulence eddies. This phenomenon is known as a turbulence energy cascade. The smallest eddies in the flow are of the Kolmogorov scale. It is important to capture most of the eddies to accurately predict the flow properties.

Both flow prediction and species concentration calculation are extremely complex problem on their own. Both can be solved together for a system but it will incur high computational costs and time. Hence, prioritising one of the two problems is the way forward. Modeling of combustion systems is therefore divided into three methods, as described below.

1. Computational Fluid Dynamics (CFD)
2. Chemical Reactor Network (CRN)
3. Hybrid CFD-CRN method

1.6.1. COMPUTATIONAL FLOW DYNAMICS (CFD)

It is a high-fidelity solution of turbulent flows. Different CFD methods differ in their approach for modeling the smaller turbulence eddies. The most accurate method is Direct Numerical Simulation (DNS), where the turbulence is explicitly resolved. The gird resolution is high enough to capture the smallest eddies, i.e., the grid size is the order of the Kolmogorov length scale. However, this method is quite computationally demanding and rarely used in the industry apart from research fields.

A coarser mesh is required to reduce the computational cost, but it demands a mathematical approximation for turbulence. These approaches are Reynolds Averaged Navier-Stokes (RANS) equations and Large Eddy Simulation (LES). RANS method solves the transport equations by replacing the flow and state quantities by their temporal mean values, as shown in equation 1.25 to 1.27. RANS equation are solved on a coarser mesh compared to DNS and LES, and the results become independent of grid resolution after a threshold grid size. This approach is the most preferred method for combustion modeling. Some of the common RANS models are Spalart-Allmaras (S-A), k-epsilon ($k-\epsilon$), k-omega ($k-\omega$), Shear Stress Transport (SST) and Reynolds stress equation model (RSM). They differ in their approximation for closing the Reynolds stress ($R_{ij} = \widetilde{u_i"u_j"}$) term in equation 1.26.

$$\frac{\partial\overline{\rho}}{\partial t} + \frac{\partial(\overline{\rho}\widetilde{\vec{v}})}{\partial x} = 0 \tag{1.25}$$

$$\frac{\partial\overline{\rho}\widetilde{u_j}}{\partial t} + \frac{\partial\overline{\rho}\widetilde{u_i}\widetilde{u_j}}{\partial x_i} = [-\frac{\partial\overline{p}}{x_j} + \frac{\overline{\tau}_{ij}}{\partial x_i} + \overline{\rho}g] - \frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{u_i"u_j"}) \tag{1.26}$$

$$\frac{\partial\overline{\rho}\widetilde{\phi_k}}{\partial t}+\frac{\partial\overline{\rho}\widetilde{\phi_k}\widetilde{u_i}}{\partial x_i}=[-\frac{\partial\overline{J}_i^k}{x_j}+\overline{\rho}\widetilde{S_k}]-\frac{\partial}{\partial x_i}(\overline{\rho}\widetilde{\phi_k"u_j"}) \tag{1.27}$$

Equation 1.25 to 1.27 are the RANS governing equations. f" and $\widetilde{f}$ are fluctuations and Favre-averaged quantities and are defined as-

$$\widetilde{f}=\frac{\langle\rho f\rangle}{\langle\rho\rangle} \tag{1.28}$$

$$\langle f\rangle=\frac{1}{n_{samples}}\sum_{n=1}^{n_{samples}}f_n \tag{1.29}$$

$$f"=f-\widetilde{f} \tag{1.30}$$

In LES, large scales of turbulence are solved, whereas smaller scales are modeled. This flow splitting is done by low-pass filtering of transport equations, removing any smaller-scale information from the numerical solution. The grid resolution determines the portion of flow being modeled. It is highly accurate compared to RANS with a finer mesh, but computational cost increases with finer mesh choice.

Furthermore, most of RANS methods employ a reduced chemistry model, which involve fewer species. This simplification assumes that minor species such as CO and NO_X do not affect the flow and its properties. This makes CFD methods ideal for modeling the flow. Minor species are often calculated using CFD post-processing models which are not accurate compared to experimental results.

1.6.2. CHEMICAL REACTOR NETWORK (CRN)

This approach utilizes a simplified flow field with a detailed kinetic mechanism for emission predictions. This way, the species concentrations, especially minor species, are calculated efficiently with low CPU consumption. However, the simplified flow field omits the turbulent fluctuations in the flow, which affects the accuracy of CRN results.

The combustion region is replaced with a network of ideal reactors, either perfectly stirred Reactor (PSR), plug flow Reactor(PFR) or partially stirred reactor (PaSR). Each ideal reactor is assumed to be in steady-state, and all reactants are perfectly mixed in the control volume. In case of PSR, there are no spatial variations in species concentrations and state properties, making the system 0D. This CRN approach simplifies the problem by eliminating the highly nonlinear momentum conservation equation. Therefore, it is required to solve only mass, species, and energy conservation equations. The conservation equations in steady state are

Mass conservation

$$\dot{m}_{in} = \dot{m}_{out} \tag{1.31}$$

where $\dot{m}_{in}$ and $\dot{m}_{out}$ are the rate of mass entering and leaving the system respectively.

Species conservation

$$\dot{\omega}_i MW_i V + \dot{m}(Y_{i,in} - Y_{i,out}) = 0; \quad i = 1,2...,N \tag{1.32}$$

$$\dot{\omega}_i = f(|X_i|_{CV}, T) = f(|X_i|_{out}, T)$$

$$Y_i = \frac{X_i MW_i}{\sum_{j=1}^{N} X_j MW_j}$$

where $\dot{\omega}_i$ is the rate of production of species i, MW_i is the molecular weight of species i, X_i and Y_i are the mole fraction and mass fraction of species i and V is the reactor volume.

Energy conservation

$$\dot{Q} = \dot{m}(\sum_{i=1}^{N} Y_{i,out} h_i(T) - \sum_{i=1}^{N} Y_{i,in} h_i(T_{in})) \tag{1.33}$$

$$h_i(T) = h^0_{f,i} + \int_{T_{ref}}^{T} c_{p,i} dT$$

where $\dot{Q}$ is the heat transfer rate due to heat loss/addition from walls separating the external environment, h_i is specific enthalpy of species i, $h^0_{f,i}$ is specific heat of formation of species i and $c_{p,i}$ is specific heat capacity of species i.

There are N+1 equations for N+1 variables (N species and temperature). It is a system of nonlinear algebraic equations which can be stiff based on the choice of reaction mechanism. CRN method provides accurate estimations for minor species concentration in the flow.

1.6.3. HYBRID CFD-CRN

Due to the CFD and CRN method's disadvantages, a hybrid approach is preferred, where the solution for the flow field and chemistry are found separately. The first step is to model the flow accurately using a simplified chemistry model.

This can be done using either CFD or by correlation-based modeling from experimental data. This approach relies on a massive assumption that minor species have a negligible impact on the modeled flow field. As a next step, a chemical reactor network is constructed by clustering physically and chemically similar cells from CFD simulation data. Some of the common properties used for clustering are local stoichiometry, temperature, velocity vectors, etc.. Each cluster is modeled as an ideal reactor, either a PSR, PFR, or PaSR. The last step involves the solution of CRN using a detailed kinetic mechanism to generate emission predictions [16].

Its high accuracy and less CPU consumption make the hybrid CFD-CRN modeling method the most promising emission prediction method. Therefore, researchers at the Delft University of Technology developed a computational tool based on the hybrid approach. This tool is called Automatic Generation of Networks for Emission Simulations (AGNES) [6] [17]. This tool is discussed in detail in Chapter 1.8.

1.7. PAST DEVELOPMENTS IN CFD-CRN

For acceptable CPU consumption, CFD-CRN combustion modeling method is widely used where computation of flow field is done using Computational Fluid Dynamics (CFD) with reduced chemistry. A chemical reactor network is then constructed using CFD simulation data based on a pre-defined clustering criteria. The detailed chemistry is then solved using Chemical Reactor Network (CRN). This is done to improve the accuracy in prediction of minor species which is integral for emission prediction.

The first account of CRN application for emission prediction was in 1998 by Ehrhardt et al. [18]. CFD-CRN method was used to predict NO_X emissions where CRN was manually constructed from CFD data. This was demonstrated using a 2D simple geometry which was re-constructed using Plug Flow Reactors (PFRs) based on k-ϵ turbulence model based CFD data. However, this method was only suitable for simpler geometry.

In 2001, Benedetto et al. [3] analyzed a 320 MW furnace, shown in Figure 1.1, using reacting flow CFD based automatic generated CRN. CRN was created by clustering cells based on temperature and stochiometry and later each cluster was represented as an ideal reactors. This work involved a major assumption that minor species have negligible impact on the flow field and CFD simulation with reduced chemistry was used to generate flow field based on the effect of reactions. The entire process of constructing the CRN was automatic except the user of input to set the minimum number of reactors to represent each zone.

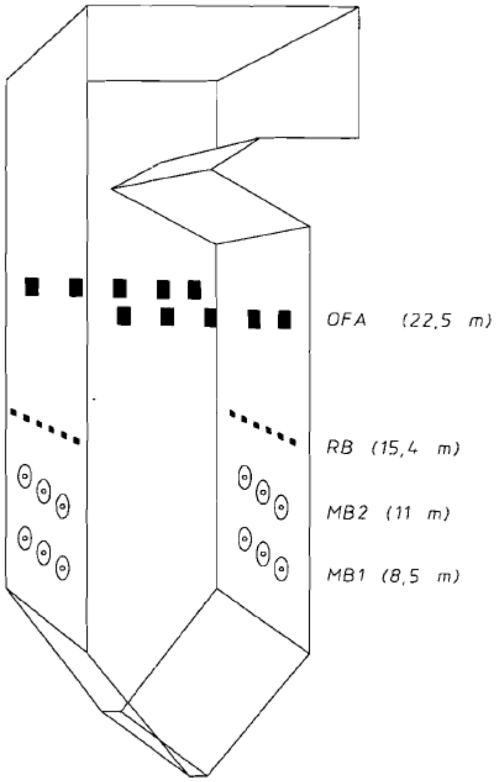

Figure 1.1: The Monfalcone burner used by Benedetto et al. [3]

Later in 2001, Faravelli et al. [10] developed an automatic CRN construction and solver algorithm called Simplified Fluid dynamics by Ideal Reactor Networks (SFIRN) and used it to determine NO_X emission from industrial boilers. This method allowed the application of more complex kinetic schemes, re-circulation zones modelling and the use of liquid fuels.

This method was further improved by Falcitelli et al. [19] in 2002. Falcitelli et al. shared the algorithm for automatic generation of CRNs based on CFD simulation data. This algorithm was described as:

- Step 1: Calculate flow field properties such temperature and major species concentration as well as local stochiometry (CFD post processing) using CFD on a fine mesh

- Step 2: Cluster similar and neighboring CFD mesh cells into homogeneous zones based on temperature and local stochiometry, and represent each zone as an ideal reactor

- Step 3: Compute each reactor operating volume and temperature by adding volumes of constituent cells and solving energy conservation expression repectively

- Step 4: Model each reactor as a PSR or PFR, based on the velocity vector distribution in the respective zones

- Step 5: Determine the mass exchange between clusters by adding massflows in cell faces of boundary cells

- Step 6: Solve the CRN with a detailed kinetic model for species present in the combustion

This algorithm was further improved by clustering the cells based on unmixedness in order to reduce the number of clusters created by temperature and stoichiometry clustering criteria [16].

In 2005, Frassoldati et al. [20] used a similar clustering approach as Falcitelli et al. to predict NO_X emissions. The research was done to estimate the minimum number of reactors required for generate correct NO_X concentration prediction. The clustering criteria used temperature and species composition to generate clusters which were later assigned as Perfectly Stirred Reactor (PSR).

Based on prior research done in [10] [19] and by Cuoci et al. (2007, 2013) [11] [4], a clustering algorithm was developed called KPP (Kinetic Post Processor). Cuoci et al. (2007) [11] modeled turbulent diffusion syngas flames using KPP algorithm. The concept of kinetic turbulent temperature was introduced by Cuoci et al. for considering turbulent fluctuations present in the actual flame and estimation of this kinetic turbulent temperature was based on temperature variance from CFD data. Later in 2013, Cuoci et al. [4] finally developed the KPP algorithm that allowed automatic construction of large reactor networks. It showed that the inclusion of turbulent fluctuation in CRN modelling had a massive impact on accuracy. Cuoci et al. made several recommendation for investigating the impact of clustering criteria for optimum balance between number of reactors and accuracy, and improvement in efficiency by working on parallelization of network solving process.

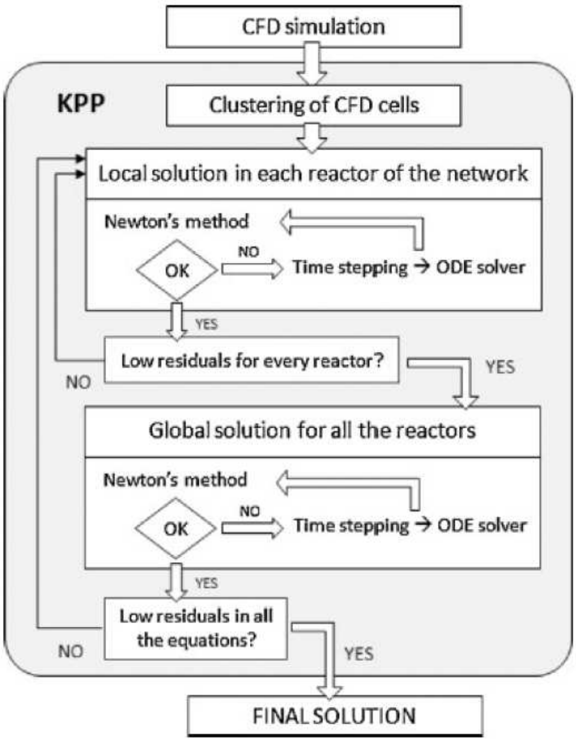

Figure 1.2: Schematic representation of the KPP solution procedure for non-linear system of equations [4]

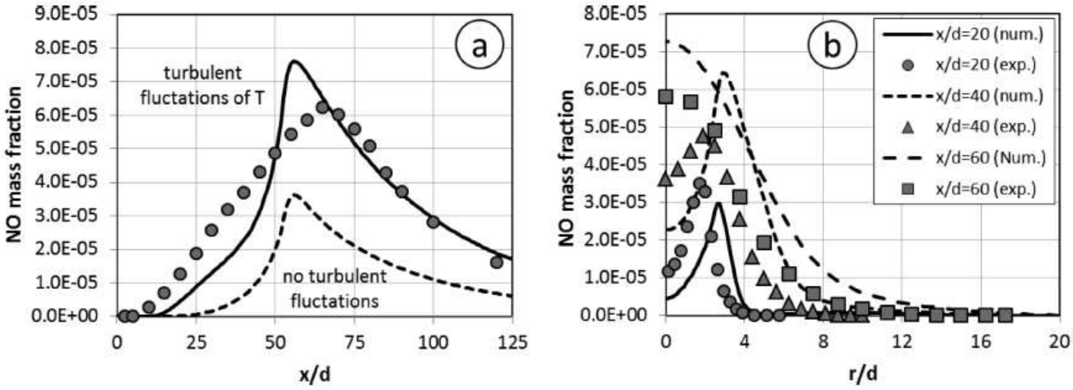

Figure 1.3: Comparison between KPP numerical results (lines) and experimental measurements (symbol) at several locations in (a) axial direction and (b) radial direction [4]

In 2013, Stagni et al. [5] addressed all the recommendations provided by Cuoci et al. and developed a fully-coupled and parallelized algorithm called KPPSMOKE. CFD simulation data was used to reduce the CFD mesh to a network of PSRs. This resulted in a highly stiff, nonlinear system of $N_C X N_S$ species mass balance equations where N_C and N_S are number of reactors and species respectively. The entire solving process was split into local and global solver, based on the findings of Cuoci et al. (2007) [11]. The use of local solver reduced the highly stiff $N_C X N_S$ nonlinear system into N_C system of N_S nonlinear equations each. Only after the local solvers reached sufficiently close to the solution, global time marching was initiated using the obtained mass fractions as initial guess. Domain distribution among processors involved distributing equal number of reactors to each processor. This method allowed efficient solving of large networks (10^5-10^6 reactors) in reasonable computation time. Furthermore, few recommendations were made regarding the use of nested parallelization to improve the parallel architecture.

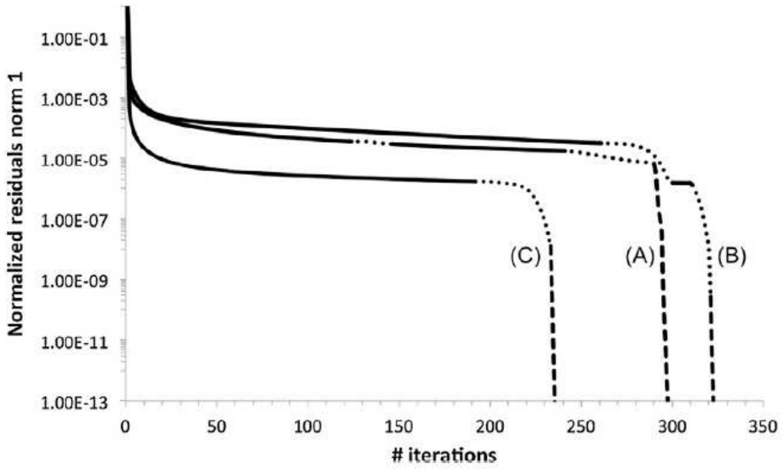

Figure 1.4: KPPSMOKE residuals norm 1 trends, normalized with respect to their initial value. Continuous lines: resolution through the sequential approach (local solver). Dotted lines: resolution through the global time integration. Dashed lines: resolution using global Newton's method [5]

Choice of clustering approach is vital for accuracy and flexibility of CFD-CRN method. In 2014, Nilsson [21] compared two different approaches for CRN construction: Principal Component Analysis (PCA) based and cluster growth based. PCA is a machine learning technique to reduce the large data sets into small sets without compromising any important information. These small data sets are generated by selecting the principal components. Principal components are new variables that are composed of linear combination of initial variables and are estimated by finding the eigenvalues and eigenvectors of its co-variance matrix. He analyzed Sandia Flame D by dividing the flame CFD mesh into multiple zones based on selected criteria. It was concluded that cluster growth based approach resulted in less number of reactors to comply with the selected criteria compared to PCA based.

For these reasons, Automatic Generation of Networks for Emission Simulation (AGNES) was developed in 2018 by Sampat, MSc [6] at the Delft University of

Technology. The detailed description and working of AGNES is presented in Section 1.8. Sampat (2018) [6] developed the hybrid CFD-CRN based simulation tool called AGNES for emission predictions from combustion systems by employing automatic CRN construction and solution, based on KPP algorithm developed by Cuoci et al.(2013) [4]. Sampat [6] [22] analyzed the effect of clustering criteria on the accuracy of emission predictions. It was observed that CO had a lower sensitivity to clustering tolerance whereas NO_X was highly sensitive to tolerance of clustering. It was concluded that AGNES results for both CO and NO_x were better in comparison with CFD results. This was due to the use detailed kinetic mechanism for emission predictions. Perpignan et al. (2019) [22] used AGNES for emission modeling of Flameless combustion. It was observed that choice of clustering criteria had a huge impact on the results and the use of AGNES highly improved NO_X prediction compared to CFD post processing. Later in 2019, Wit [17] made further improvements to AGNES by investigating the effect of temperature recalculation in CRN using energy equation on its results and performance.It was observed that the use of energy equation not only increased the computation time but also had a larger impact on mass-fraction predictions of minor species than major species. However, It was concluded that inclusion of energy equation was not recommended due to over-prediction in temperature, which was due to lack of any heat transfer modeling in the CFD-CRN solver. Based on recommendations provided by Sampat (2018) and Wit (2019), further research is being done to improve AGNES performance by utilizing the latest developments within CFD-CRN and promoting parallel architecture for CRN solution.

1.8. AGNES

Automatic Generation of Networks for Emission Simulation (AGNES) tool is a Python based emission prediction tool that employs the hybrid CFD-CRN approach of combustion modelling [6]. It reads CFD simulation results, automatically constructs a CRN by grouping similar CFD mesh cells and then solves it to generate emission predictions. Based on research [6][17], the main framework of AGNES is well developed but it is not fully optimised yet. Therefore, the Delft University of Technology wishes to improve AGNES further using state-of-the-art tools and latest developments, along with furthering research in the field of CFD-CRN modelling.

AGNES is written in Python due to its relatively high flexibility and robustness, along with its expansive library. AGNES utilizes chemistry solver **Cantera** [23] and detailed kinetic reaction mechanisms such as **GRI-Mech 3.0**[12]. Cantera [23] is an open-source chemistry library for solving problems related to chemical kinetics and transport processes. It is used as both a CRN solver as well as a chemistry bookkeeping tool. GRI-Mech 3.0 is a widely used reaction mechanism for natural gas combustion. This is one of the best CH_4/Air combustion mechanism and include NOx formation pathways in it. Furthermore, this reaction mechanism is selected for this Master's thesis due to its lower solution time, accuracy and reduced complexity.

AGNES CRN algorithm can be split into four major steps: post-processing of CFD output into graphs, construction of CRN, solution of CRN and visualization of results. Figure 1.5 shows a schematic representation of AGNES's CRN construction and solution approach. The solution process starts with the user's input for clustering criteria, preferred number of reactors and tolerance. This is followed by converting the CFD simulation output data into a graph. A graph is a data structure (created using Python dictionary data type in AGNES) comprising of nodes with inter-connections (edges) among them. Cell centers are represented as nodes and common face between two cells is referred as edge. The nodes contains all the flow and geometrical properties of computational cells and their inter-linkage information with neighbouring cells is stored in the graph edges. This is an important step for the clustering procedure. This graph defining step is only dependent on the CFD results, thus they can be re-used for different CRN setups.

This is followed by CRN construction step using graph traversal that identifies and groups similar CFD cells into reactors based on user-defined clustering criteria. Graph traversal means exploring its structure by visiting each nodes based on some systematic rule. Accuracy of a CRN solver greatly depends on the ability to construct good clusters of CFD cells which are not only physically and chemically similar, but also geometrically interconnected. This makes the clustering process one of the most integral part of a CRN solver. This can either be done manually clustering all similar neighboring cells or automatically using a clustering algorithm. A good clustering algorithm needs to ensure only similar cells and geometrically connected cells are grouped together without any discrepancy in conservation of transport properties. However, the clustering problem is an optimization problem for the number of reactors because there is a trade-off between good CPU consumption and correct CFD data representation for ensuring good accuracy. AGNES uses an iterative, automatic clustering process using the previously defined graphs.

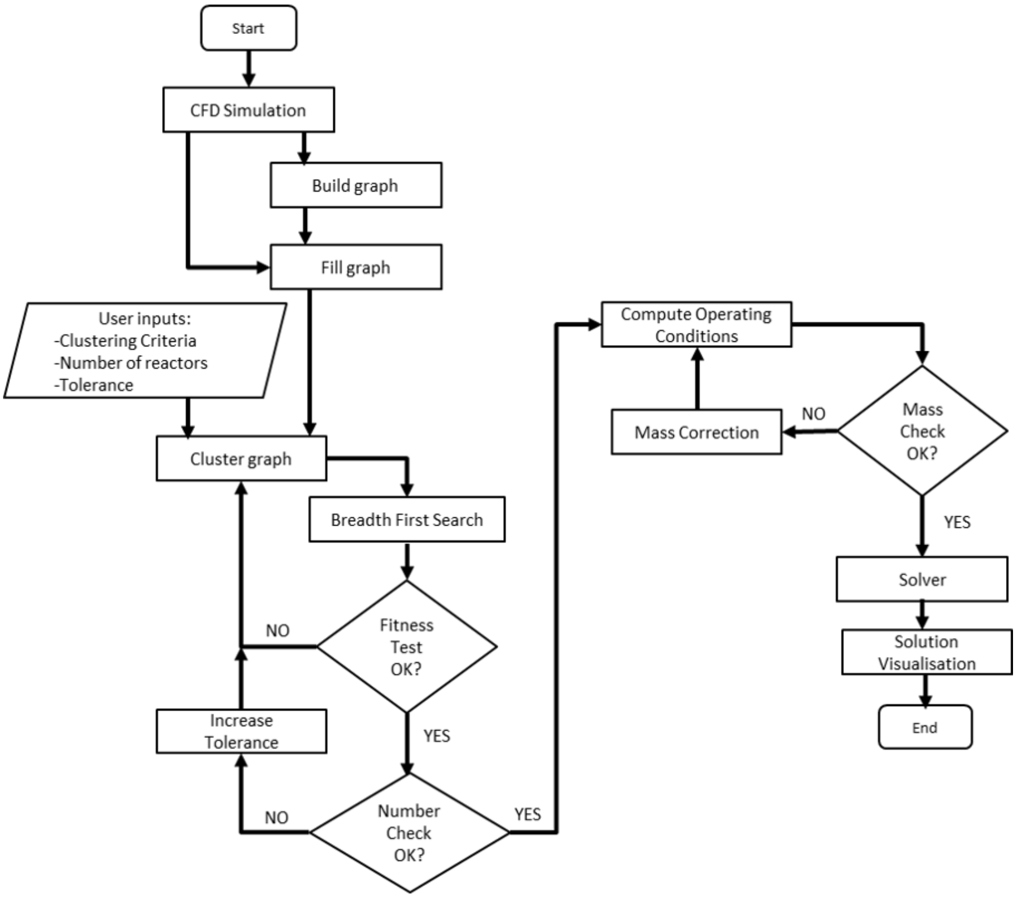

Figure 1.5: Schematic description of AGNES CFD-CRN approach

AGNES clustering method involves traversing through CFD cells graph, checking the fitness of selected cell, adding the cell into the reactor, updating cluster quantities based on constituent cells and finally updating graph of reactors forming the CRN. Detailed description of each step is provided below [6].

1. **Graph Traversal:** The first step of clustering process involves searching the graphs for similar inter-connected cells. The success of graph traversal depends on how frequently each move encounters a geometrically connected cell which is also similar in physical and chemical nature to the cluster. In AGNES, this is done using an algorithm called **Breadth First Search (BFS)** where the next cell selection is done by searching in the cluster close proximity before delving deeper in the graph.

2. **Fitness Test:** This is a safety check step where a selected cell undergoes a fitness check. The cell is added to the current reactor only if it passes the fitness test or else a new reactor is created with this cell. This fitness test involves:

    - An additional common face check over the BFS traversal to make sure only geometrically connected cells are added into reactors

- A tolerance based check where tolerance value is estimated based on the maximum and minimum value of a quantity in the CFD domain. If the absolute difference between quantity in reactor and in the selected cell is less than the tolerance, the cell passes the test. This tolerance check is relaxed in subsequent iterations to add more cells to the reactor.

$$\epsilon = \delta(c_{max} - c_{min}) \tag{1.34}$$

$$|c_{cluster} - c_{cell}| \leq \epsilon \tag{1.35}$$

- Other user-defined testing criteria such as angle between the velocity vectors are also utilized to generate a correct cluster.

3. **Quantity and Graph Update:** After adding a new cell to the reactor, the value of quantities in reactor is updated to reflect the new cell addition and a new graph is created. This process is repeated on the new graph till all cells are assigned in correct reactors.

After constructing the CRN, reactor properties like volume and mass are defined by adding its constituent CFD cell's volume and species mass. Next, each reactor's thermodynamic properties are calculated by mass-averaging temperature, pressure and other CFD cell-related properties. Generally, the total inflow and outflow mass flow rates associated with a CFD cell are not equal but rather differ by a slight difference, equal to the tolerance set during the CFD simulation process. Furthermore, the agglomeration of cells amplifies this inherent mass imbalance. Therefore, a mass correction is required before starting the solution process. This correction involves determining the relative mass flow rates leaving the reactor into the adjacent clusters and then solving for those with the total inflow supplied [6].

After setting up the operating conditions of created clusters, a Cantera reactor network object is defined with each reactor labelled as a PSR. The solution of created reactor network corresponds to solving of a highly stiff nonlinear system of equations. The stiffness of the system is due to a wide range of time scales for all reactions involved. For example, reactions responsible for heat release are mostly faster compared to reactions for emission formation such as NO_X. Thus, small time-steps are needed to integrate fast reactions but it would make it difficult to integrate slower reactions. AGNES employs two levels of solver: a local and a global solver [6][17]. This enables the solver to get to the solution faster while still being stable. The local solver is used to nudge the system closer to the solution so that the global solver does not diverge. The local solver uses Cantera as both book keeping tool as well as a ODE solver based on SUNDIALS for sequential resolution of the reactor network. The local solver advances every reactor in time,using Backward Differentiation Formulation (BDF) for each time step until local convergence is reached. The global solver only employs Cantera for book keeping and solves the custom written governing equations using Python's SciPy library. Furthermore, it uses a combination global time integration and global Newton's method for faster convergence. A detailed description of the pre-research version of AGNES is provided in Section 2.5.

1.8.1. RESEARCH POSSIBILITIES IN AGNES

Based on previous version of AGNES [6][17] along with latest developments in CFD-CRN, following research areas were recommended to improve the performance and accuracy of AGNES.

- **Improved solver implementation:** In the pre-research version of AGNES, CRN solver uses a combination of local solver using Cantera, global time integration and global Newton's method [17]. There are two glaring issues with the current solver implementation. First, the local solver scales poorly with increasing number of reactors. As, it solves every reactor sequentially till convergence, this become computational expensive with a high reactor count. Second, the global time integration is only active for less than 10% of the total solving time, hence its converging capabilities are not utilized to its fullest extent. Hence,it is recommended to design a better optimised solver routine that utilizes global resolution via time integration to arrive at a suitable initial guess of the fast-converging global Newton's solve. Furthermore, the use of time-integration for global solution is required as it would reduce the risk of instability and require less computational time compared to a sequential approach.

- **Turbulence**: In the current version of AGNES, temperature fluctuations due to turbulence are not considered. Nevertheless, Cuoci et al. (2013) [4] concluded that use of temperature fluctuations in combustion modelling led to significant improvements in emission predictions. Thus, it is suggested to verify the impact of modelled turbulent fluctuation on the accuracy of emission predictions using different test cases.

- **Liquid fuels**: Currently, AGNES is only tested with gaseous fuel [17]. However, the use of liquid fuels such as kerosene and gasoline is possible with need minor adjustments like mass-flow corrections accounting for droplet evaporation and inclusion of liquid phase chemistry.

- **Zoned/Scaled clustering**: Current version of AGNES uses only one clustering tolerance value for the entire domain. This often results in either too high resolution at less interesting locations, poor resolution at important locations or a combination of both. Although, Wit [17] solved this issue by exploring the concept of zoned clustering as proposed by Monaghan et al. [9]. However, further exploration in zoned/ scaled clustering is required for better CRN construction and accuracy.

- **Parallel architecture**: The current version of AGNES solves the CRN in sequential manner which leads to high computation time. Cuoci et al. [4] recommended a parallel architecture for KPP algorithm which led to the development of KPPSMOKE application by Stagni et al. [5] which utilizes faster parallel computation. Application of such parallel architecture for solving CRNs in AGNES could lead to improved performance and low computation time.

- **Complex geometries**: Till date, AGNES is not tried with complex geometries, which could prove to be challenging due to AGNES being designed to handle only a single Fluent fluid zone [6]. It is suggested to enable multiple zones to allow modelling of complex combustion systems.

- **Plug flow reactors (PFRs)**: Based on several observations made by Wit [17], it is recommended to utilize PFR instead of PSR to reduce the axial elongation of clusters due to lack of modelled molecular diffusion.

1.9. RESEARCH OUTLINE

In the previous Section 1.8.1, out of the listed research areas, it was concluded that designing an improved solver implementation using a combination of resolution method is an interesting research gap in the field of CFD-CRN. Most of the CFD-CRN solvers such as KPPSMOKE [5] use some kind of local resolution aiming to resolve every reactor sequentially before even trying a global resolution. Motivation for this approach was to ensure the global Newton's method that offers fast convergence but is quite unstable and prone of divergence if the initial guess is not close to the final solution. However, using globalised Newton's method either using line search or trust region could solve this issue of convergence. These globalised methods allows for any arbitrary starting point. This is further discussed in Section 2.4.5 and 2.5. Furthermore, no explanation was given for why a global time integration cannot be used to achieve similar results as the sequential approach but even faster due to time marching of the entire network. From the works in the literature study, it was concluded that there is a glaring absence of any credible reason behind the choice of sequential resolution instead of opting a global time-stepping approach. Also, possibility of new solver implementations were not well explored. TU Delft's in-house CFD-CRN solver called AGNES, can be improved by updating the solver using state-of-the-art PETSc solver library that offers wide collection of time-integration, nonlinear and linear solvers. This led to the formation of the research questions for this project.

1. **How are the performance parameters, such as computational time and convergence rate, affected by implementing a PETSc based solver for hybrid CFD-CRN method?**
2. **Does the choice of the time-stepping or nonlinear or linear scheme have any impact on the convergence rate of the CFD-CRN solver?**

It was hypothesized that reducing time spent on the local solver could lead to decrease in solution time. But a stable alternative in the form of global time integration is required so that final approach towards solution using Newton's method does not diverge. Three distinct solver implementations are developed and tested both in terms of accuracy and performance. This improvement is validated using Sandia Flame D CFD test case. CFD simulations were performed in ANSYS Fluent by André A.V. Perpignan, PhD and were later used by Wit, MSc [17] for AGNES CRN construction and validation.

1.10. OVERVIEW

An overview of the forthcoming chapters is provided in this section. Chapter 2 describes the governing equations of CRN system along with detailed description of pre-research version of AGNES solver and newly developed PETSc based solver implementations. Information on the test case combustion chamber, which is used for the construction of CRN, is provided in Chapter 3 followed by discussion of the results, including accuracy and improved solver performance in Chapter 4. Finally, the research conclusions are summarized in Chapter 5 and recommendations for future improvement of AGNES are also listed in the same chapter.

2

CHEMICAL REACTOR NETWORK SOLVER

For successful use of hybrid CFD-CRN method, the combustion region is replaced with a network of ideal reactors, either perfectly stirred Reactor (PSR), plug flow Reactor (PFR) or partially stirred reactor (PaSR). Each ideal reactor is assumed to be in steady-state, and all reactants are perfectly mixed in the control volume. This chapter provides details on the procedure to build to reactor network and later information on the solver implementation to resolve this network to generate emission predictions. Section 2.1 presents information on the Cantera [23] chemistry library and its applications on building a PSR network. Section 2.2 and 2.3 describe the governing equation and its corresponding Jacobian, that are required to resolve the system. A detailed description on the selected solver library PETSc [1] is provided in Section 2.4. Finally, the improved solver implementation is well described along with pre-research version in Section 2.5.

2.1. BUILDING REACTOR NETWORK: CANTERA

Cantera [23] is an open-source collection of tools for solving systems involving thermodynamics, chemical kinetics and transport processes. AGNES uses Cantera objects to build the reactor network and as chemistry book keeping tool. Cantera as a book keeping tool helps in storing reaction mechanisms with all its thermodynamic properties as well as maintaining a consistent state of reactors based on density, temperature and species mass fractions. Furthermore, pre-research AGNES v1.1 also uses Cantera as a time integration solver for local sequential resolution [6]. It calls SUNDIALS [24] to perform implicit BDF based

ODE time integration.

Cantera is used to maintain the consistency of reactor states i.e the sum of all mass fractions should be equal to 1 inside each reactor and the value of every species mass fraction (ω_{ki}) should be between 0 and 1. Cantera automatically adjust the reactor states in case of violation of these conditions.

$$\sum \omega_{ki} = 1 \tag{2.1}$$

$$0 \leq \omega_{ki} \leq 1 \tag{2.2}$$

Cantera offers a collection of models which can be assembled to generate a reactor network such as Mass-flow Controllers, Valves, Ideal Gas Reactor, Reservoir, etc. To setup a correct reactor network, following components are needed to be defined:

1. Reservoir: It is defined as a large volume kept at a pre-defined and constant state. It is used often as a source or sink.

2. Wall: The wall object is defined to separate either two reactors or a reactor and a reservoir. The wall is only required if energy equation is solved for each reactor. It supports heat transfer phenomena like conduction and radiation.

3. Mass Flow Controller: This Cantera object acts a flow device that governs mass flow between reactors or a reactor and a reservoir. However, this mass transfer is independent of the pressure difference. As this is independent of both upstream and downstream conditions, this object is used to design mass flow from any reactor to any object despite the pressure difference. The mass flow can be set either as constant or time dependent.

$$\dot{m} = \dot{m}_0 g(t) \tag{2.3}$$

4. Valve: It is a flow device that facilitates mass flow between reactors as a function of pressure drop across it. It is a one-way passage a negative pressure difference would mean zero mass flow rate.

$$\dot{m} = \begin{cases} K_v(P_1 - P_2) & \text{if } P_1 > P_2 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

5. Ideal Gas Reactor: It is closed reactor object set as a constant volume system by default but can be altered to a changing volume system. The reactor system is defined using independent variables such as temperature, density and species mass fractions. The state variables are mass, volume, temperature and specie composition in mass fractions.

In AGNES, all reactors are modelled as PSRs. The reactor network is created with a Ideal Gas Reactor object as core with reservoir connected on both inlet and outlet. Each reservoir is connected through two mass-flow controllers and one valve, one on each directions to act as boundary conditions, as shown in Figure 2.1.

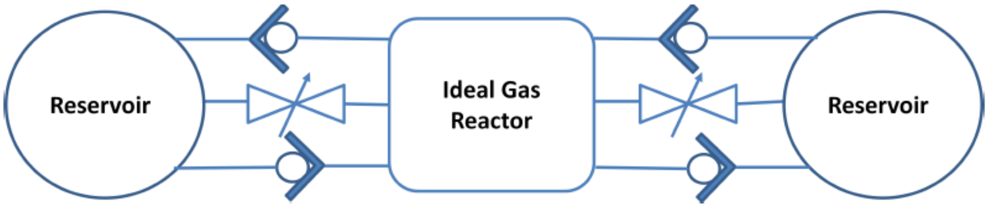

Figure 2.1: Schematic representation of a PSR model in Cantera [6]

2.1.1. BOUNDARY CONDITIONS

Based on ANSYS CFD boundary conditions, similar BCs are set for the CRN using Cantera objects like Valve and Mass Flow Controller. These BCs are Mass flow inlet, Outlet, Periodic BC and Wall.

1. Mass Flow Inlet: This boundary condition has to be manual input from the user based on ANSYS Fluent zone IDs. Hence, zone IDs should be clearly stated and readily available in the Fluent case file. For every inlet face, corresponding mass flow rate are extracted from data dictionary. Cantera Reservoir object is defined with its state matching the CFD conditions. For example, if the face is a fuel inlet, the Reservoir is defined with temperature, pressure and species composition of the fuel mixture i.e 300K, 1 atm, $Y_{CH_4} = 1$.

2. Outlet: Similar to inlets, mass flow rate is extracted form the data dictionary for face defined as outlet. The mass flow rate is used to define the mass flow controller object. It also involved a Valve, that release excess mass into a reservoir, defined at ambient conditions.

3. Periodic BC: As most of the combustion chambers are axi-symmetric such as Sandia flame D, it is crucial to capture this boundary condition well. A periodic BC consist of two faces: a periodic face and a shadow face. These faces are identified and selected as pairs and mass flow controllers are added based on the flow direction.

4. Wall: This boundary condition is only important in case of temperature recalculation. This is defined as Cantera's Wall object to model wall heat flux. Every face that is defined as wall has a heat flux component attached to it, as calculated from CFD simulation.

2.1.2. MASS IMBALANCE CORRECTION

CFD often carries slight imbalance between inflow and outflow mass flow rate associated to a CFD mesh, which is of same order close to the tolerance set by the CFD solver. This is acceptable within the error limits of a CFD simulation but could lead to major problems in CRN solution. The mass imbalance adds up into a significant error as CFD cells are clustered together to form reactors. In order to solve the CRN, this mass imbalance has to be corrected. Sampat [6] developed an approach to correct the mass flows of all reactor simultaneously by solving a linear system of equations of form $AX = B$ [4]. The independent variable for this system is the total outflow mass flow rates. The inflow into a reactor is defined as a fraction of the total outflow from the neighbouring reactors and this fraction is calculated using CFD data. The mass conservation equation is defined as total outflow should be equal to total mass inflow and is described in Equation 2.5.

$$M_k - \sum_{j\neq k}^{N_R} \alpha_{kj} M_j = \sum f_k \tag{2.5}$$

where M_k and M_j is the total outflow from k^{th} and j^{th} reactor, α_{kj} is the fraction of mass inflow from j^{th} reactor into k^{th} reactor and f_k is any external mass inflow. By rearranging the above equation, we arrive at Equation 2.6 of form $AX = B$. By solving this system of equation, the correct outflow from all reactors can be calculated.

$$\begin{bmatrix} \alpha_{11} & -\alpha_{12} & -\alpha_{13} & .. & .. \\ -\alpha_{21} & \alpha_{22} & -\alpha_{23} & .. & .. \\ -\alpha_{31} & -\alpha_{32} & \alpha_{33} & .. & .. \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \sum f_1 \\ \sum f_2 \\ \vdots \\ \vdots \end{bmatrix} \tag{2.6}$$

2.2. GOVERNING EQUATIONS

The goal of this project is solve the following steady-state governing equations and obtaining the solution only at the end (i.e $t = t_{end}$). Solution at intermediate time t ($0 < t < t_{end}$) is not of interest. For construction of chemical reactor network, the combustion region is replaced with a network of ideal reactors, preferably perfectly stirred Reactor (PSR). Let first assume each reactor is at fixed temperature and all reactants are perfectly mixed in the control volume. Each reactor is exchanging mass (by convection and diffusion) with the adjacent reactors and with external environment.

The mass balance for the i^{th} ($i = 1,...,N_S$) species in reactor k ($k = 1,...,N_R$) can described by equating the net output mass flow rate from reactor k with the sum of input mass flow rate into reactor k from neighbouring reactors and external environment plus the production term due to reaction, as shown in Equation 2.7.

$$M_k \omega_{ki} = \sum_{j=1}^{N_R} D_{kj} \omega_{ji} + f_{ki} + \Omega_{ki} V_k \tag{2.7}$$

In the equation above, M_k is the output mass flow rate from the k^{th} reactor, D_{kj} is mass flow rate entering reactor k from j^{th} reactor, f_{ki} is mass flow rate of species i entering reactor k from the external environment, V_k is the reactor volume, Ω_{ki} is the formation rate of species i (in $kg/sec/m^3$) and ω_{ki} is the mass fraction of species i in reactor k. Additionally, the inflow into a reactor k from reactor j (D_{kj}) is defined as a fraction of the total outflow from the neighbouring reactor j and this fraction is calculated as product of coefficient α_{kj} and total outflow from reactor j M_j, as shown in Equation 2.8. Furthermore, M_k and D_{kj} are extracted from the CFD simulation and hence stay constant throughout the solving process. These terms are related based on conservation of mass, as shown in Equation 2.9.

$$D_{kj} = \alpha_{kj} M_j \tag{2.8}$$

$$M_k = \sum_{j=1}^{N_R} D_{kj} + \sum_{i=1}^{N_S} f_{ki} \tag{2.9}$$

Equation 2.7 can be modified into a equation for net production rate of species i in reactor k (g_{ki}) which should be ideally zero at steady state, as shown below

$$g_{ki} = \sum_{j=1}^{N_R} (D_{kj} - M_k \delta_{kj}) \omega_{ji} + f_{ki} + \Omega_{ki} V_k = 0 \tag{2.10}$$

$$g(\omega) = -C\omega + f + R(\omega) = 0 \tag{2.11}$$

In a more compact form, Equation 2.11 describes the governing system of equations that are required to be solved for emission predictions, lets call it the RHS or residual function. This RHS function describes the net rate of production of all species in each reactor. In steady state, this should be equal to zero. Here, C is (N_EXN_E) sparse matrix containing all convection terms, N_E is total number of equations i.e N_E=$N_R * N_S$, f and R are (N_EX1) vectors corresponding to external feed and reaction terms. The transport contributions in the RHS function varies linearly with species concentration and is calculated using CFD simulation data. Non-linearity in the system is due to the reaction contributions.

For CRN, when energy conservation is considered, temperature is used as the state variable instead of specific enthalpy and internal energy. The additional equation for each reactor is shown in Equation 2.12. Using 2.13 and 2.14, the conservation equation for temperature is derived.

$$\frac{dU_k}{dt} = \sum_{in} \dot{m}_{in} h_{in})_k - \sum_{out} \dot{m}_{out} h_{out})_k - \dot{Q}_{wall,k} \tag{2.12}$$

$$U_k = m_k \sum_i \omega_{ki} u_{ki}(T) \tag{2.13}$$

$$\frac{dU_k}{dt} = u_k \frac{dm}{dt} + m_k c_{v,k} \frac{dT_k}{dt} + m_k \sum_i u_{ki} \frac{d\omega_{ki}}{dt} \tag{2.14}$$

$$m_k c_{v,k} \frac{dT_k}{dt} = \sum_{in} \dot{m}_{in} h_{in})_k - \sum_{out} \dot{m}_{out} h_{out})_k - \dot{Q}_{wall,k} - \sum_i V_k \Omega_{ki} u_{ki} \tag{2.15}$$

where U_k is internal energy in reactor k, m_k is the total mass in reactor k, T_k is the temperature in reactor k, $c_{v,k}$ and $c_{p,k}$ is specific heat capacity at constant volume and pressure resp. for reactor k, h is the specific enthalpy, $\dot{m}$ is the mass flow rate, $\dot{Q}_{wall}$ is the heat loss through the reactor walls with the external environment (only needed for reactors sharing a common wall with environment), Ω_{ki} is formation rate of species i in reactor k and u_{ki} is specific internal energy of i^{th} species in reactor k.

$$m_k c_{v,k} \frac{dT_k}{dt} = \sum_{j \neq k}^{N_R} D_{kj} c_{p,j} T_j - M_k c_{p,k} T_k - \kappa A_{wall} (T_k - T_{env}) - \sum_i V_k \Omega_{ki} c_{v,k_i} T_k \tag{2.16}$$

Here, κ is the heat transfer coefficient (W/m^2K), A_{wall} is area of the enclosure walls connecting the environment and T_{env} is environment temperature Rewriting Equation 2.16 in form of Equation 2.11:

$$g(T_k) = \sum_{j=1}^{N_R} (D_{kj} c_{p,j} - M_k c_{p,k} \delta_{kj}) T_j - \kappa A_{wall} (T_k - T_{env}) - \sum_i V_k \Omega_{ki} c_{v,k_i} T_k = 0 \tag{2.17}$$

Therefore, this equation is solved along species mass conservation for every reactor. The total number of equations N_E now becomes $N_R * (N_S + 1)$. With energy ON. N_R additional equations need to be solved. Both size of C, f and R is increased with respect to the new N_E.

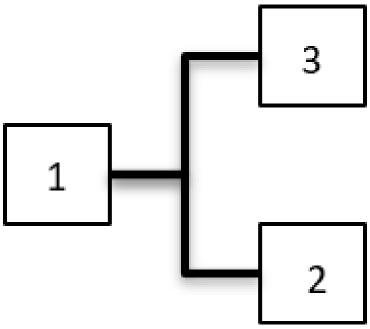

Figure 2.2: Three reactors system

In order to explain above described governing equation, let consider a simplified reactor network as shown in Figure 2.2. This systems consists of three reactors ($N_R = 3$) each containing only three species ($N_S = 3$). There is mass inflow and outflow between reactor 1 and 2 and reactor 1 and 3. Based on this, the nonlinear system of equation ($N_E = 9$) becomes:

$$-C\begin{bmatrix}\omega_{11}\\ \omega_{12}\\ \vdots\\ \omega_{33}\end{bmatrix}+\begin{bmatrix}f_{11}\\ f_{12}\\ \vdots\\ f_{33}\end{bmatrix}+\begin{bmatrix}\Omega_{11}V_1\\ \Omega_{12}V_1\\ \vdots\\ \Omega_{33}V_3\end{bmatrix}=0 \tag{2.18}$$

In Equation 2.18, C is a 9X9 matrix and its sparsity is described in Equation 2.19

$$-C=\begin{bmatrix}-M_1 & 0 & 0 & D_{12} & 0 & 0 & D_{13} & 0 & 0\\ 0 & -M_1 & 0 & 0 & D_{12} & 0 & 0 & D_{13} & 0\\ 0 & 0 & -M_1 & 0 & 0 & D_{12} & 0 & 0 & D_{13}\\ D_{21} & 0 & 0 & -M_2 & 0 & 0 & 0 & 0 & 0\\ 0 & D_{21} & 0 & 0 & -M_2 & 0 & 0 & 0 & 0\\ 0 & 0 & D_{21} & 0 & 0 & -M_2 & 0 & 0 & 0\\ D_{31} & 0 & 0 & 0 & 0 & 0 & -M_3 & 0 & 0\\ 0 & D_{31} & 0 & 0 & 0 & 0 & 0 & -M_3 & 0\\ 0 & 0 & D_{31} & 0 & 0 & 0 & 0 & 0 & -M_3\end{bmatrix} \tag{2.19}$$

$$-C=\begin{bmatrix}-M_1 & D_{12} & D_{21}\\ D_{21} & -M_2 & D_{23}\\ D_{31} & D_{32} & -M_3\end{bmatrix}\otimes\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{2.20}$$

Equation 2.20 shows a compressed version of C as Kronecker product of reactors mass flow rate matrix with positive sign convention for inflow and $N_S X N_S$ identity matrix. Now if we consider recalculation of temperature for each reactor, three additional equations are added to the system. For simplicity, lets assume no heat transfer with external walls. For this case, size of vectors and matrices increased to $N_R * (N_S + 1)$. These update equations are shown in Equation 2.21.

$$-C' \begin{bmatrix} \omega_{11} \\ \omega_{12} \\ \omega_{13} \\ T_1 \\ \vdots \\ \omega_{33} \\ T_3 \end{bmatrix} + \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ 0 \\ \vdots \\ f_{33} \\ 0 \end{bmatrix} + \begin{bmatrix} \Omega_{11} V_1 \\ \Omega_{12} V_1 \\ \Omega_{13} V_1 \\ \sum_{i=1}^{3} V_1 \Omega_{1i} c_{v,1_i} T_1 \\ \vdots \\ \Omega_{33} V_3 \\ \sum_{i=1}^{3} V_3 \Omega_{3i} c_{v,3_i} T_3 \end{bmatrix} = 0 \tag{2.21}$$

$$-C' = \begin{bmatrix} -M_1 & 0 & 0 & 0 & D_{12} & 0 & 0 & 0 & D_{13} & 0 & 0 & 0 \\ 0 & -M_1 & 0 & 0 & 0 & D_{12} & 0 & 0 & 0 & D_{13} & 0 & 0 \\ 0 & 0 & -M_1 & 0 & 0 & 0 & D_{12} & 0 & 0 & 0 & D_{13} & 0 \\ 0 & 0 & 0 & -M_1 c_{p,1} & 0 & 0 & 0 & D_{12} c_{p,2} & 0 & 0 & 0 & D_{13} c_{p,3} \\ D_{21} & 0 & 0 & 0 & -M_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D_{21} & 0 & 0 & 0 & -M_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{21} & 0 & 0 & 0 & -M_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{21} c_{p,1} & 0 & 0 & 0 & -M_2 c_{p,2} & 0 & 0 & 0 & 0 \\ D_{31} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -M_3 & 0 & 0 & 0 \\ 0 & D_{31} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -M_3 & 0 & 0 \\ 0 & 0 & D_{31} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -M_3 & 0 \\ 0 & 0 & 0 & D_{31} c_{p,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -M_3 c_{p,3} \end{bmatrix}$$

2.3. JACOBIAN CALCULATION

An accurately calculated Jacobian is critical for the solution process as it is required by both time integration and Newton's method for fast convergence. This section provides information on how this Jacobian is analytically calculated.

2.3.1. ONLY TRANSPORT CONTRIBUTIONS WITH ENERGY OFF

Lets first assume only transport terms are present in the RHS function i.e $g_{ki}(\omega) = -C\omega + f = \sum_{j=1}^{N_R} (D_{kj} - M_k \delta_{kj}) \omega_{ji} + f_{ki}$. Also, the Jacobian matrix of size $N_E X N_E$ is decomposed into $N_R X N_R$ square matrices J_{kj} of size $N_S X N_S$.

$$J_{kj} = \begin{cases} -M_k I & \text{if } k = j \\ D_{kj} I & \text{if } k \neq j \end{cases} \tag{2.22}$$

$$J_S = -C \tag{2.23}$$

J_S is the transport Jacobian matrix. It is sparse matrix containing terms in governing equations related to inter-connectivity between reactors. It is sparse because each reactor is connected to only few other reactors. It is basically equal to negative of C matrix from Equation 2.11.

2.3.2. ONLY REACTION CONTRIBUTIONS WITH ENERGY OFF

If only reaction terms are considered, J_R (reaction Jacobian matrix) contains terms related species production in the reactor itself. J_R is sparse and block diagonal which is because only species inside each reactor have effect on their respective species production and reactions. It is calculated using a simplified analytical expression with parameters, such as reaction rate and concentration, calculated using Cantera library. It involves varying each reactant and product, and calculating the change in reaction rate for each reaction.

$$J_R = diag(J_{\dot{\Omega}_1}, J_{\dot{\Omega}_2}, ..., J_{\dot{\Omega}_{N_R}}) \tag{2.24}$$

J_R is a block diagonal matrix that comprises of multiple $J_{\dot{\Omega}}$ square sub-matrices along the diagonal, each sub-matrix is sparse and belong to a single reactor. Equation 2.25 splits the sub-matrix expression into two terms. The first terms (A matrix) depicts the change in species production with change in all species concentration whereas the second term (Γ matrix) relates the change in all species concentration with that one particular species concentration.

$$J_{\dot{\Omega}} = \frac{\partial \dot{\Omega}}{\partial Y} = \frac{\partial \dot{\Omega}}{\partial C}\frac{\partial C}{\partial Y} = A\Gamma \tag{2.25}$$

Lets consider a general chemical reaction,

$$\nu'_1 S_1 + \nu'_2 S_2 + ... + \nu'_{n1} S_{n1} \rightleftharpoons \nu''_1 S_1 + \nu''_2 S_2 + ... + \nu''_{n2} S_{n2} \tag{2.26}$$

The net reaction rate (in $kmol/m^3 s$) for this reaction is calculated using Cantera, using Equation 2.27. k' and k" are rate constants, estimated using the Arrhenius equation 2.28.

$$r = r_f - r_b = k' \prod_{i=1}^{n1} C_i^{\nu'_i} - k'' \prod_{j=1}^{n2} C_j^{\nu''_j} \tag{2.27}$$

$$k = A(T)\exp(\frac{E_a}{RT}) \tag{2.28}$$

The net rate of increase in species $\dot{\Omega}$ (kg/s) is obtained as a product of constant α, reaction rate r, stochiometric coefficient ν_i and volume of the reactor V_r.

$$\dot{\Omega}_i = \alpha_i r_i \nu_i V_r \tag{2.29}$$

$$\alpha_i = \begin{cases} -1, & \text{if i is a reactant} \\ 1, & \text{if i is a product} \end{cases}$$

The derivative of r with respect to reactant species concentration C_i is given as

$$\frac{\partial r}{\partial C_i})_{reactant} = \nu'_i k' C_i^{\nu'_i-1} \prod_{j=1, j\neq i}^{n1} C_j^{\nu'_j} = \frac{\nu'_i}{C_i} r_f \tag{2.30}$$

Similarly, derivative of r with respect to product species is given as

$$\frac{\partial r}{\partial C_i})_{product} = -\frac{\nu''_i}{C_i} r_b \tag{2.31}$$

’A’ matrix in equation 2.25 is expressed as a summation of changes due to different reaction (1 to I) in the reactor.

$$A = \frac{\partial \dot{\Omega}}{\partial C} = \sum_{l=1}^{I} S_l \frac{\partial r_l}{\partial C} \tag{2.32}$$

where S_l is the contribution of coefficients related to the l^{th} reaction. Concentration of species is expressed as the ratio of number of moles of that species to the total volume.

$$C_i = \frac{n_i}{V} = \frac{Y_i m_T}{V M W_i} \tag{2.33}$$

where Y_i is the mass fraction species i, m_T is the total mass of all species and V is the volume. Therefore,

$$\Gamma = \frac{\partial C}{\partial Y} = \frac{m_T}{V M W_i} \tag{2.34}$$

This analytical method of Jacobian calculation is faster as the evaluation of rate constant needs to be done only once at the beginning of the iteration whereas in the numerical Jacobian, the rate constants need to be re-evaluated for change in every species [6].

If the reaction term is considered along with transport contributions, Jacobian becomes

$$J_{kj} = \begin{cases} -M_k I + J_{\dot{\Omega}} & \text{if } k = j \\ D_{kj} I & \text{if } k \neq j \end{cases} \tag{2.35}$$

$$J = J_S + J_R \tag{2.36}$$

From above equations, it is clear that Jacobian evaluation can be split into two matrices J_S and J_R. Both J_S and J_R Jacobian matrices are stored as Compressed Sparse Row (CSR) matrix. Figure 2.3 and 2.4 shows sparsity pattern in both transport and reaction Jacobian. Please note, diagonal blocks in J_R are not dense but sparse.

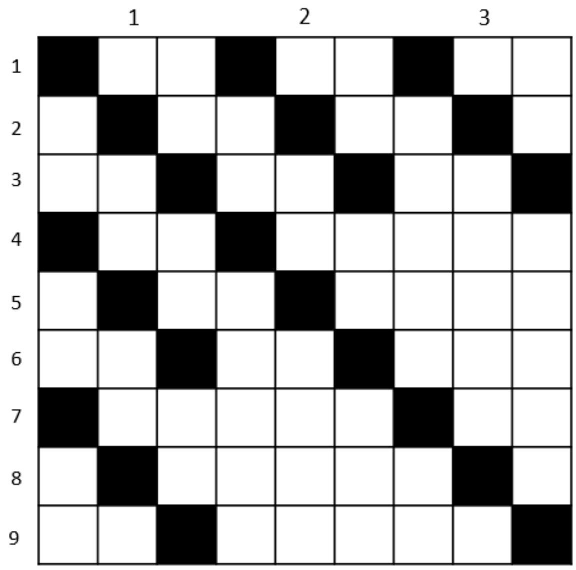

Figure 2.3: Transport Jacobian (J_S) sample structure accounting for reactor inter-connectivity, based on Figure 2.2 configuration and energy (temperature recalculation) OFF

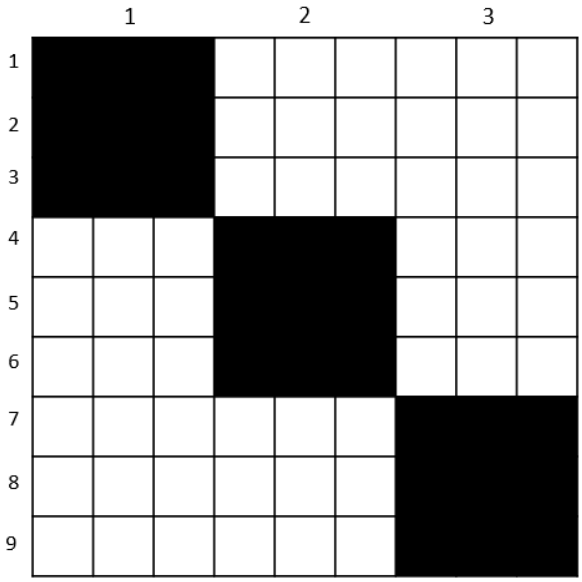

Figure 2.4: Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Figure 2.2 configuration and energy (temperature recalculation) OFF

2.3.3. ENERGY ON

In case of energy equation ON, an additional equation is solved for each reactor. Hence, new components are needed to be added in the Jacobian. J_S Jacobian matrices is updated by adding a diagonal entry for each sub matrices with

$$J_S^{(kj)}{}_{added} = D_{kj} c_{p,j} - M_k c_{p,k} \delta_{kj} \tag{2.37}$$

J_R Jacobian matrix requires several additional terms related to addition of an extra equation. Terms in the sub-matrix for reactor k are expressed as

$$[J_R^{(k)}]_{ij} = \frac{\partial R_j^{(k)}}{\partial \omega_{ki}} \tag{2.38}$$

As T_k is an added variable in ω_k vector and temperature equation is added to the system of non-linear equations, $2N_S+1$ new entries are added to all sub-matrices of the reaction Jacobian. For example, the additional term in the main diagonal is given below in Equation 2.39.

$$[J_R^{(k)}]_{ii,added} = -\kappa A_{wall} - \sum_{i} V_k(\Omega_{ki} c_{v,k_i} + c_{v,k_i} T_k \sum_{m=1}^{I} \nu_{mi}(\frac{\beta}{T_k} + \frac{E_{am}}{R{T_k}^2}) r_m) \quad (2.39)$$

Figure 2.5 and 2.6 presents the updated Jacobian structure if re-calculation of temperature is selected. The gray blocks are the added terms and it is confirmed that the sparsity pattern does not change with addition of N_R energy equations.

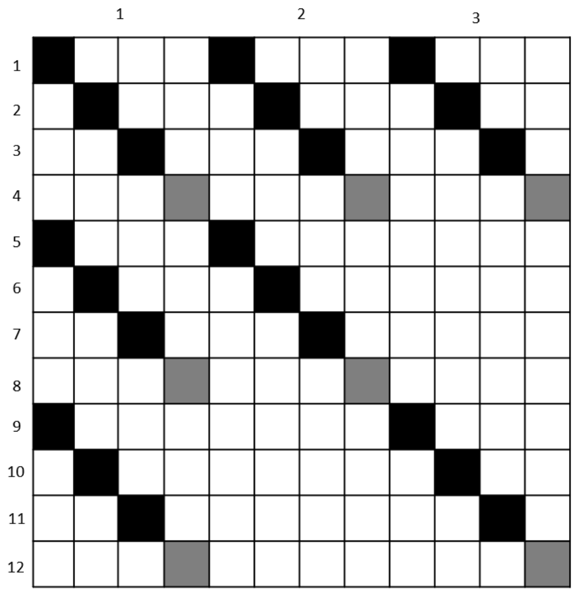

Figure 2.5: Transport Jacobian (J_S) sample structure accounting for reactor inter-connectivity, based on Figure 2.2 configuration and energy (temperature recalculation) ON. Gray blocks are additional Jacobian contributions due to solving the energy equation for each reactor

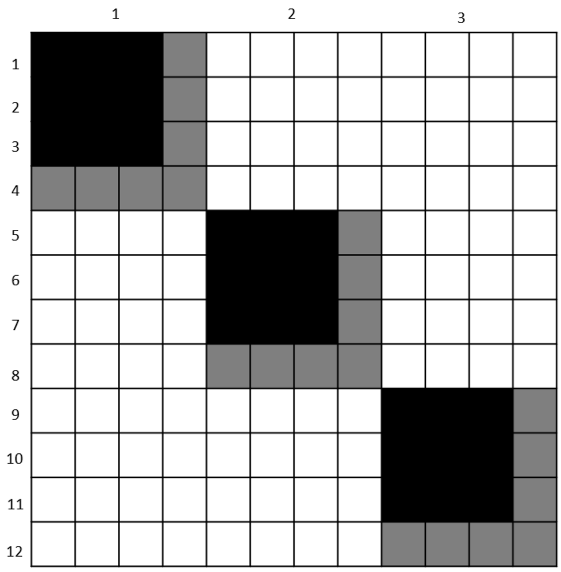

Figure 2.6: Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Figure 2.2 configuration and energy (temperature recalculation) ON. Gray blocks are additional Jacobian contributions due to solving the energy equation for each reactor

2.4. PETSC: ODE AND SNES SOLVER

Solution of CRN requires solving a sparse nonlinear system of equations and highly stiff ordinary differential equations. Several numerical solver libraries with faster computation and better efficiency are widely available. Some of the best known solver suites capable of solving these problems in Python environment are SUNDIALS [24], PETSc [1], Trilinos [25] and Python's SciPy. However, this project will extensively use PETSc [1] solver library developed by Argonne National Laboratory. Portable, Extensive Toolkit for Scientific Computation (PETSc) [1][7] is a collection of state-of-the-art algorithm and data structures that provide sequential as well as parallel solution of scientific problems. This object-oriented numerical solver supports Message Passing Interface (MPI) [26], Graphical Processing Units (GPUs) through CUDA or OpenCL and also MPI-GPU parallelism. It is fully usable in C, C++, Fortran, Python and MATLAB (sequential). For the context of this research work, PETSc [1] is selected because PETSc offers a wide collection of intuitive, dynamic and user-friendly modules with their respective routines to build a perfect solver to tackle CRN governing equations in form of stiff ODEs and sparse nonlinear system of equations. Unlike other solver suites, PETSc allows more flexibility in form of wide selection of ODE solvers, linear and nonlinear solvers with preconditioners along with interfaces to external software. It includes everything that is currently offered by the other three solver suites. PETSc's careful design allows users to have detailed control over the entire solution process. SUNDIALS [24] uses CVODE time integration package which only supports dense, diagonal or banded sparse matrices [27]. Banded matrix is a form of sparse matrix where non-zero entries are confined to the diagonal band. In case of AGNES, non-banded sparse matrix is involved as evident for the sparsity pattern of the transport Jacobian. PETSc supports such non-banded sparse matrices, hence is the correct choice for this project. Furthermore, all PETSc libraries are designed to work in both sequential and parallel environment. Although other solver suites are easier to learn, PETSc will probably pay off more in the long run when parallel computation will be explored.

However, the use of PETSc with AGNES comes with its own problems. AGNES is a Python based tool and PETSc is employed using a Python wrapper called petsc4py [28]. PETSc for Python provides unrestricted access to most of PETSc libraries and functionalities in Python environment. However, this Python binding for PETSc is still not fully developed as it is missing recent updates, newly developed modules, such as DMNetwork [29] and provides almost no support material for the first time users. Most of the commands, originally written in C are not working or are completely scrapped for the Python version like 'log_summary' command in C. Moreover, the potential of PETSc in speeding up AGNES's solution process is not fully realised in current Python environment, as it is inherently slower compared to C language. In future, atleast the solver section of AGNES has to be re-written in C for accessing the full potential of PETSc.

Balay et al. (2020) [7] provides PETSc user manual detailing how to run PETSc efficiently. The ODE integrator library includes explicit solvers, implicit solvers as well as a collection of implicit-explicit solvers. User enjoys tremendous flexibility in selection of linear and nonlinear solvers which is supported by PETSc extensive suite of iterative solvers and direct solvers. Finally, PETSc is designed to be highly modular by enabling high compatibility with several specialized parallel libraries.

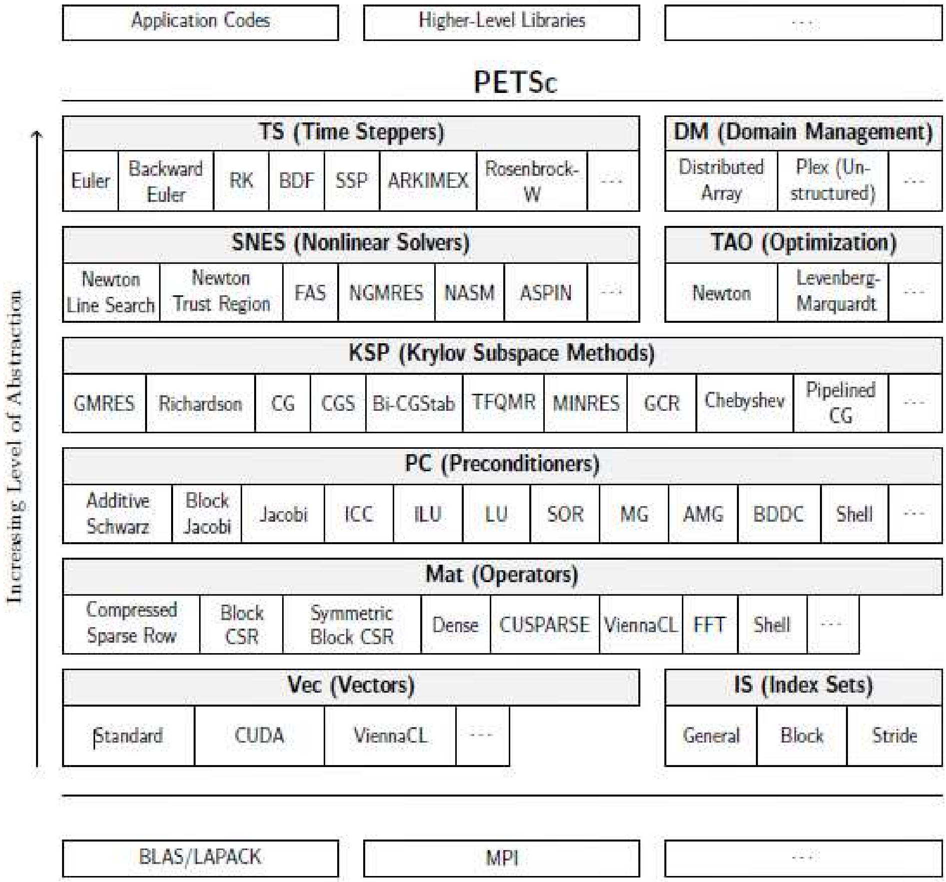

Figure 2.7: Numerical libraries of PETSc [7]

PETSc consists of various libraries where each manipulate a particular set of objects and their respective operations [30]. Figure 2.7 shows PETSc hierarchical organization. This enables users to freely design re-usable and flexible solver for a particular problem. Some of the modules provided in PETSc are:

- Index sets (IS)
- Vectors (Vec) and matrices (Mat)
- Linear Solvers: Krylov Subspace Methods (KSP)
- Preconditioners (PC)
- Nonlinear solvers (SNES)
- Time-stepping solvers (TS)
- Data management (DM) for parallel structured and unstructured meshes

2.4.1. Index Sets

Index Set (`IS`) is a collection of a set of integer indices, which is used to define several operations of vectors and matrices. In most of general routines, `IS` object refers to the variables that contains the particular indices in a vector or matrix. It is integral for defining vector scatters and gathers. Scatters and gathers are type of memory addressing for vectors that allows inserting any subset of one vector into another vectors. This operation is a combination of scatters and gathers, and is referred as generalized scatters.

2.4.2. Vectors

Vectors (`Vec`) is one of the simplest object in PETSc that is used to store, access and manipulate vectors corresponding to the solution of system of nonlinear equations or ODEs. `Vec` may also used to describe adaptive grid parameters and quadrature information. PETSc can create a Vec object either in sequential or parallel (MPI based) manner. Some of the basic routines with their respective description is provided below:

- **Vector creation and assembly:** Routines for creating vector objects are `Vec.createSeq()` for sequential Vector and `Vec.createMPI()` for parallel Vector. However, the user can either specify the number of components that will be stored on each process or let PETSc decide. More general routine for creating a Vector is `Vec.create()` where the user's input is used to generate the appropriate Vector type (sequential or parallel) over all processes in communication. After creation, the vector object is then filled in by using `Vec.setValues()` routine that requires arrays of index and their associated values as inputs. After the object is defined, vector assembly is done through `Vec.assemble()` function.

- **Vector access and update:** Some routines are dedicated for 'Read-Only' or 'Read-Write' access of vectors. This is done using `Vec.getArray()` with readonly Boolean input set either at False or True. After updating the vector entries, this should be followed by routine `Vec.restoreArray()` to record any changes.

- **Vector operations:** PETSc also provides most of the basic arithmetic operations like addition, multiplication, etc. along with norm computations. However, these operations are limited to same vector type, that means operations between parallel and serial vector are not permitted. A detailed list of all available PETSc vector operations is provided in PETSc manual [7].

2.4.3. MATRICES

PETSc supports both dense and sparse matrix type in sequential and parallel versions. By default, PETSc generates a sparse AIJ format (also called the Yale sparse matrix format or compressed sparse row format, CSR). Most matrices involved in this project are also sparse. A sparse matrix has most of its element as zero. In contrast, if most of the elements are non-zero, then it is a dense matrix. The most common way of storing sparse matrix is Compressed Sparse Row (CSR) format. As the name suggest, this format compresses the row indices. To store a sparse matrix in this format, four parameters are required: number of non-zero entries (NNZ), 1D array of values (V), Column indices (COL) and row indices (ROW). Here, the row indices array encodes the index in V and COL where the given row starts. The first element is always zero and the last element is NNZ. For a matrix of size m x n, length of array of values and column indices is NNZ whereas length of row array is m+1. As different rows might have very different numbers of non-zeros, array of size m, called nnz_array, containing number of non-zero entries for each column is also required. Lets consider an example, assume A is a sparse matrix as shown below.

$$A = \begin{bmatrix} 10 & 20 & 0 & 0 & 0 \\ 0 & 30 & 40 & 0 & 0 \\ 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 80 \end{bmatrix} \tag{2.40}$$

A is 4x5 matrix (20 entries) and has 8 non-zero entries (NNZ=8). The four 1D arrays are the following:

Array	Content	Size
V	[10,20,30,40,50,60,70,80]	8X1
COL	[0,1,1,2,1,2,3,4]	8X1
ROW	[0,2,4,7,8]	5X1
nnz_array	[2,2,3,1]	4X1

After having correct estimations of these 4 1D arrays, a CSR matrix is constructed in PETSc by using `Mat.setValuesCSR(ROW, COL, V)`. This is usually preceded by pre-allocating number of non-zero entries present in each row by using `Mat.setPreallocationNNZ(nnz_array)`. Preallocation is essential for efficient memory allocations. If the nnz_array is accurate, this assembly will not require any additional memory allocations but if the nnz estimates are incorrect, PETSc will automatically obtain additional space at a slight loss of efficiency.

A typical PETSc matrix creation can be broken into three steps:

1. `Mat.create()`: Command to begin creation of a PETSc matrix with size and matrix type altered using `Mat.setSizes()` and `Mat.setType()`. Shortcut methods are also present to build specific types of matrices, for example, using `Mat.createSEQAIJ()` or `Mat.createDense()`, sparse AIJ matrix or dense matrix can be easily created.
2. `Mat.setValues()`: A quick way to fill-in non-zero entries. The most common approach is to input previous discussed 1D arrays for CSR creation. But it will always require memory preallocation for efficient matrix construction (discussed in earlier).
3. `Mat.assemble()`: Matrix creation and fill-in is followed by this command to save changes made to the matrix and to make it ready for future use.

2.4.4. Linear System Solvers

A system of linear equations can be solved by either direct methods such as Gauss Elimination or iterative method like preconditioned Krylov Subspace methods (KSP). Direct methods compute solution in finite number of steps. This is done using factorization methods such as Gauss Elimination and QR Factorization. Iterative methods start from an initial guess and converge to the exact solution using approximations. Iterative methods are ideal for complex systems. PETSc provides a user-friendly set of routines to set both direct and iterative solvers in sequential as well as parallel manner. Linear System Solver library, referred as `KSP` object, is intended for solving a non-singular system of linear equation such as

$$Ax = b \tag{2.41}$$

where A is matrix representation of linear system, b is a non-zero RHS vector, n is the length of the vector and x is the solution vector. Krylov Subspace method with preconditioners is the most preferred iterative solution of linear systems.

Krylov Subspace Methods

In Krylov Subspace methods (KSP), a j-order Krylov subspace is generated from n-by-n A matrix and n dimension b vector as function of first j powers of A, as shown below.

$$K_j(A,b) = span(b, Ab, A^2b, ..., A^{j-1}b) \tag{2.42}$$

The solution $x = A^{-1}b$ can be expressed as a product of polynomial of A and vector b, i.e. $x = p(A)b$. Krylov subspace methods approximate the solution x with

$$x_j \in x_0 + K_j(A, r_0) \quad , r_0 = b - Ax_0 \tag{2.43}$$

$$x_j = p_j(A) r_0 = \sum_{i=0}^{j-1} \alpha_i^{(j)} A^i r_0 \tag{2.44}$$

The choice of polynomial $p_j(A)$ varies for different methods. To solve a linear system with `KSP`, the user needs to create a solver context and later set the matrices and vectors involved in the system.

By default, restarted GMRES is used as KSP solver, preconditioned with Block Jacobi method in multi-process case and Incomplete LU (ILU) Factorization in uni-process case.

Method	KSPType
Richardson	`RICHARDSON`
Chebyshev	`CHEBYSHEV`
Conjugate Gradient	`CG`
Biconjugate Gradient	`BICG`
Biconjugate Gradient Stabilized	`BCGS`
Generalized Minimal Residual (GMRES)	`GMRES`
Minimal Residual Method	`MINRES`
Conjugate Residual	`CR`
Generalized Conjugate Residual	`GCR`

Table 2.1: List of PETSc linear solvers

CONJUGATE GRADIENT (CG) METHOD

Conjugate Gradient method is only applicable only when A is symmetric ($A{=}A^T$) and positive definite ($x^T Ax > 0$ for all non zero x). Staring from initial guess x_0 and initial descent direction $p_0 = r_0 = b - Ax_0$, iterative solution of x can be found as

$$x_{n+1} = x_n + \alpha_n p_n \tag{2.45}$$

$$r_{n+1} = r_n - \alpha_n A p_n \tag{2.46}$$

$$p_{n+1} = r_{n+1} + \beta_{n+1} p_n \tag{2.47}$$

where α_n and β_n are chosen to minimize the norm of error $||e_{n+1}||^2 = ||x_{n+1} - x||^2$. This provides the expression for α_n and β_n.

$$\alpha_n = \frac{(r_n, r_n)}{(Ap_n, p_n)} = \frac{||r_n||_2^2}{||p_n||_A^2} \tag{2.48}$$

$$\beta_{n+1} = \frac{||r_{n+1}||_2^2}{||r_n||_2^2} \tag{2.49}$$

Unlike CG, Biconjugate Gradient (BICG) and Biconjugate Gradient Stabilized (BCGS) method do not require A to be symmetric and self-adjoint. Moreover, both BICG and BCGS have faster and smoother convergence than CG.

GENERALIZED MINIMAL RESIDUAL (GMRES) METHOD

Generalized Minimal Residual (GMRES) method is ideal for large, unsymmetric stiff systems. At the j^{th} iteration, it minimizes residual $||r_j||$, where $r_j = b - Ax_j$, in $K_j(A, b)$. This done by finding an optimal j-order polynomial $p_j(A)$ such that $r_j = p_j(A)r_0$ and $||r_j||$ is minimized. It supports both right and left preconditioners. This method is used for this project with ILU preconditioner with Approximate Minimum Degree (AMD) reordering and element fill-ins during factorization.

PRECONDITIONERS WITH KRYLOV METHOD

Preconditioners are used for linear systems to accelerate the convergence rate and to reduce the condition number of the system. Preconditioners are applied by using preconditioner matrices M_L and M_R as shown below.

$$M_L^{-1} A M_R^{-1} (M_R x) = M_L^{-1} b \tag{2.50}$$

Preconditioner Method	PCType
Jacobi	`JACOBI`
Block Jacobi	`BJACOBI`
Incomplete Cholesky	`ICC`
Incomplete LU	`ILU`
Additive Schwarz	`ASM`
Generalized Additive Schwarz	`GASM`

Table 2.2: List of PETSc preconditioners

Jacobi and Block Jacobi are diagonal preconditioner matrices and are only applicable if A is symmetric positive definite. Incomplete LU (ILU) method uses identity matrix as right preconditioner ($M_R = I$) and left preconditioner matrix as $M_L = LU$, where L is a lower triangular matrix and U is an upper triangular matrix such that LU is a good approximation of A. These matrices have a prescribed nonzero pattern. After using ILU preconditioner, the system of linear equations become

$$U^{-1} L^{-1} A x = U^{-1} L^{-1} b \tag{2.51}$$

Algorithm 1 presents the basic steps to define and solve a linear system of equations of form $Ax = b$. Some of the important commands available in PETSc KSP class are-

Algorithm 1 PETSc KSP routine to solve linear equations system (Equation 2.41)

```
1. Create KSP object using KSP.create()
2. Define vectorsize n, RHS vector b, solution vector x
3. Define matrix A and set it as the linear operator using KSP.setOperators(A)
4. Select KSP solver type and suitable preconditioner
5. Start solution using KSP.solve(b,x)
6. Monitor progress and convergence using KSP.setMonitor()
```

- `KSP.create()`: Command needed to create KSP solver context
- `KSP.setOperators(A)`: Command to define the matrix A associated with the linear system
- `KSP.setFromOptions()`: Initialize solver parameters like tolerances and norm setting
- `KSP.setType()`: Set the correct KSP solver scheme. Also used along with `KSP.getPC()` and `PC.setType()` to define and select an appropriate preconditioner
- `KSP.solve(b,x)`: Set the RHS vector b and solution vector x, and start the solution process

2.4.5. NONLINEAR SYSTEM SOLVERS

PETSc nonlinear solvers are handled by `SNES` library. Scalable Nonlinear Equations Solvers (SNES) library provides set of routines for solving nonlinear system of equations. Newton-like method, including both line search and trust region technique, and nonlinear Krylov method constitute the core of this module. `SNES` module is used for solving non-linear system such as

$$F(x) = 0 \tag{2.52}$$

Using n-dimensional Newton's method, equation 2.52 is reduced to

$$x_{k+1} = x_k - J(x_k)^{-1} F(x_k), \quad k = 0, 1, ... \tag{2.53}$$

In equation 2.53, $J(x_k)$ refers to the Jacobian matrix which is non-singular at each iteration and expressed as $J(x_k) = F'(x_k)$. x_0 is the user's initial guess.

Newton line search method is the default `SNES` solver but user can change the solver type. Additionally, the user need to add custom routines for evaluationg the function, given in equation 2.52 and its associated Jacobian. Matrix free

Jacobian and finite difference Jacobian approach can also be used. PETSc user manual [7] provides detailed routines for function and its Jacobian evaluation.

Method	SNESType
Line Search Newton	`NEWTONLS`
Trust Region Newton	`NEWTONTR`
Nonlinear Richardson	`NRICHARDSON`
Nonlinear CG	`NCG`
Nonlinear GMRES	`NGMRES`
Quasi-Newton	`QN`
Nonlinear ASM	`NASM`

Table 2.3: List of PETSc nonlinear solvers

LINE SEARCH AND TRUST REGION METHOD

Newton line search and trust region method are some of the widely used globalised nonlinear solvers. In line search method, search direction d_k is computed by utilizing the exact approximation for the Jacobian. Thus, d_k is computed by solving a linear system of equations which is given as 2.54.

$$J(x_k)d_k = -f(x_k) \tag{2.54}$$

And the next iteration x_{k+1} is expressed in terms of the search direction and a line search parameter α_k. α_k is chosen such that $\alpha_k > 0$ and $||f(x_{k+1})|| < ||f(x_k)||$.

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.55}$$

On the other hand, trust region method uses an assumption that for a small step-size h, the function $f(x_{k+1}) = f(x_k + h)$ can be reduced to a linear expression i.e.

$$f(x_k + h) = f(x_k) + J(x_k)h \tag{2.56}$$

This method uses B_k as an approximate for the Jacobian and solves for the correct step-size. This is done by solving an optimization problem, expressed as

$$min[||f(x_k) + B_k h|| : ||D_k h|| \leq \Delta_k] \tag{2.57}$$

where D_k is a scaling matrix and Δ_k is the trust-region radius. The step-size is accepted if the ratio in equation 2.58 is greater than user defined σ. If step-size is not accepted, trust region radius is decreased and ratio is re-computed.

$$\rho_k = \frac{||f(x_k)|| - ||f(x_k + h)||}{||f(x_k)|| - ||f(x_k) + B_k h||} \tag{2.58}$$

Algorithm 2 PETSc SNES routine to solve system of nonlinear equations (Equation 2.52)

```
1. Create SNES object using SNES.create()
2. Define vectorsize n, solution vector x
3. Specify routines to evaluate residual function $F(x)$ and its associated Jacobian $J(x)$ using SNES.setFunction() and SNES.setJacobian()
4. Select SNES solver type and suitable linear solver routines
5. Start solution using SNES.solve(None,x)
6. Monitor progress and convergence using SNES.setMonitor()
```

Algorithm 2 presents the basic steps to define and solve a nonlinear system of equations of form $F(x) = 0$. Some of the important commands available in PETSc SNES class are-

- `SNES.create()`: Command needed to create SNES solver context
- `SNES.setFunction()`: Command to specify a routine to evaluate the residual function $F(x)$ at the current solution x
- `SNES.setJacobian()`: Command to specify a routine to form some approximation of the Jacobian matrix $J(x)$ at the current solution x
- `SNES.setSolution()`: Command to set the initial guess for the solving process
- `SNES.setFromOptions()`: Provides an interface to set solver parameters, convergence tolerance, monitoring routines and customised routines (like specialised line search variants). Some parameters can be altered specifically such as tolerances can be changed by using `SNES.setTolerances()`.
- `SNES.setType()`: Set preferred SNES solver scheme (default choice in Newton's line search). Also it is generally used along with `SNES.getKSP()` and `KSP.getPC()` to control and define linear solver routines with appropriate preconditioner
- `SNES.solve(None,x)`: Set solution vector x, and start the solution process. Using `SNES.setMonitor()`, the solving process and converged reason can be actively monitored.

2.4.6. TIME-STEPPING SCHEMES

PETSc provides several time-stepping routines (`TS`) for both time-dependent explicit and implicit ODEs. PETSc interface assumes the following form for time dependent ODEs.

$$\dot{u} = f(t, u), \quad u(t_0) = u_0 \tag{2.59}$$

In the above equation, $f(t, u)$ is referred as Right Hand Side (RHS) function. To define such a time dependent ODE problem, the user is required to provide routines to correctly evaluate the RHS function (`TS.SetRHSJacobian`) and its associated Jacobian (`TS.SetRHSJacobian`) at every time step in order to successfully setup time integrator. Table 2.4 shows different PETSc time integration methods, few of which are discussed in detail later in this section.

`TSType`	Method	Type
euler	Forward Euler	explicit
ssp	Multistage Strong Stability Preserving [31]	explicit
beuler	Backward Euler	implicit
alpha(2)	alpha-method [32]	implicit
glle	General Linear	implicit
eimex	Extrapolated IMEX	adaptive
arkimex	IMEX Runge-Kutta	IMEX
rosw	Rosenbrock-W	implicit
glee	General linear with Global Error Estimation	explicit/implicit
bdf	Standard Backward Differentiation Formula	implicit
sundials	Standard Backward Differentiation Formula	implicit

Table 2.4: List of PETSc time integration methods

BACKWARD DIFFERENTIATION FORMULAS (BDF) METHOD

It is a low-order (1-6) multi-step implicit time integration method, mostly suitable for stiff ODEs problems. It computes differential of solution u as a linear combination of past(backward) values i.e.

$$\dot{u}_n = f(u_n, u_{n-1}, \dots u_1, u_0) \tag{2.60}$$

$$u_n = \sum_{i=1}^{k} \alpha_i u_{n-i} + h\beta_0 \dot{u}_n \tag{2.61}$$

where α_i and β_0 depends on the method order m and h is the step size.

Three possible step-sizes are allowed for BDF method: interpolated fixed-step BDF, fully variable-step BDF, and fixed-leading coefficient BDF. Additionally, BDF methods can be expressed simply based on order k.

Implicit backward Euler method:

$$k = 1: \quad u_{n+1} = u_n + h\dot{u}_{n+1} \tag{2.62}$$

BDF-2 method:

$$k = 2: \quad u_{n+1} = \frac{4}{3} u_n - \frac{1}{3} u_{n-1} + \frac{2}{3} h \dot{u}_{n+1} \tag{2.63}$$

and

$$k = 3: \quad u_{n+1} = \frac{18}{11} u_n - \frac{9}{11} u_{n-1} + \frac{2}{11} u_{n-2} + \frac{6}{11} h \dot{u}_{n+1} \tag{2.64}$$

Algorithm 3 PETSc TS routine to solve ODEs (Equation 2.59)

```
1. Create TS context using TS.create()
2. Define vectorsize n, solution vector u
3. Specify routines to evaluate RHS function $G(t, u)$ and its associated Jacobian using TS.setRHSFunction() and TS.setRHSJacobian()
4. Select TS problem type and time stepping scheme along with suitable nonlinear and linear solver routines
5. Set time step size, maximum time and maximum steps for the solver
6. Start solution using TS.solve(u)
7. Monitor progress and convergence using TS.setMonitor()
```

Algorithm 3 presents the basic steps to define and solve a nonlinear system of equations of form $F(t, u, \dot{u}) = G(t, u)$. Some of the important commands available in PETSc TS class are-

- `TS.create()`: Command needed to create TS solver context
- `TS.setProblemType()`: Command to choose the problem type between linear and nonlinear
- `TS.setRHSFunction()`: Command to specify a routine to evaluate the RHS function $G(t, u)$ at the current solution u
- `TS.setRHSJacobian()`: Command to specify a routine to form some approximation of the Jacobian matrix at the current solution u
- `TS.setSolution()`: Command to select the initial guess for the solving process
- `TS.setFromOptions()`: Enable to set default values for solver parameters. Some parameters can be altered specifically such as tolerances can be changed by using `TS.setTolerances()`.
- `TS.setType()`: Set preferred time stepping scheme. Also it is generally used along with `TS.getSNES()`, `SNES.getKSP()` and `KSP.getPC()` to define internal nonlinear and linear solver routines.

- `TS.setPostStep()` and `TS.setPreStep()`: Command to set complex routines that are processed after and before a time step respectively. For example, calculation of custom decided convergence parameters can be done using a post step routine.
- `TS.setTimeStep()`: Command to set the initial time step for time integration process. End goal for the solver is set either by choosing the maximum time (`TS.setMaxTime()`) or steps (`TS.setMaxSteps()`).
- `TS.solve(u)`: Set solution vector u, and start the solution process. Using `TS.setMonitor()`, the solving process and converged reason can be actively monitored.

2.5. SOLVER IMPLEMENTATION

To achieve accurate emission prediction, it is necessary to have a highly coupled solver that solves transport and reaction terms together. Although it is computationally heavy, it provides more stability and reduces inherent errors associated with separate solving of transport and reaction terms. Based on research done by Cuoci et al. (2007) [11] and Stagni et al. (2013) [5], this complex coupled strategy can be reduced to a sparse system of N_E non-linear equations, as in equation 2.65. Here, ω is an 1D array of species mass fractions in all reactors with matrix size of N_E X 1, N_E is the total number of equations ($N_E = N_R * N_S$), N_R is the number of reactors and N_S is the number of species reacting inside each reactor. Due to this simplified structure, a resolution routine is designed by combining several numerical methods to optimize the solving process.

This non-linear equation system in Equation 2.65 is most commonly solved using Newton's method, transforming the non-linear system into a linear and iterative problem as in Equation 2.66. This method provides the fastest convergence but is highly prone to divergence. Suppose only this method is used to solve the system. In that case, it will definitely fail to converge because the initial guess, which is based on CFD simulation, is not anywhere close to the actual solution. However, this can be fixed with number of ways. First, use of a globalised Newton's method with path search algorithm like line search or trust region search can work well with any arbitrary initial guess. The solution process is done in two stages: Damped phase and Pure phase. Damped phase is performed from start till entering a good neighbourhood near the final solution, followed by pure phase for the remaining solution process.

Secondly, a precursor numerical method is required to push the reactor network to a state where the species concentration is close to the final solution. This can be achieved in multiple ways: (1) Sequential resolution (or Local Solve) (2) Global resolution through time stepping.

$$g(\omega) = -C\omega + f + R(\omega) = 0 \tag{2.65}$$

$$-J\Delta\omega = g(\omega) \tag{2.66}$$

For Sequential resolution, the reactors are individually resolved in time in sequential order. The reactor network is then updated based on the reactor states after every reactor is advanced by one time-step. This process is then repeated till local convergence, or steady-state is achieved. In Equation 2.65, three terms in the right-hand side are convection, external inlets and reaction terms respectively. Convection term is highly important for this solver as this is highly dependent on the adjacent reactors and the cell itself. On the other hand, reaction term is only dependent on individual reactors as reaction is a local phenomenon. Hence, for local resolution of each reactor, a modified equation can be derived by fixing the inlet transport contributions, which are evaluated through reactors states from previous iteration, as shown below.

$$[-C_{in}\omega + f]_{old} - C_{out}\omega + R(\omega) = 0 \tag{2.67}$$

Once the local solver has achieved convergence, the solution process is then governed by the global resolution method.

Global resolution using time integration is carried out using a modified false-transient version of Equation 2.65, as shown below in Equation 2.68. Using a suitable time-stepping scheme such as the BDF method, this ODE system is transformed into a non-linear system, which is further linearized as shown in Equation 2.69 and 2.70. Here, m_k is the the total mass of the reactor k, C_{ki} is single row of size 1 X N_E of convection matrix C corresponding to reactor k and species i, ω_{ki} is the mass fraction of species i in reactor k, Δt is the time-step size and J^n is the analytical Jacobian for the n^{th} time-step. After successful time integration, if the obtained mass fractions are close enough to the final solution, the solving process is handed to the global Newton's method. Also, the Jacobian for both time integration and Newton's solve are identical except the added diagonal terms in case of the former, as shown in Equation 2.70.

$$m_k \frac{d\omega_{ki}}{dt} = g(\omega_{ki}) = -C_{ki}\omega + f_{ki} + \Omega_{ki} V_k \tag{2.68}$$

$$m_k \frac{\omega_{ki}^{n+1} - \omega_{ki}^{n}}{\Delta t} = g(\omega_{ki}^{n+1}) \tag{2.69}$$

$$(\frac{m_k}{\Delta t} I - J^n)\Delta\omega_k = g(\omega_k^n) \tag{2.70}$$

2.5.1. AGNES V1.1 SOLVER IMPLEMENTATION

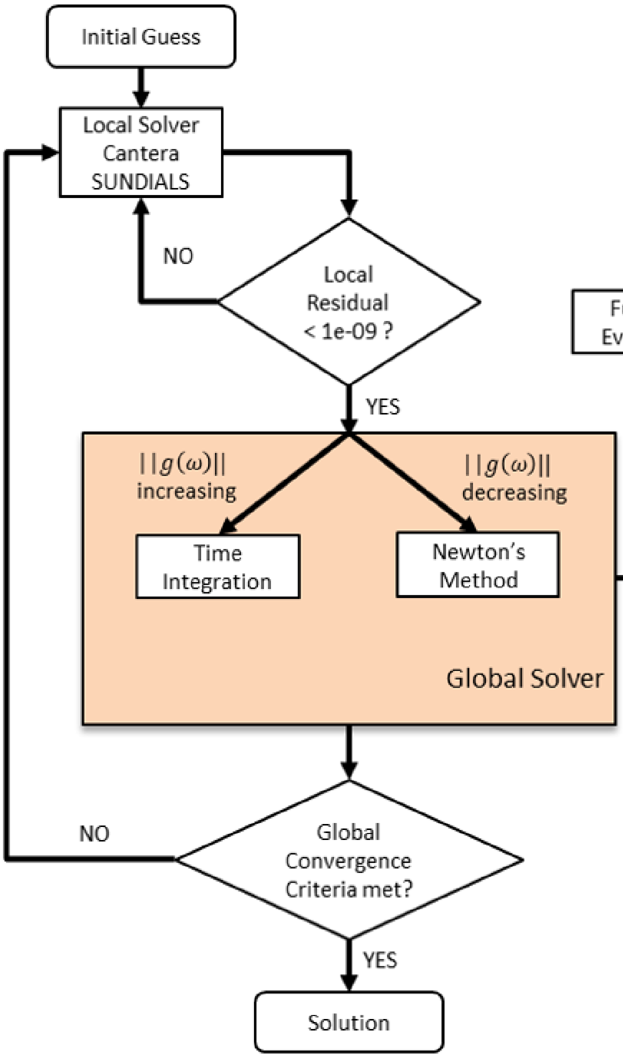

Figure 2.8: Schematic to visualise AGNES v1.1 resolution methodology

In the pre-research AGNES v1.1 version, the CRN solution process consists of three resolution methods: (1) sequential resolution using local solver, (2) global resolution using time integration and (3) global Newton's method. Each resolution method has its own specific purpose and all of them are integrated together quite well. The solution process begins with the Cantera based local solver. Each reactor in the chemical reactor network is advanced in time using Cantera numerical solver, SUNDIALS based on BDF method. This means one-by-one all reactors are integrated in time sequentially with different time steps based on their respective residence time. Cantera solves in built mass, species and energy equations rather than hard coded governing equations. After each reactor is advanced by a single time step, the entire CRN is updated and this process sequential time integration is repeated till local convergence is achieved. For local convergence, maximum relative change in species mass fraction in each reactor (ϵ_l) for the current and past 3 iterations drop below the threshold of 1e-09. It is defined as $\epsilon_l = max(\frac{\omega_k - \omega_{k,past}}{\omega_{k,past}})$, where ω_k is the species mass fraction in reactor k.

After achieving local convergence, solution process is passed on to the global resolution methods based on the behaviour of the residual. The residual norm $||g(\omega)||$ is the net production rate of all species in the CRN and when divided by the total number of equations presents the average production rate of each species. If residual norm is decreasing, then it is valid to state that the system is moving towards convergence and its diverging if the parameter starts increasing. Wit [17] used this behaviour to switch between fast converging Newton's method and highly stable but slow BDF based time integration. When the system is converging, Newton's method governs the solving process by making large advancements and when the system is diverging, time integration is used to take small time steps and variable increments. Together, the global solver remains stable and this residual behaviour acts as control system. Furthermore, Python SciPy package is selected for accessing user friendly BDF time integration and Newton's method routines. Both methods require the exact same input of residual function and its Jacobian.

For global convergence is achieved after satisfying two residual conditions. The first residual is denoted by ϵ, which is defined as the maximum change in species mass fraction divided by maximum species mass fraction from previous iteration. The second residual ϵ_{rate} is the maximum rate of production of a species among all other species present in the CRN. Furthermore, maximum changes in minor species mass fraction like NO and CO are monitored to ensure correct solution.

$$\epsilon = max(\frac{\omega - \omega_{old}}{max(\omega_{old})}) \tag{2.71}$$

$$\epsilon_{rate} = max(g(\omega)) \tag{2.72}$$

Theoretically, at steady state, there should not be any change in species mass fraction and net production rate of all species should be zero i.e $\epsilon = 0$ and $\epsilon_{rate} = 0$. Therefore, keeping computational cost in mind, the final solution is achieved when both $\epsilon < 1e-09$ and $\epsilon_{rate} < 1e-15$. These values provide acceptable accuracy at a reasonable computational cost.

Advantages: In the current state, this solver is generating accurate emission prediction results. This solver is validated based on the research done by Sampat [6] and Wit [17]. The solution process has good stability and convergence but more testing with complex CFD simulation cases is required. Python SciPy based time integration and Newton's method provide the user good accuracy and accessibility without any need of advanced Python programming needs.

Disadvantages: As mentioned earlier, this implementation relies on a local solver. It solves every reactor sequentially till convergence, this become computational expensive with a high reactor count. Secondly, the global time integration is only active for less than 10% of the total solving time, hence its converging capabilities are not utilized to its fullest extent. In order to improve tool's efficiency, a better alternative approach is required that relies mostly on global resolution via time integration.

2.5.2. AGNES V1.2.1: KPPSMOKE APPROACH

The first improved approach over AGNES v1.1 is based Stagni et al. [5] KPPSMOKE software. Similar to AGNES v1.1, this approach also uses three solution methods: local solver for sequential resolution, global time integration and globalised Newton's method with line search. The place where it differs from pre-research AGNES is its assigned high importance on global time integration and how all these resolution methods interact and handle the entire solution process. The solution process begins with the Cantera based local solver. Each reactor in the chemical reactor network is advanced in time by one time step, the entire network is updated and this process is repeated. The effectiveness of this sequential method is evaluated by investigating the behaviour of the residual function norm. The ratio between the current residual value to the previous iteration is calculated and if average of this ratio over latest 5 iterations is more than 0.95, local solver is stopped and control is passed onto the global time integration. This average residual ratio shows that there in no significant reduction in residuals using local solver.

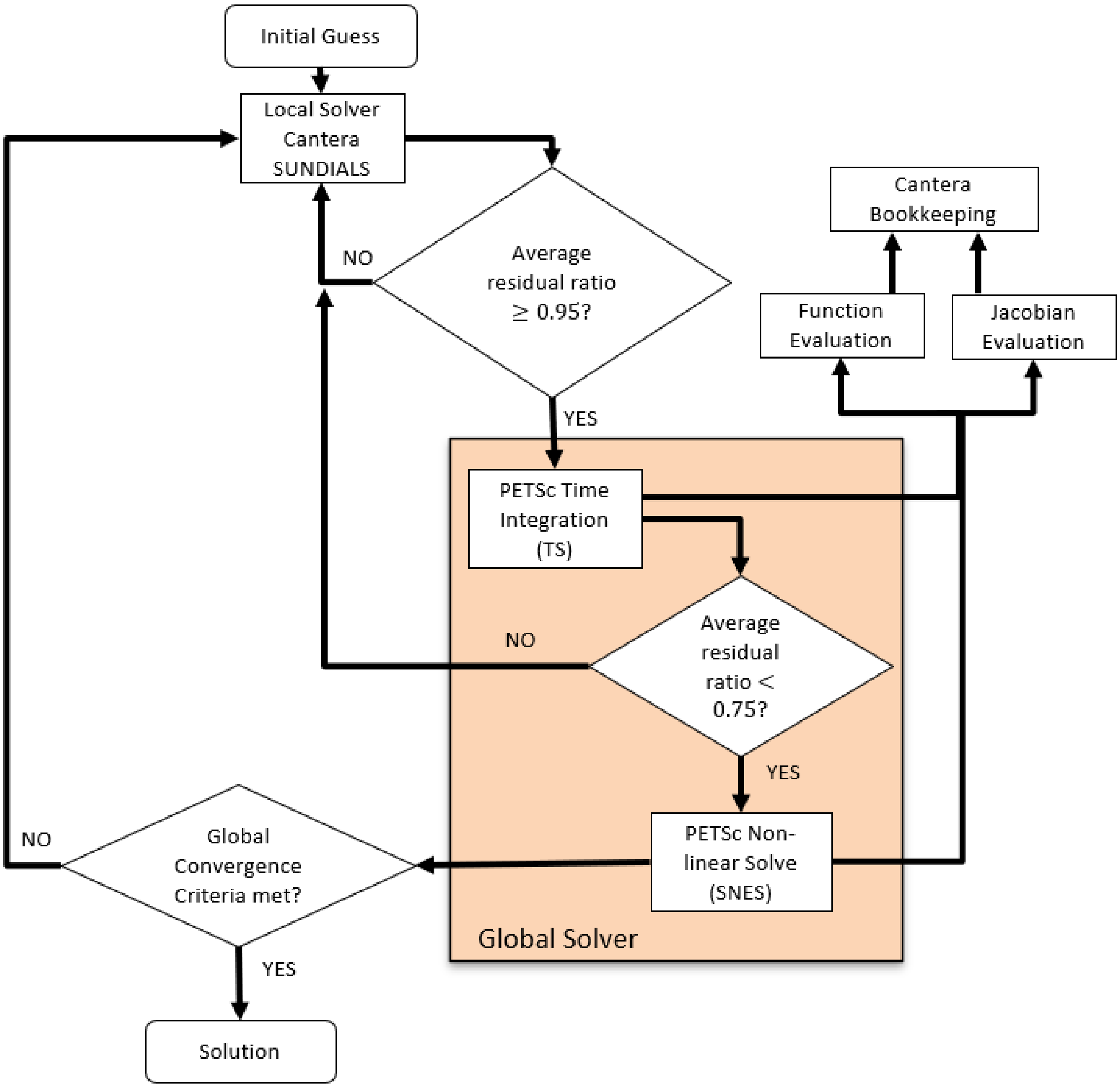

Figure 2.9: Schematic to visualise AGNES v1.2.1 resolution methodology

When dealing with complex flows with strong recirculations and species mixing, for accurate modelling a fine resolution with large number of reactors is required. In these case, local solver or sequential resolution approach fails to reduce the residuals significantly for all equations so that Global Newton's method could be applied. Hence, after reaching a saturation point in residual reduction, global resolution is attempted by solving the global ODE. Time integration is done using PETSc TS module that offers user flexibility in choosing preferred nonlinear and linear solvers. Instead of dealing with SciPy BDF solver as black box, all parameters can now be changed based on performance. BDF time stepping scheme is used along with restarted GMRES scheme (restarted every 40 iterations) as the linear solver and ILU preconditioner. Generally, the time stepping method is robust but computationally expensive if compared with Newton's method, which offers faster convergence but has undesirable convergence properties. therefore, after few time steps, AGNES v1.2.1 attempts to solve the steady state problem using PETSc based Globalised Newton's method. If the steady state solution fails, the solver returns to false transient time stepping procedure. If the average residual ratio drops below 0.75, which means the solver is approaching the solution faster, time integration is discontinued and Newton's method using line search method is attempted. This process is repeated till convergence criteria is satisfied. For convergence, the necessary condition is residuals for all governing equations (ϵ_{rate}) are below 1e-17. Also, maximum relative change in species mass fraction (ϵ) for minor species like NO and CO are actively monitored and attempted to reduce below 1e-09.

Advantages: The biggest advantage of this approach is that its is already validated by several researchers [4][5] and this is already available as an open source software for all. The results are accurate and at a reasonable computational cost. It is superior than AGNES v1.1 because of all three resolution methods have a similar importance. Unlike pre-research AGNES, global time integration is actually contributing to around 30 % of the solving process

Disadvantages: The problems associated with local solver is still lingering in this approach. Solution process becomes heavily reliant on global resolution methods when number of reactors are increased. This means, local solver can be entirely removed without much loss in performance.

2.5.3. AGNES V1.2.2: ONLY TIME INTEGRATION (TS)

Based on the shortcomings of sequential resolution or local solver, need for a robust global time integration approach that is able to solve the stiff system of ODEs is highly desirable. This brings us to AGNES v1.2.2 implementation which is a purely global time integration based approach. For this approach, focus is on the solution accuracy rather than on solution speed.

As mentioned above, this approach only involves a global time integration method that resolves the entire CRN in time by solving system of ODEs, shown in Equation 2.68. This involved the use of PETSc TS class where specific routines are provided for evaluating the RHS function $g(\omega)$ and its associated Jacobian. One important difference for this resolution method is the the time dependent Jacobian term, shown in Equation 2.70, which is simply an added diagonal term with the time independent Jacobian, used in the Global Newton's method. This produces a better conditioned system and the solution from the n^{th} time step acts as an excellent initial guess for the $n+1^{th}$ time step.

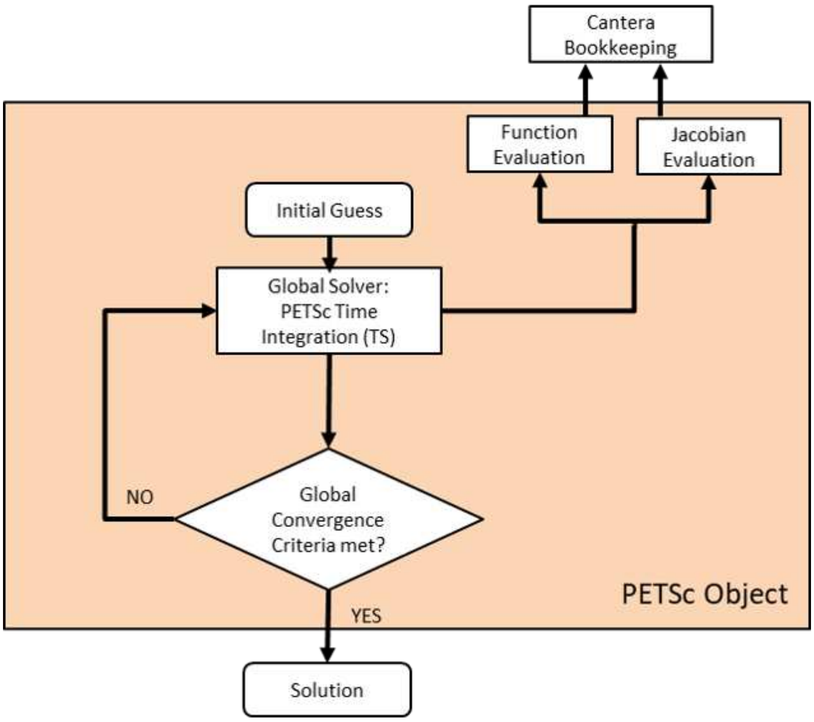

Figure 2.10: Schematic to visualise AGNES v1.2.2 resolution methodology

Additionally, the time step size is not kept constant during the time integration approach. As the species conservation equation is solved for all species, time step size reduction is used as a way to fix inconsistencies like total sum of mass fraction not equal to 1 in each reactor, negative mass fractions and even mass fractions more than 1, and otherwise time step size is gradually increased. This along with PETSc adaptive time stepping, the time step sizes are varied throughout the solving process. The choice of atol and rtol play an significant role on the jumps in the time stepsize and accuracy in each time step. The solving process for each time step is relatively inexpensive and it employs the same nonlinear and linear solvers as the Global Newton's method. generally, it will always be slower than the Newton's method but this approach offers good stability and desirable convergence properties. For checking convergence, same parameters (ϵ, ϵ_{rate}) as before are used. But as we increase atol and rtol, satisfying these global convergence criteria becomes increasing difficult. Three combination of atol and rtol are tested: Accurate (atol=1e-11,rtol=1e-09), moderate (atol=1e-08,rtol=1e-06) and inaccurate (atol=1e-05,rtol=1e-03).

Advantages: The major advantage of this approach is the complete omission of local solver. There is no added cost with high number of reactors, Also, this application has quite simple and reduced complexity.

Disadvantages: This approach is capable of achieving accurate results but at high computational cost. With small values of atol and rtol, time step sizes remain quite low for majority of the solving process which makes time advancement quite tedious. Hence, this can only used as precursor to the faster Newton's method. This approach is discussed in the next subsection.

2.5.4. AGNES v1.2.3: TIME INTEGRATION AND GLOBALISED NEWTON'S METHOD

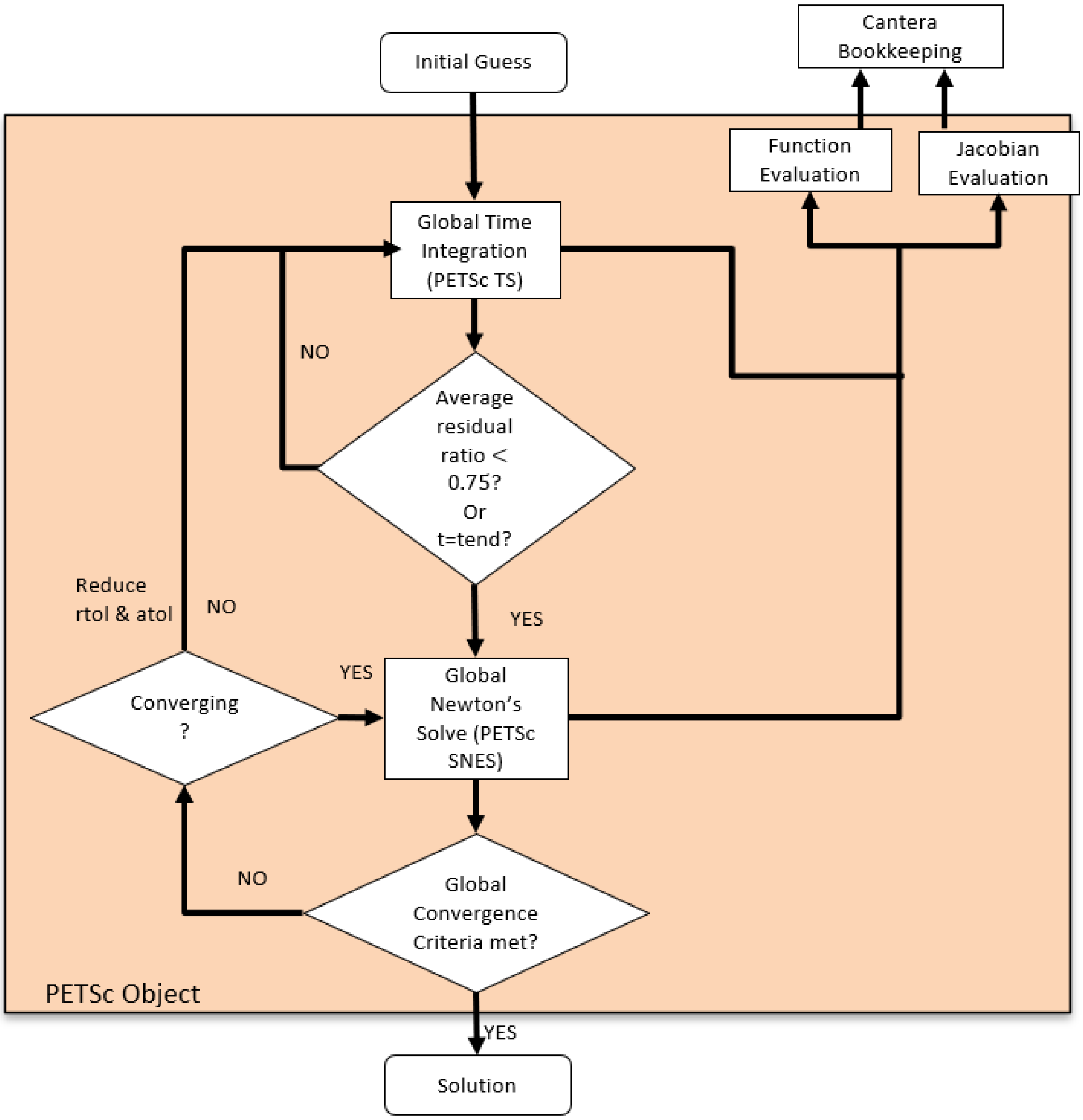

Figure 2.11: Schematic to visualise AGNES v1.2.3 resolution methodology

This final solver implementation is a pure global resolution approach that utilizes an inaccurate time integration method to simply reduce the residuals to an acceptable level after which globalised Newton's method is used for faster convergence. This combination of two global resolution method offers good solution time for cases with high number of reactors. This is due to the absence of any sequential resolution. Here, the global time integration has exact same functionality as a sequential approach. This method only focuses on reducing the residuals related to all governing equations to form a better initial guess for the next step. Obviously, the better the starting guess for the Newton's method, the less likely AGNES will have to resort time integration. Bulk of the solving process is handled by Newton's line search method.

Both methods are designed based on PETSc modules and act exactly similar to the previous discussed solvers. Both time integration and Newton's method require specific routines to evaluate residual function (RHS function in case of TS) and its associated Jacobian. The calculation part of function vector and Jacobian matrix remain exactly the same for both methods. For the first few iterations, only time dependent system of ODEs are solved. If the solver is approaching the solution faster, as observed based on average residual ratio dropping below 0.75 [5], steady state equations are solved using Newton's line search method. In case Newton's method fails, the solver goes back to time integration but now with stricter atol and rtol for few time steps. This process is repeated till convergence is achieved. As all previously discussed solver implementations are compared, the convergence criteria is kept similar.

Advantages: The approach performs quite well compared to both AGNES v1.2.1 and AGNES v1.1 when the number of reactors are high. This validates our doubt on local solver's capability to resolve CRN system with fine resolution. It struggles for low reactor count cases but it can be further improved by making time integration more adapt in changing time step sizes. As most of the convergence work is done using Newton's method, the solution process is fast. This combined with robustness and stability of time integration makes a suitable solver when dealing with complex flow features like recirculations.

Disadvantages: As mentioned before, an inaccurate global resolution via time integration is used in place of the traditionally used local solver and global time integration combination. However, further case studies are required to definitely answer till what limits atol and rtol can be decreased such that solver fails entirely. This will allow to improve the performance of the solver even further.

3

CFD TEST CASE: SANDIA FLAME D

AGNES performance depends highly on CFD simulation data as an input to make species concentration and temperature predictions. However, there are certain restrictions on the choice of test case. First, the test setup/geometry cannot be too complex. This is because AGNES, in its current state, can only handle one Fluent fluid zone. Additionally, complex geometry increase CPU consumption thus reducing the overall number of simulations. Second, only gaseous fuel can be used as currently AGNES does not allow the use of liquid fuels and the liquid fuel breakdown mechanism's complexity would also increase the CPU time. Third, the test case should include species concentration and temperature data which can be used to validate AGNES predictions. Based on the restrictions, Sandia Flames are the best choice.

3.1. SANDIA FLAMES

Sandia Flames are piloted methane/air jet flames and are labelled from C to F [8]. Flame C to F are turbulent flames with increasing Reynolds' number. The flow velocity for both main and pilot flame increase over the range of these flames thus, increasing the probability of local extinction.

The measurement data for mainly flames C to F include temperature, mass fraction for both major and minor species such as N_2, O_2, H_2O, H_2, CH_4, CO, CO_2, OH and NO [8]. CO is measured using both Raman scattering and Laser Induced Fluorescence (LIF) but CO-LIF measurements are more accurate and should be used for comparison. These measurements include axial and radial profiles of temperature and Reynolds and Favre averaged species mass fraction and its fluctuations.

The main jet is a partially-premixed mixture of CH_4 and air in 1:3 proportion. The partial premixed nature results in a shorter but more robust flame than pure methane or nitrogen-diluted methane. Due to high mixing rate,the main flame burn as non-premixed flame with a single reaction zone near stochiometric mixture fraction and no premixed reaction zone in fuel rich mixtures. The pilot flame consists of a premixed lean mixture (ϕ=0.77) of C_2H_2, H_2, air, CO_2 and N_2 with similar enthalphy and equilibrium composition of main jet composition of CH_4/air. The pilot flame flow-rate is altered for each flame such that the pilot energy release is approximately 6% of the main jet flame. Figure 3.1 and 3.2 shows Sandia Flame D and flame holder dimensions.

Figure 3.1: Close up of Sandia flame [2]

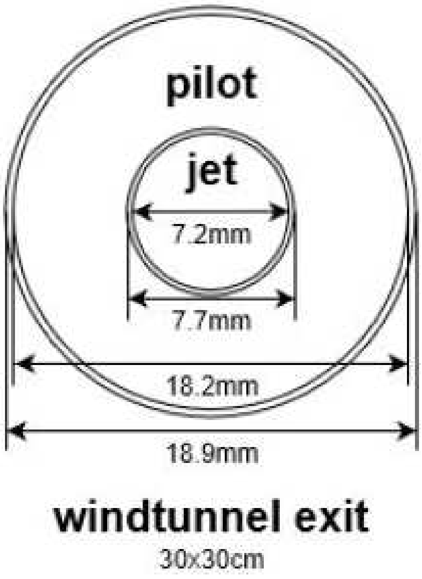

Figure 3.2: Dimensions of Sandia flame test setup [2]

For this research project, Sandia Flame D [2] is selected due to its high Reynolds number (Re=22400) which makes it an ideal choice for turbulent non-premixed combustion model validation with a small probability of local extinction. Additionally, emission species measurement data from experiments is available for flame D. Prior research done by Monaghan et al. (2012,2014) [33] [34] shares the application of Sandia Flame D for research into pollutant formation and emission prediction using CFD-CRN method. In 2012, Monaghan et al. (2012) [33] analyzed Sandia Flame D to study NO_X formation pathways using CRN. Monaghan et al. used temperature, mixture fraction and axial position to split the diffusion flame into multiple zones. Each zone was later clustered and modeled as a PSR. Furthermore, in research done by M.D. Wit (2019) [17], Sandia Flame D is used as a CFD test case for automatic CRN construction and validation of AGNES emission predictions and the effect of solving energy equation of the species concentration.

3.2. EXPERIMENTAL RESULTS

Measurements of flame properties and concentration of species like N_2, O_2, CH_4, CO_2, H_2O, H_2, CO, OH, and NO for Sandia flame D were obtained as axial profiles at locations x/d = 5, 10, 15,...,75 and as radial profiles at x/d = 1, 2, 3, 7.5, 15, 30, 45, 60, and 75 for radial profiles. Figure 3.3 depicts the axial profiles of mixture fraction and temperature for piloted flame C to F.

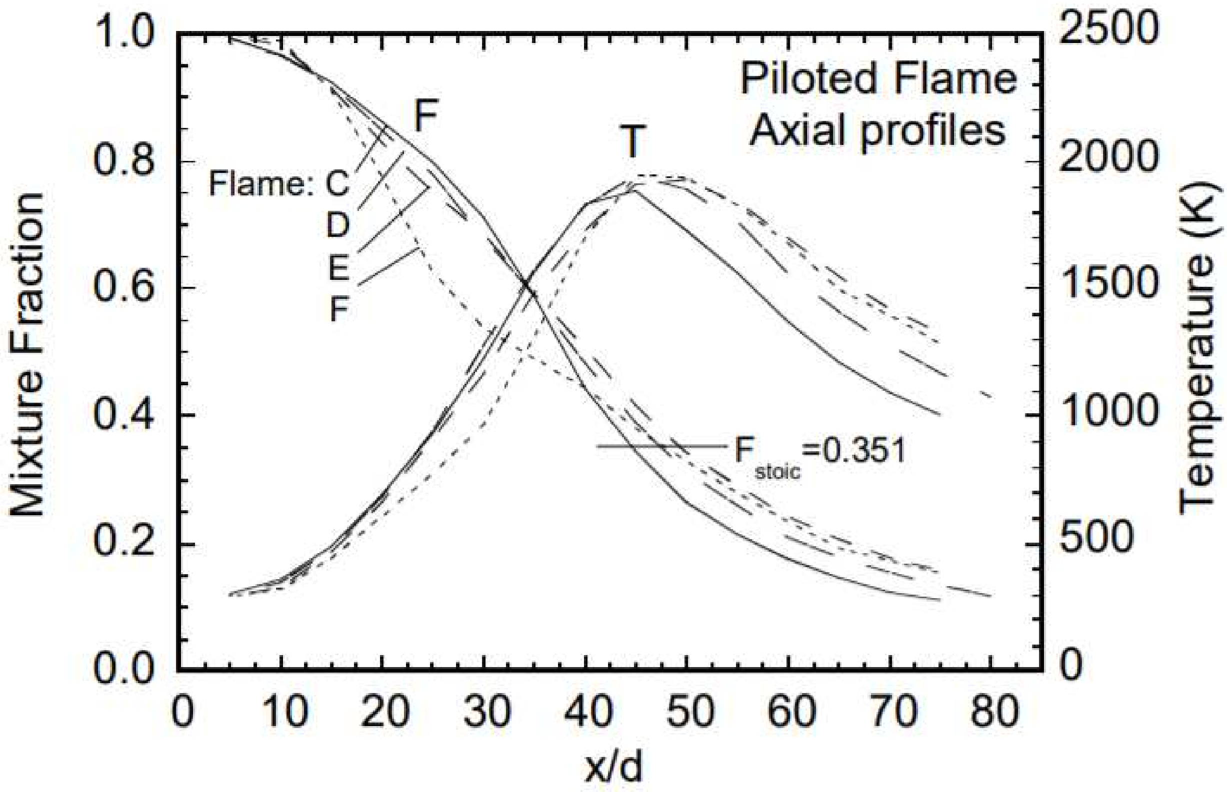

Figure 3.3: Axial profiles of measured mixture fraction and temperature (Favre averaged) in piloted flames C, D, E, and F [8]

Table 3.1 provides the axial locations and their respective radial ranges for which Sandia flame D experimental results are provided as radial profiles. Figure 3.4 presents a visual representation of theses axial locations with respect to the flame.

x/d	1	2	3	15	30	45	60	75
r/d	1.9	1.9	2.18	3.06	5.83	7.78	9.72	11.11

Table 3.1: Axial locations and their respective radial range of Sandia flame D experimental results (d=0.0072 m)

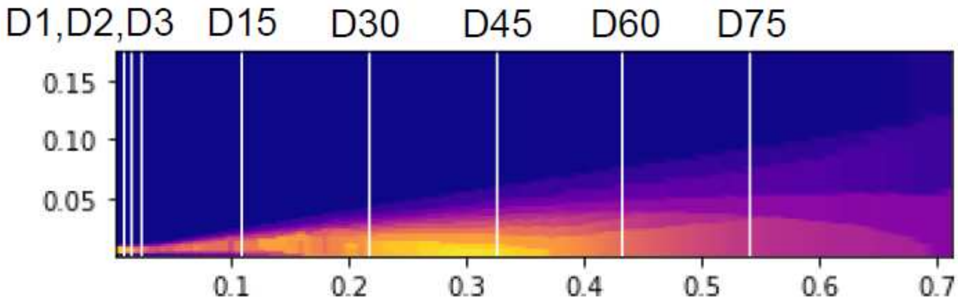

Figure 3.4: Visualisation of all axial locations where temperature and species concentration are measured

3.3. CFD Simulation

CFD simulation data is an integral part of CRN generation and computation. Without a good starting point, the accurate construction of reactor network will suffer significantly. As discussed in section 3.1, Sandia Flame D is selected as the CFD test case. Existing CFD results are used for this project. These simulations were performed by A.A.V Perpignan, PhD and were later used by Wit de Wit, MSc for AGNES CRN construction and validation [17].

The CFD simulations were done using ANSYS Fluent with a similar flame derived from prior research done by Habibi et al.[35] and Merci et al.[36]. The CFD grid consisted of 22278 2D cells with cell spacing increasing with distance from origin. The CFD simulation used Reynolds Stress turbulence model along with the Discrete Ordinates model for radiation. Turbulence-chemistry interaction was modelled using Flamelet Generated Manifolds (FGM) approach where flamelets were generated using mixture fraction, progress variable and enthalpy based on work done by van Oijen and de Goey [37].

3.4. CRN SETUP

Sandia flame D is an axisymmetric turbulent flame case with no re-circulation. Turbulent diffusion is the phenomenon of heat and mass transport within a system due to chaotic, random time-dependent fluctuations. Also, turbulent diffusion occurs more rapidly compared to molecular diffusion. This becomes increasingly important to model in test cases that involve rapid mixing of species. Therefore, modelling turbulent diffusion in AGNES becomes a critical addition to ensure correct species mixing and emission predictions. Wit de Wit, MSc [17] added turbulent diffusion to AGNES using the Peclet number. Peclet number is a non-dimensional number that describes the ratio of advective rate of a physical quantity, such as mass flow rate, to its diffusive rate. Peclet number for turbulent mass transfer ($Pe_{m,t}$) is defined as the ratio of species' advective mass flow rate ($\dot{m}_{adv}$) to its turbulent diffusive mass flow rate ($\dot{m}_{diff,t}$). This can further simplified in terms of velocity (u), characteristic cell length (l_{cell}), which is approximated as cube root of cell volume and turbulent mass diffusivity ($D_{m,t}$)

$$Pe_{m,t} = \frac{\dot{m}_{adv}}{\dot{m}_{diff,t}} = \frac{u l_{cell}}{D_{m,t}} \tag{3.1}$$

Turbulent mass diffusivity is further dependent on turbulent Schmidt number (Sc_t), density (ρ) and turbulent viscosity (μ_t). Turbulent Schmidt number is another non-dimensional number that relates turbulent transport of momentum and mass/heat. This is assumed to constant at 0.7 [9]. Therefore, turbulent diffusive species mass flow rate is then calculated as

$$D_{m,t} = \frac{\mu_t}{\rho Sc_t} \tag{3.2}$$

$$\dot{m}_{diff,t} = \frac{\dot{m}_{adv}}{\rho u}\frac{\mu_t}{l_{cell} Sc_t} = \frac{A\mu_t}{l_{cell} Sc_t} \tag{3.3}$$

Here, A is the cross-sectional are of the cell face.

Pre-research version of AGNES [6] used a common clustering criteria and tolerance. This tolerance determines the range within which similar cells can be clustered together into reactors. The tolerance and clustering criteria remains the same for the entire domain. However, this has a glaring disadvantage for cases that have some less interesting zones. For example, in case of Sandia flame D, several outward radial locations are insignificant compared to locations close to the symmetric axis. Therefore, Wit [17] split the computational domain into zones with each their own tolerances according to the method by Monaghan et al. [9], which uses the static temperature (T), mean mixture fraction (f) and axial-coordinate (x) are selected as clustering criteria. Table 3.2 presents information regarding the zoning limits and tolerance as specified by Monaghan et al [9].

Zone limits	**Reactor criteria limits**
$0 \leq f \leq 0.01$	$\Delta x = 0.2$ m
$0.01 < f \leq 0.1$	$\Delta T = 100$ K
$0.1 < f \leq 0.9$, $T \leq 1800$ K	$\Delta T = 100$ K
$0.1 < f \leq 0.9$, $1800 < T \leq 2000$ K	$\Delta T = 50$ K
$0.1 < f \leq 0.9$, $T > 2000$ K	$\Delta T = 2$ K
$0.9 < f \leq 1.0$	$\Delta f = 0.01$, $\Delta x = 0.01$ m

Table 3.2: Zone limits and reactor criteria as specified by Monaghan et al. [9]

Due to Monaghan et al. [9] clustering approach, better radial profiles and average quantities are achieved and good matching with the experimental results. Local tolerances can be altered in order to group more important cells into reactors thus allows for more adaptability. Also, due to zoning, less important locations can be removed from computation, thus saving in computational cost. Therefore, this same clustering approach is adapted for this project due to its proven effectiveness in modeling Sandia flame D and accurate species concentration predictions [9][17].

4

RESULTS AND DISCUSSION

Using the governing equations and PETSc solver implementations described in Chapter 2, AGNES emission prediction results are obtained. First, Section 4.1 presents the PETSc solving capabilities in solving a single reactor system using GRI-Mech 3.0 reaction mechanism. This step is important to decode the several functionalities available within PETSc and its Python wrapper petsc4py library. Increased complexity with high reactor count is first tested using a three reactor systems and its results are shown in Section 4.2. After finalizing all necessary choices for PETSc implementation, such as preconditioner, linear and non-linear schemes, etc., AGNES v1.2 is tested using Sandia flame D CFD test case with the original Monaghan et al. [9] zoned clustering. Section 4.3 consists of reactor independence test to arrive at a reactor count that offers both good accuracy and solving speed, results comparing AGNES v1.2 with pre-research version AGNES v1.1 [**?**], experimental data and CFD results. Finally this section is concluded with a discussion on the designed solver performance based on solving time and convergence rate. Section 4.3.4 shares results that are important to resolve the research question posed in this thesis.

4.1. SINGLE REACTOR

Let consider a single reactor system at 1500 K temperature and 1 atm pressure. The reactant mixture is methane and air at equivalence ratio ϕ. The global reaction is shown below in Equation 4.1. Here, v_i are the stochiometric coefficients for reactants and all possible major products. Estimating these stochiometric coefficients are important to calculate product mass fractions, assuming no other products are formed due to this reaction.

$$\nu_1\,\mathrm{CH_4} + \nu_2(\mathrm{O_2} + 3.76\,\mathrm{N_2}) \longrightarrow \nu_3\,\mathrm{CO_2} + \nu_4\,\mathrm{H_2O} + \nu_5\,\mathrm{N_2} + \nu_6\,\mathrm{O_2} + \nu_7\,\mathrm{CH_4} \qquad (4.1)$$

For $\phi \geq 1$, CH_4 is in excess, hence it is correct to assume $\nu_6 = 0$. With this the global balanced equation is derived:

$$\mathrm{CH_4} + \frac{2}{\phi}(\mathrm{O_2} + 3.76\,\mathrm{N_2}) \longrightarrow \frac{1}{\phi}\mathrm{CO_2} + \frac{2}{\phi}\mathrm{H_2O} + \frac{7.52}{\phi}\mathrm{N_2} + 0\,\mathrm{O_2} + (1\text{-}\frac{1}{\phi})\,\mathrm{CH_4} \tag{4.2}$$

For $\phi < 1$, O_2 is in excess, hence it is correct to assume $\nu_7 = 0$. With this the global balanced equation is again derived:

$$\mathrm{CH_4} + \frac{2}{\phi}(\mathrm{O_2} + 3.76\,\mathrm{N_2}) \longrightarrow \mathrm{CO_2} + 2\,\mathrm{H_2O} + \frac{7.52}{\phi}\mathrm{N_2} + 2\,(\frac{1}{\phi} - 1)\,\mathrm{O_2} + 0\,\mathrm{CH_4} \tag{4.3}$$

Based on the above balanced global reaction, mass fractions for all possible major products are calculated and presented in Table 4.1, 4.2 and 4.3 based on ϕ = 1, 1.5 and 0.5 respectively. The governing equation (Equation 2.11) is then solving using Cantera, time integration and Newton's method using PETSc library. Cantera [23] uses a numerical solver, called SUNDIALS [24], for ODE integration with time, by using 'advance()' function after setting appropriate time-step size (dt) and tolerances like rtol and atol. PETSc based time integration and Newton's method are set using PETSc TS and SNES class respectively, discussed in detail in Section 2.4.6 and 2.4.5. Motivation for this comparison was to gain basic understanding of PETSc functionalities and arrive at a working PETSc TS and SNES layout for future steps. Also, only species mass conservation equation is considered and energy equation is not solved.

Parameters	Hand calculations	Cantera	Time integration	Newton's method
Y_{CH_4}	0	9.31E-17	5.47E-17	5.23E-17
Y_{O_2}	0	6.29E-08	4.72E-08	4.72E-08
Y_{CO_2}	0.151	0.144	0.143	0.143
Y_{H_2O}	0.124	0.123	0.123	0.123
Y_{CO}	NA	6.09E-03	6.25E-03	6.25E-03
Y_{NO}	NA	9.28E-06	5.46E-06	5.46E-06

Table 4.1: Product species concentration for methane-air combustion at ϕ = 1.0 inside a single PSR

Parameters	Hand calculations	Cantera	Time integration	Newton's method
Y_{CH_4}	0.027	1.31E-03	1.25E-03	1.25E-03
Y_{O_2}	0	1.21E-11	7.54E-12	7.48E-12
Y_{CO_2}	0.147	0.0890	0.0879	0.0879
Y_{H_2O}	0.121	0.114	0.113	0.113
Y_{CO}	NA	0.0796	0.0813	0.0813
Y_{NO}	NA	2.58E-09	4.37E-10	4.39E-10

Table 4.2: Product species concentration for methane-air combustion at $\phi = 1.5$ inside a single PSR

Parameters	Hand calculations	Cantera	Time integration	Newton's method
Y_{CH_4}	0	3.64E-28	6.53E-30	8.61E-30
Y_{O_2}	0.113	0.112	0.112	0.112
Y_{CO_2}	0.078	0.0768	0.0768	0.0768
Y_{H_2O}	0.064	0.063	0.063	0.063
Y_{CO}	NA	7.62E-07	8.54e-07	8.54e-07
Y_{NO}	NA	3.53E-06	1.47E-06	1.47e-06

Table 4.3: Product species concentration for methane-air combustion at $\phi = 0.5$ inside a single PSR

Based on the results presented in the above tables, it is quite clear that PETSc based solver predict all relevant species with good accuracy, compared to Cantera. For this case, it is assumed that Cantera results are correct as major species concentration are close to hand-calculated concentrations. Slight difference is present for minor species mainly NO.

Figure 4.1 provide solving time comparison between these three solvers. All parameters like rtol, atol, initial time step size and even order of BDF method (default order=1) are kept same for Cantera and PETSc based solvers. Surprisingly, Cantera is the fastest when solving small reactor networks. Cantera uses SUNDIALS solver suite for solving species conservation equations in time where it performs dense numerical linear algebra. The 'advance' method is called till the desired end time is reached, and the solver takes as many internal, variable-sized, time steps as necessary to get to that end time similar to PETSc time integration. The only explanation for this fast computation is Cantera uses element potential method to arrive at equilibrium state. This approach reduces the system of N_S equations into K equations where N_S is the number of species and

K is the number of elements [38]. However, with increased complexity associated with high reactor count, Cantera shows slow solution rate. As the governing equation is a system of non-linear equations, it is natural that Newton's method using line search will converge faster than time integration of false transient stiff system of ODEs. Also, for future implementations a combination of time integration and Newton's method is also verified.

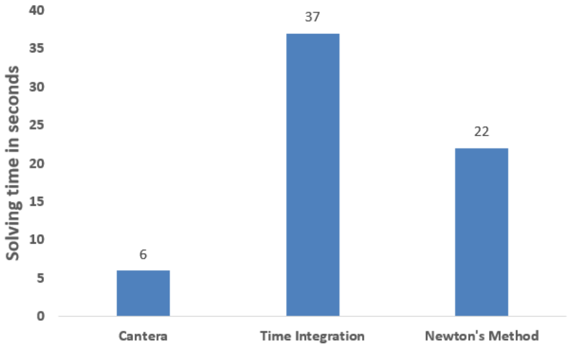

Figure 4.1: Comparison of solution time to resolve a single reactor system using three solvers

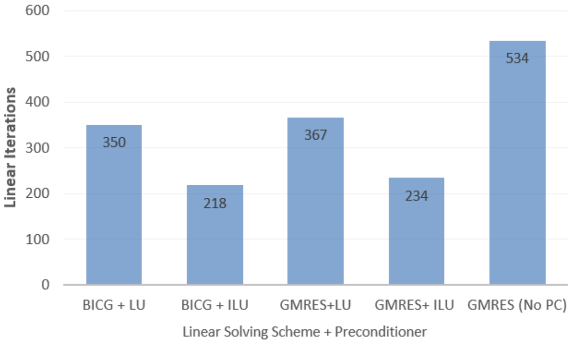

Figure 4.2: Number of linear solve iterations based on different choices of solving scheme and preconditioner for solving a single reactor system

Another important observation for this case study is the importance of choosing the correct linear solving scheme and its associated preconditioners. Detailed description of several important linear solving schemes and use of preconditioning is presented in Section 2.4.4. PETSc time integration and nonlinear solve is heavily reliant on the performance of internal linear solves, hence selecting the best possible combination is paramount before moving to more complex problems. From Figure 4.2, it is visible that with appropriate preconditioner (PC), solution is achieved with few number of linear solve iterations which in turn reduces solving time. With no preconditioner used, linear solve using GMRES method [7] requires 534 iterations which is reduced to 234 iterations with Incomplete LU (ILU) preconditioner [7]. This shows the importance of correct preconditioner.

4.2. MULTIPLE REACTORS

This section presents emission predictions for a three reactors system, as shown in Figure 4.3. Each reactor is operating at 1 atm pressure and contains methane and air mixture at $\phi = 1.0$. Temperature is kept different for all three reactors and there is 50-50 mass flow rate split from R1 to R2 and R3. GRI-Mech 3.0 reaction mechanism is used with 53 species. Figure 4.4 shows the analytical Jacobian structure for this reactor network. In the figure, the big blocks are the reaction contributions inside of each reactor. as reaction is local phenomenon, this part of the Jacobian is only dependent on the reactor itself. On contrary, transport contributions can be seen as four diagonal block matrices, showing connections between R1-R2 and R1-R3. This portion of the Jacobian is dependent on the adjacent reactors and it remains constant throughout the solving process as it is calculated using only CFD data.

Similar to Section 4.1, product species mass fraction are calculated using Cantera and PETSc based solver. Table 4.4 shows the collective concentration predictions inside all three reactors. Reactor R1 is maintained at room temperature, hence there is no ignition or any progress in the combustion reaction. However, this mixture is pumped into reactors R2 and R3 using a 50% mass flow split. R2 and R3 are maintained at 1500 K and 2000 K temperature respectively.

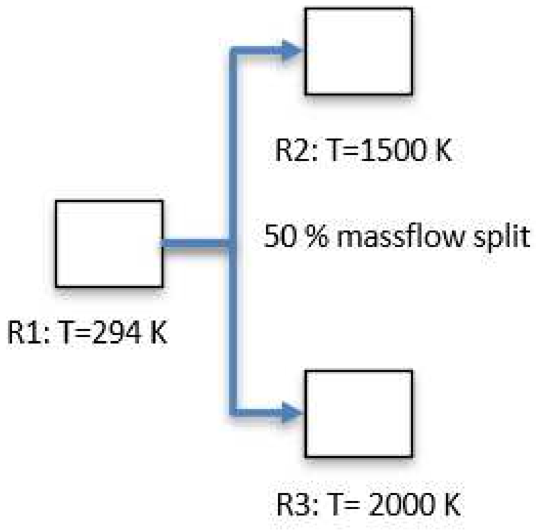

Figure 4.3: Three reactors system

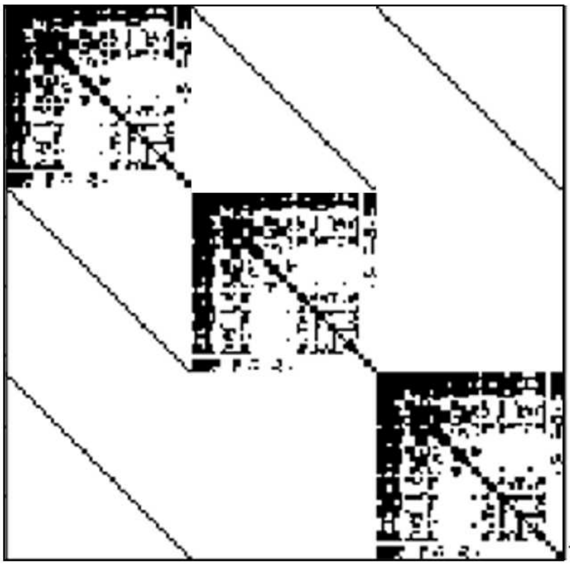

Figure 4.4: Analytical Jacobian for the three reactors system

As combustion is occurring at stochiometric conditions $\phi = 1.0$, almost all reactants are consumed. At T=2000 K, there is more NO production due to dominant thermal pathway. Also, at this high temperature, there is slightly high accumulation of CO molecules and less CO_2 production due to oxidation of CO.

$Y_{Species}$ / Solver	Cantera			PETSc		
	R1	R2	R3	R1	R2	R3
Y_{CH_4}	0.056	8.13E-15	5.42E-17	0.056	8.13E-15	5.42E-17
Y_{O_2}	0.217	6.37E-09	2.48E-04	0.217	6.37E-09	2.48E-04
Y_{CO_2}	0	0.144	0.141	0	0.143	0.140
Y_{H_2O}	0	0.123	0.123	0	0.123	0.123
Y_{CO}	0	6.09E-03	8.10E-03	0	6.25E-03	7.86E-03
Y_{NO}	0	9.28E-06	2.53E-04	0	5.46E-06	2.61E-04

Table 4.4: Product concentration for methane-air combustion at $\phi = 1.0$ inside a three PSR network

Figure 4.5 presents the difference in solving time. The only new observation is that combination of time integration and Newton's method proved to be working quite well. It has an acceptable solving time with added improvement in remedying any convergence issue regarding Newton's method. This approach is further tested using Sandia flame D test case with high reactors count.

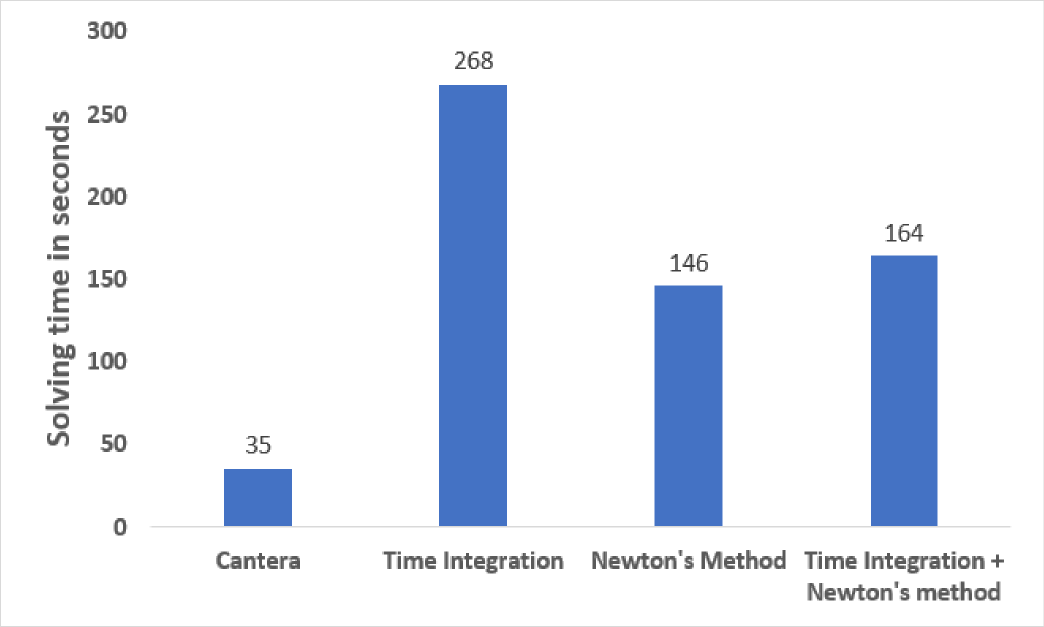

Figure 4.5: Comparison of solution time to resolve a three reactors system using three solvers

4.3. SANDIA FLAME D

This section contains the results of the CFD-CRN method for emission prediction based on Sandia Flame D CFD simulation. One of the main reasons for using the AGNES CFD-CRN method is accurate concentration predictions, mainly for minor species, which is hard to recreate using only CFD simulation. Therefore, the results are compared with SFD experimental results for the test case and CFD results, which acted as the starting point in constructing the CRN. Furthermore, two versions of AGNES are compared in order to validate the newly designed solver implementation with the pre-research version. This is done to ensure that the accuracy is not compromised by using a different algorithm for CRN solution. This section is concluded with a comparative analysis of the solver performance using parameters like solving time and convergence rate. Figure 4.6 and 4.7 shows the sparsity pattern in the transport and reaction Jacobian based on reactor network, generated using CFD simulation.

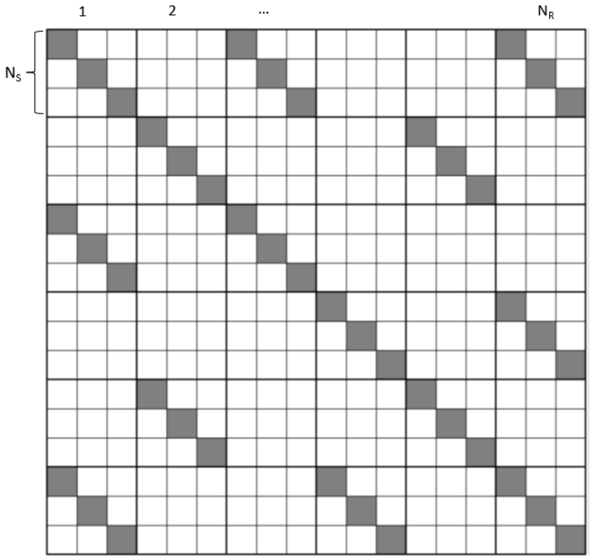

Figure 4.6: Transport Jacobian (J_S) sample structure accounting for reactor inter-connectivity, based on Sandia flame D CFD simulation and energy (temperature recalculation) OFF

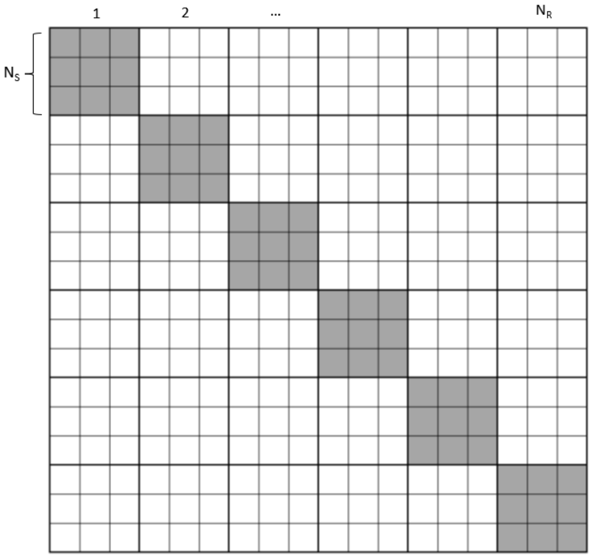

Figure 4.7: Reaction Jacobian (J_R) sample structure accounting for all reactions inside each reactor, based on Sandia flame D CFD simulation and energy (temperature recalculation) OFF

4.3.1. REACTORS INDEPENDENCE STUDY

Grid independence study is an essential part of CFD simulations to arrive at the best grid resolution, after which the results become independent of the grid. Similarly, a reactor independence study is carried out to ensure the computational domain is accurately represented by the reactor network, and the results do not vary heavily with an increasing number of reactors after that point. It is expected that an increase in reactors would lead to an improved solution due to a better flow-field resolution. But at a certain value of reactors, the improvement would reach a convergence point, after which further increase in reactors would not contribute towards improvement in emission prediction. However, this limit can be different for major and minor species due to their highly different order of magnitudes. So, a trade-off is required to arrive at an optimum computational cost with serviceable accuracy in predicting all species concentration.

This study compares radial profiles for major and minor species with an increasing number of reactors until the volume integral average results converge, suggesting that the results became independent of the CRN resolution. Also, the CRN results should ideally get closer to the experimental results in terms of average value as well as radial profile at the best reactor resolution. This test is done for CRNs with 499, 1001, 2487 and 4590 PSRs.

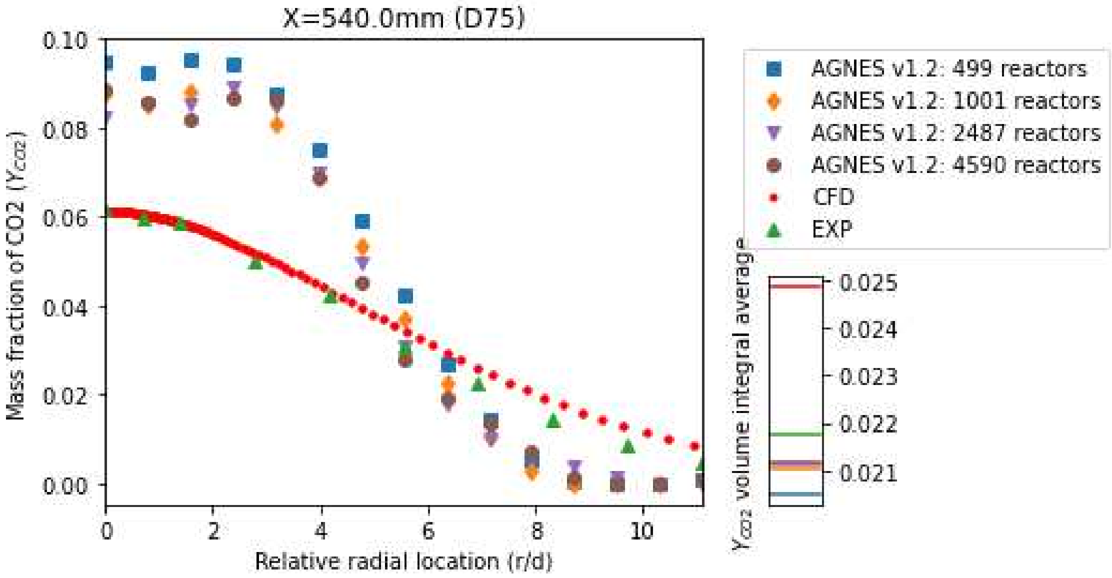

Figure 4.8: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (◆), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of CO_2 mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$)

Figure 4.8, 4.9, 4.10 and 4.11 provides radial profiles of CO_2, H_2O, CO and NO respectively at the most downstream axial location (D75) with experimental results. Figure 4.8 suggests that CO_2 concentration does not vary heavily and stay close to experimental results after increasing reactor resolution above 1001 PSRs. Similar trend is also visible in Figure 4.9 for the concentration of H_2O. Average concentration at this axial location also tends to converge after 1001 PSRs network. This can be concluded that 1001 reactors provides a suitable amount of resolution independence with a good balance in computational load.

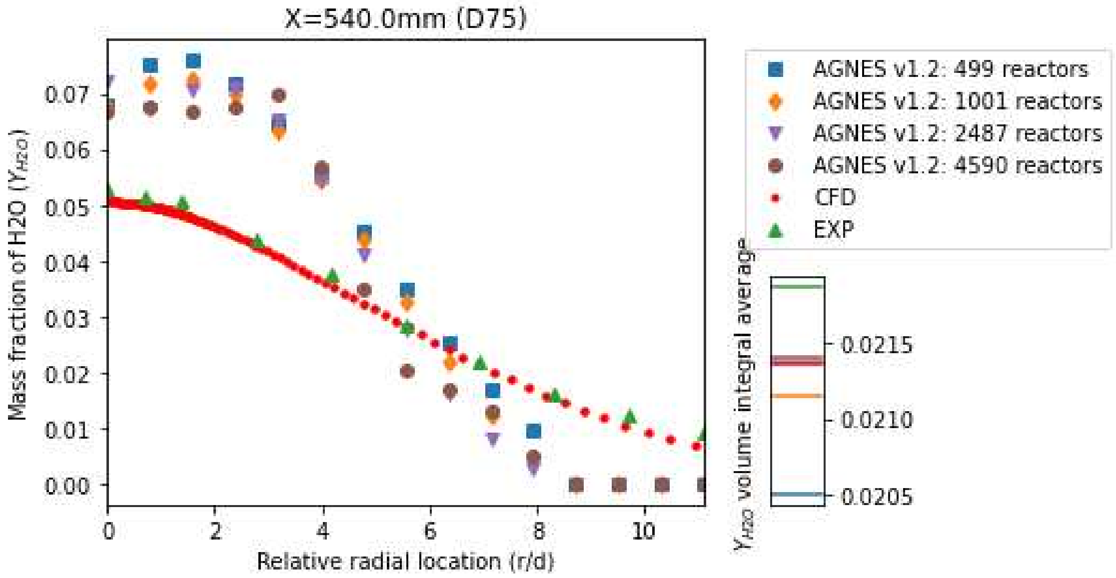

Figure 4.9: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of H_2O mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75)

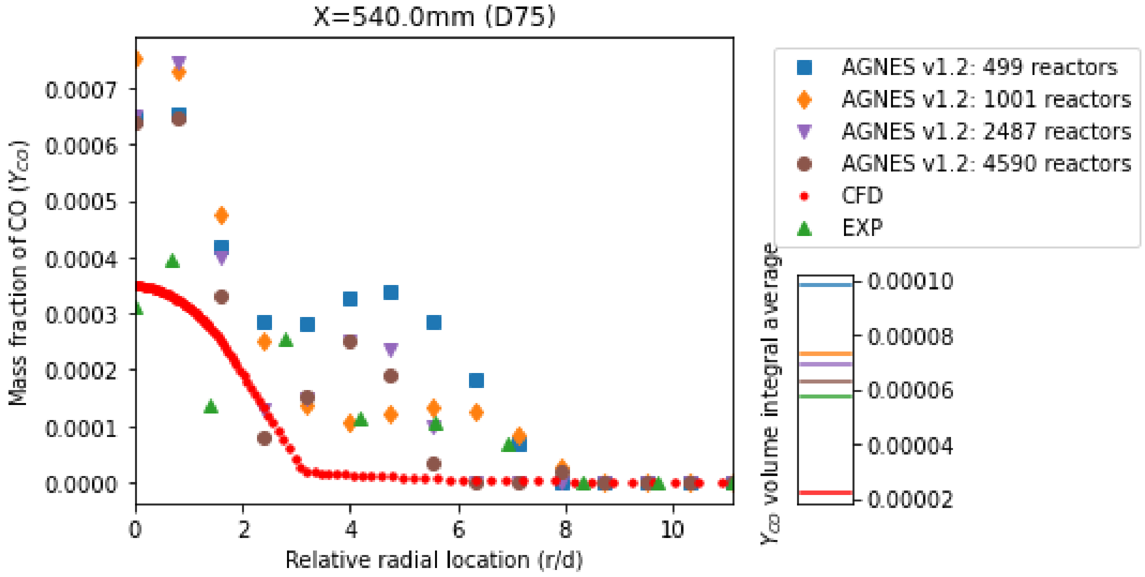

Figure 4.10: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of CO mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75)

However, based on Figure 4.10 and 4.11 minor species such as CO and NO concentration tend to improve with increasing PSR count, which is expected due to their respective magnitude to be in order of 1e-05. CO is formed due to relatively fast reaction. On contrary, NO production is dependent on several NO_X

pathways which are slow and consists of several intermediate species. All these reaction get well-resolved with a high reactor resolution but could lead to increased computational needs. So, there is a need for a trade-off between accuracy and computational time. 1001 PSRs network provide the necessary accuracy in predicting minor species and the average value is close to the experimental results.

Figure 4.12 to 4.15 shows the volume integral average value of species concentration at D75 axial location. It is quite clear that major species move closer to the experimental results and at 1001 PSR count reach a convergence point or there is no significant improvement in results. Minor species continue to improve and move closer to experimental results. An additional test with 10000 reactors is recommended to verify if there are any other interesting observations in case of minor species.

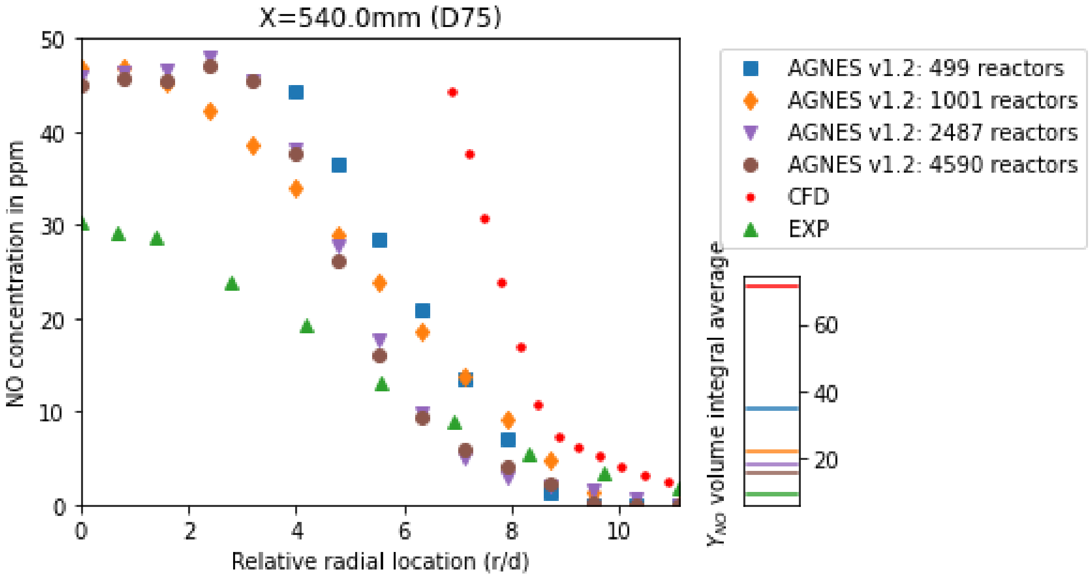

Figure 4.11: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results using AGNES v1.2 (left) for CRNs with 499 (■), 1001 (♦), 2487 (▼) and 4590 PSRs (●), and volume integral averages (right) of *NO* mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$)

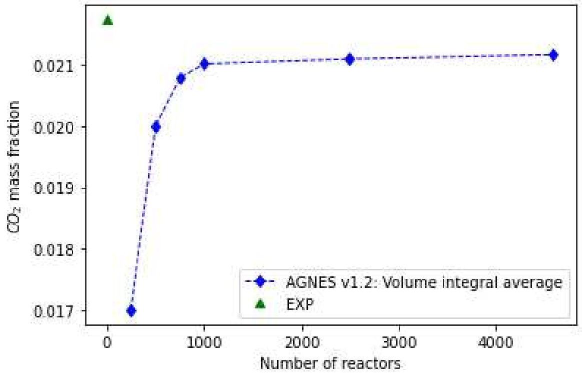

Figure 4.12: Variation in volume integral averaged CO_2 mass fraction at D75 axial location with increasing number of reactors

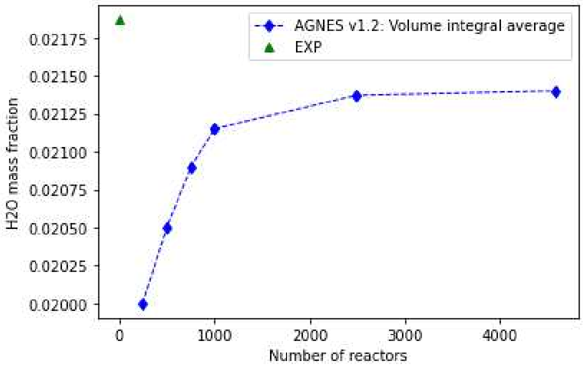

Figure 4.13: Variation in volume integral averaged H_2O mass fraction at D75 axial location with increasing number of reactors

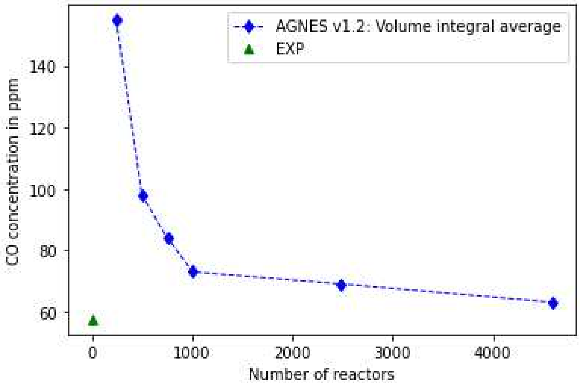

Figure 4.14: Variation in volume integral averaged CO mass fraction at D75 axial location with increasing number of reactors

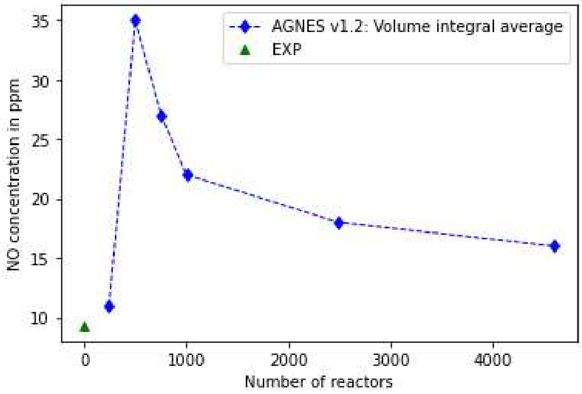

Figure 4.15: Variation in volume integral averaged NO mass fraction at D75 axial location with increasing number of reactors

4.3.2. MAJOR SPECIES CONCENTRATION

The aim of this subsection is to validate the newly developed solver implementation, aptly named AGNES v1.2 using the pre-research version AGNES v1.1 which was developed by Rishikesh (2018) [?] and Maaike (2019) [?]. Accuracy in predicting major species concentration of both these version are compared to ensure that with an improved solver, there is no compromise in the accuracy and to understand how these predictions compare with the experimental data and CFD simulation results.

METHANE CH_4

Methane is the limiting reactor in case of Sandia flame D due to lean combustion mixture. So, it is expected that CH_4 mass fraction to continuously reduce to finally value of zero with each passing subsequent downstream locations. Figure 4.16 to 4.20 correctly show the capability of AGNES in calculating methane concentration more accurately compared to CFD simulation.

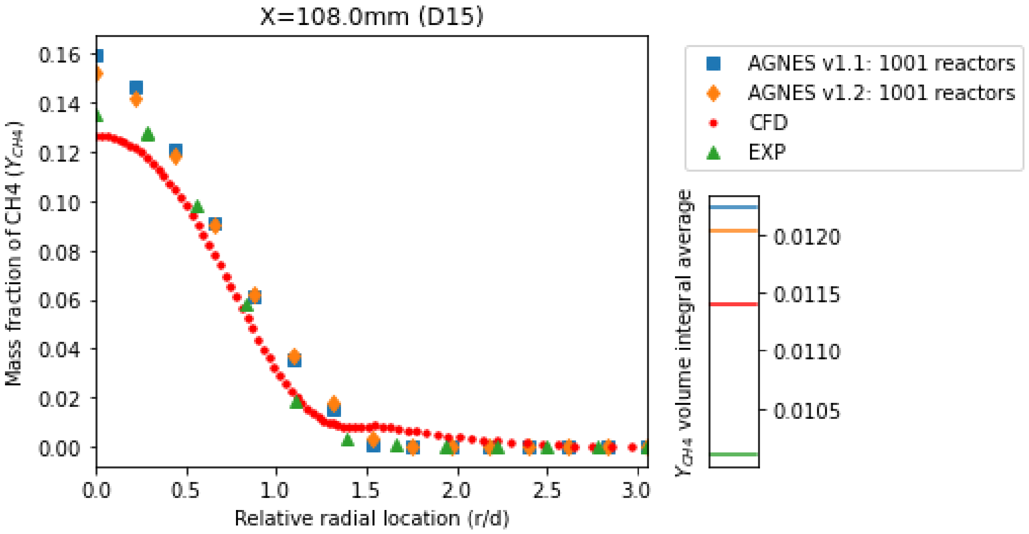

Figure 4.16: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D15 (x = 108 mm & $x/d_{jet} = 15$)

For upstream locations like D15 and D30, AGNES lightly under-estimates the methane consumption, mainly in the radial location ranging from $r/d = 0$ to $r/d = 1.5$. It is also evident from the volume integral average plot that both AGNES v1.1 and v1.2 have a higher predicted Y_{CH_4} compared to both CFD and experimental data. One reason might be the center jet is not reacting as much, which suggests a slow diffusion of O_2 despite the modelled turbulent diffusion. However, this issue seems to get fixed as move downstream to axial location D45, as seen in Figure 4.18, where AGNES predicted radial profiles closely match that of the experiment.

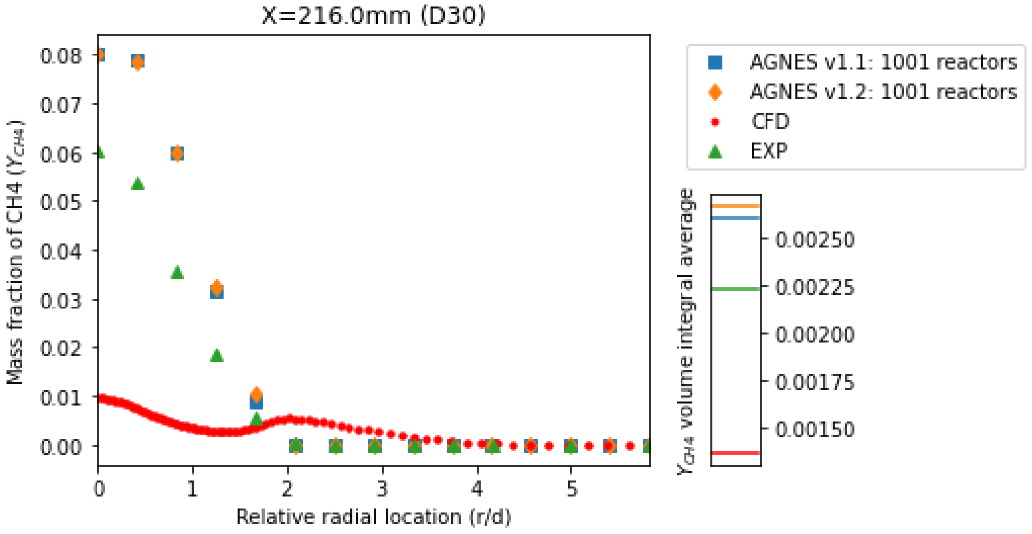

Figure 4.17: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30)

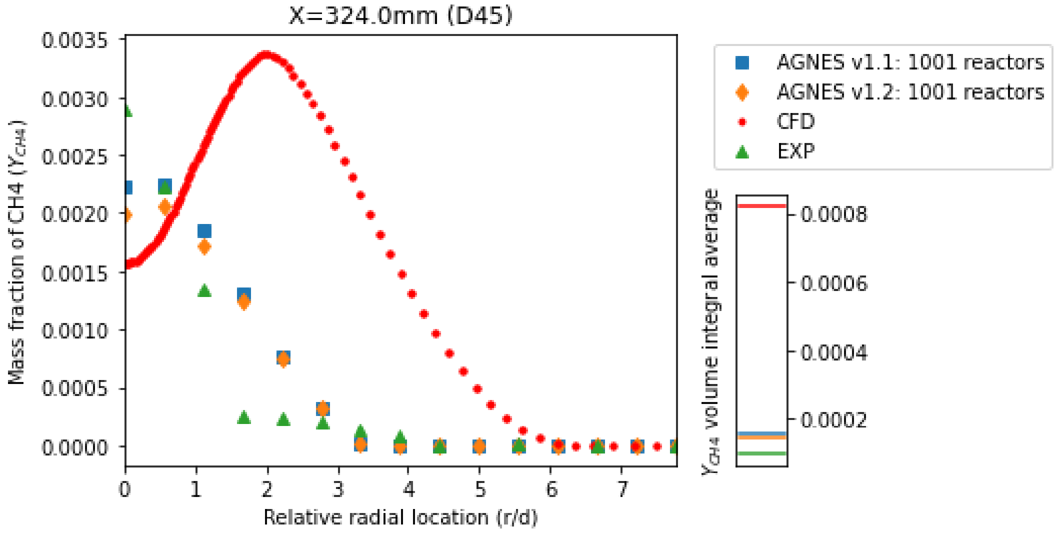

Figure 4.18: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45)

For downstream axial locations like D60 and D75, CH_4 concentration predicted by AGNES is close to zero, similar to the experimental results. This might be due to large amount of minor intermediate species that are formed from several dissociation reactions. One reason contributing to this can be the quick diffusion of methane into the hot pilot stream. On the other hand, CFD relatively struggles to accurate calculated methane mass fraction at these locations. It extremely over-estimates volume integral averaged Y_{CH_4} by almost five times to that of experimental result.

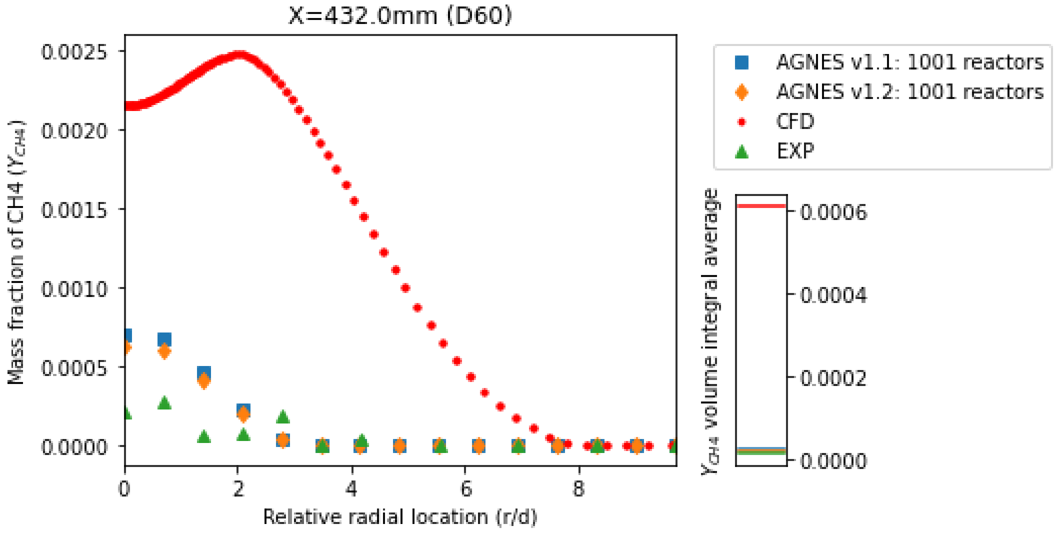

Figure 4.19: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D60 (x = 432 mm & x/d$_{jet}$ = 60)

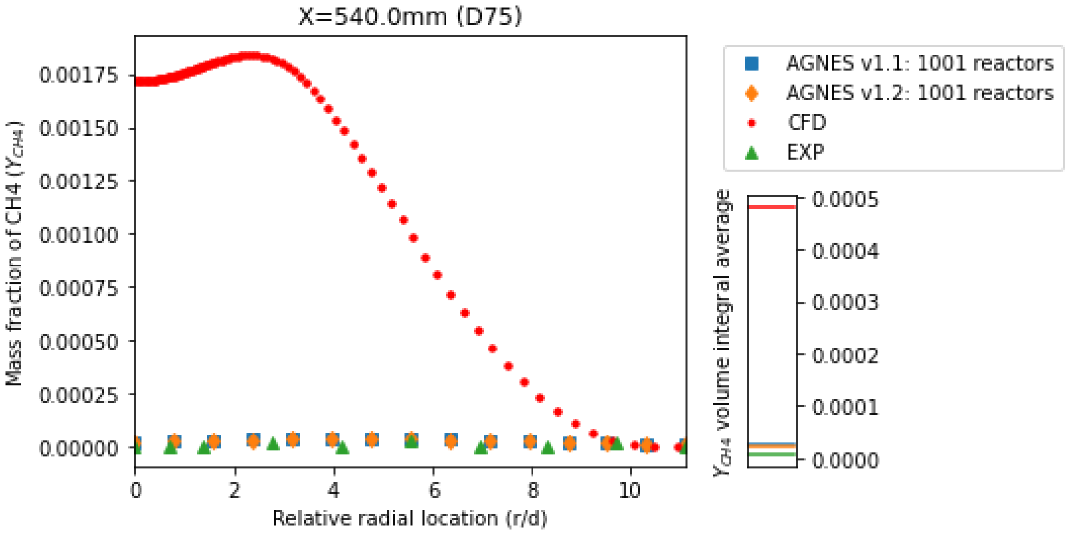

Figure 4.20: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CH_4 mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75)

OXYGEN O_2

Oxygen is the next reactor species which acts as oxidizer for the combustion process. The combustion process or methane depletion is heavily dependent on the correct convective and diffusive transport of oxygen. Due to lack of molecular diffusion, AGNES suffers slightly in accurately predicting the radial profile but the average value comes close to the experimental results. Figure 4.21 to 4.25 shows the radial profile of oxygen mass fraction at five axial locations.

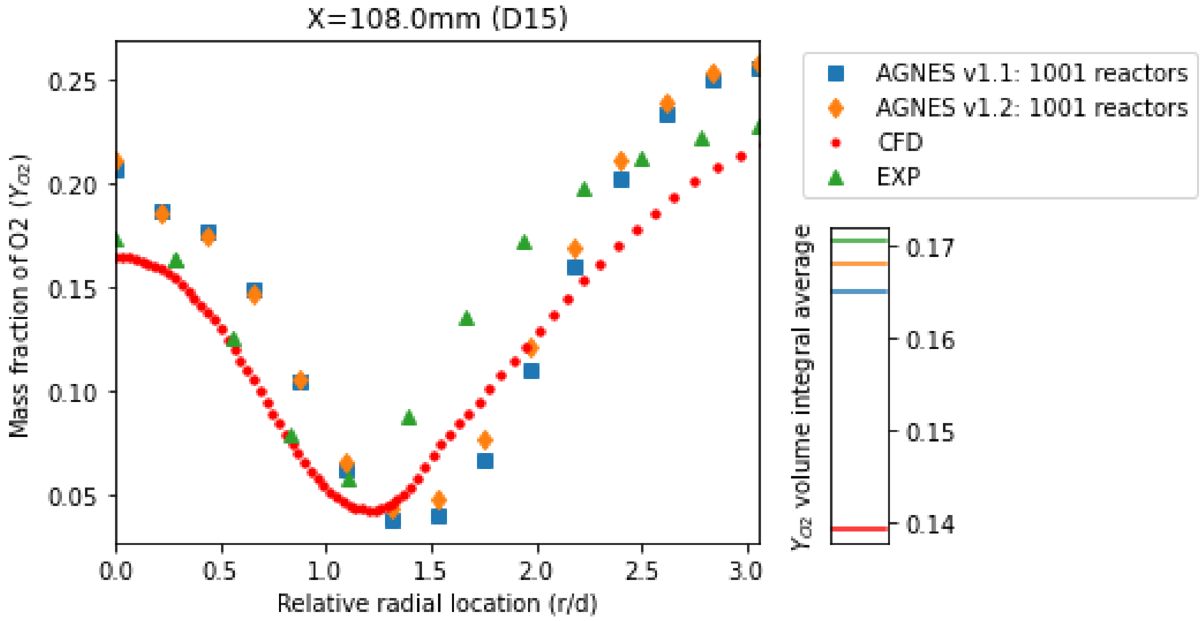

Figure 4.21: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D15 (x = 108 mm & x/d$_{jet}$ = 15)

At the most upstream location D15, AGNES over-predicts O_2 concentration close to the symmetric axis ($r/d = 0$) but starts for decrease as we move radially outward till $r/d = 1.5$ and after which it starts increasing again to match the experimental profile. Overall it does better at calculating oxygen compared to CFD simulation. Turbulent diffusion makes the results almost correct but molecular diffusion is required to accurately recreate the transport processes happening in the real world.

For locations D30 and D45, AGNES does well and matches closely with experiment and CFD. On average, it predicts slightly higher but not heavily in contrast to the CFD results. Correct diffusion of O_2 is a proof of correct representation of high temperature areas due to the discretized flow field with reactors. However, as we move further downstream, there are steep trends in O_2 concentration despite the use of turbulent diffusion. But the volume integral average is within the acceptable range. Please note the difference in integral is only present in the plotted region. Overall oxygen (O) outflow is conserved.

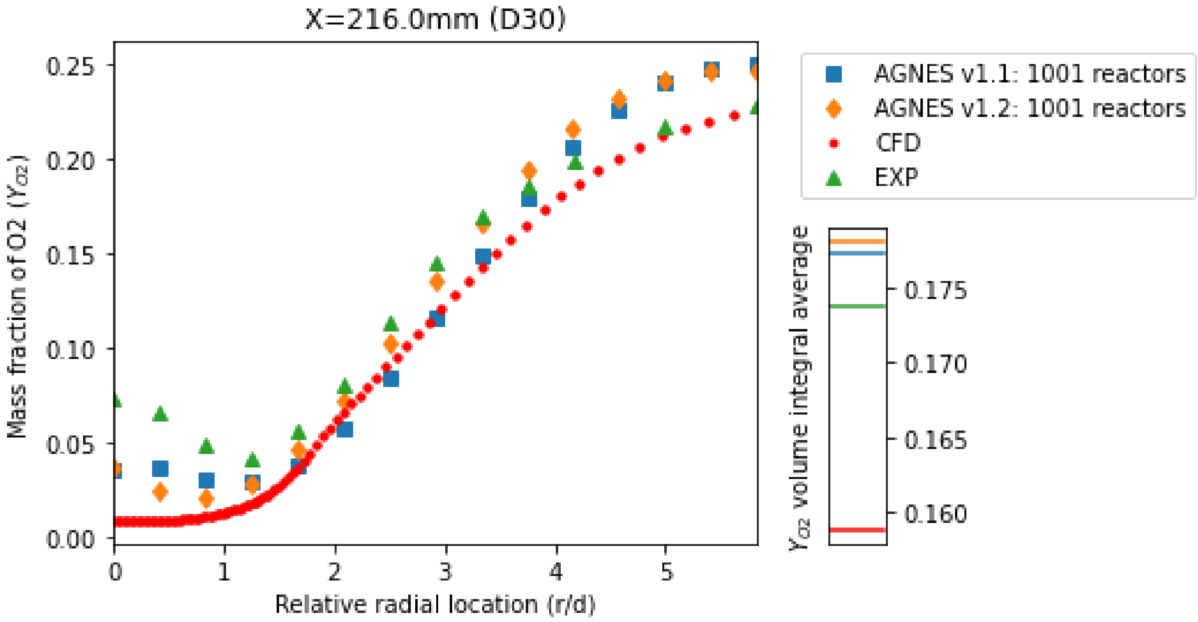

Figure 4.22: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30)

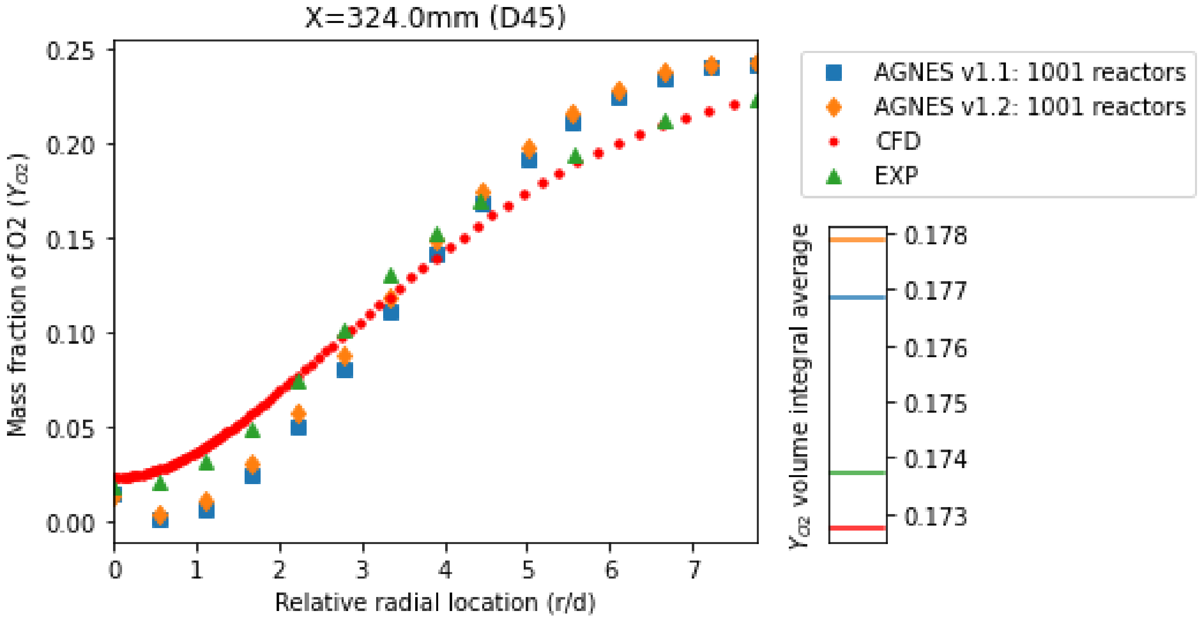

Figure 4.23: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45)

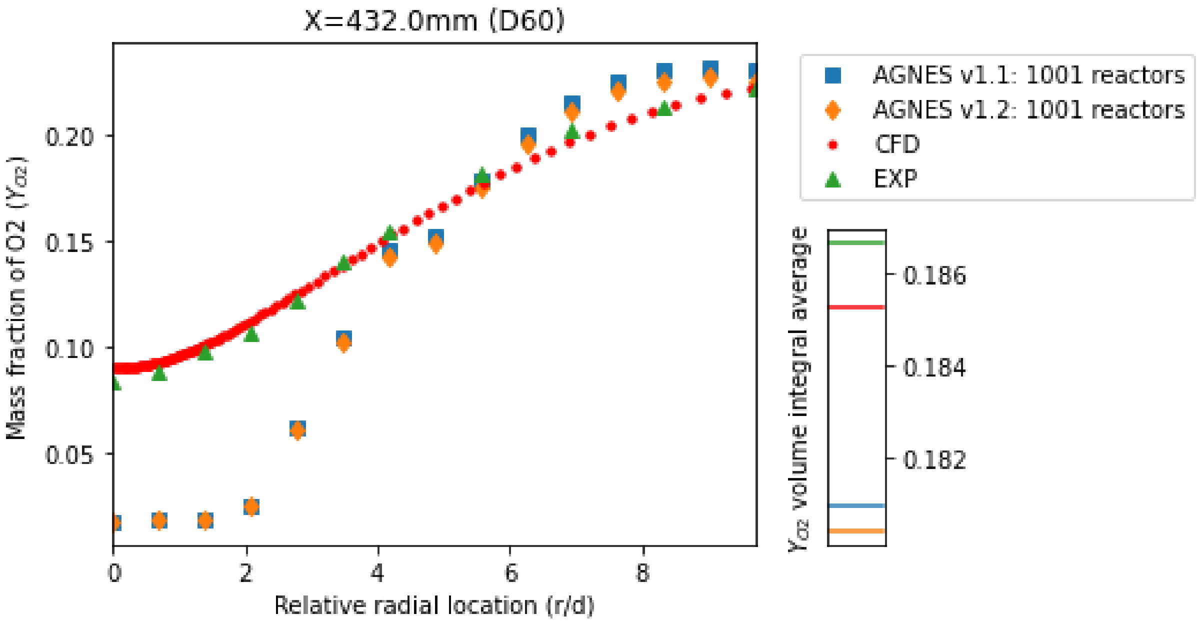

Figure 4.24: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$)

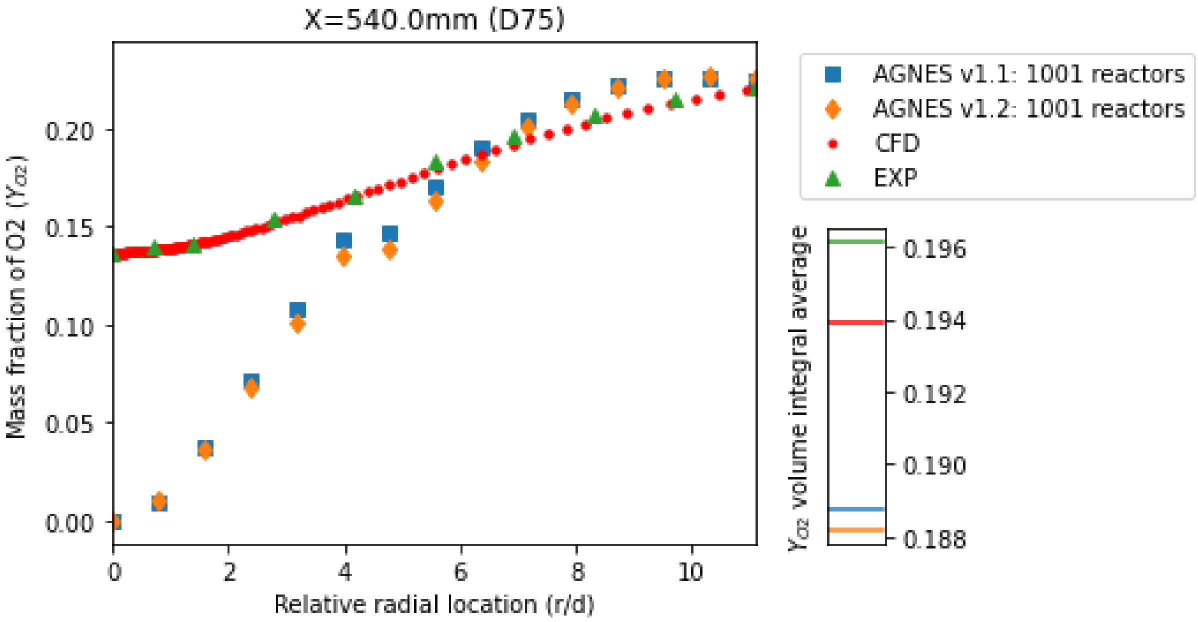

Figure 4.25: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of O_2 mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$)

WATER VAPOUR H_2O

Water vapour is one of the main major species products of a hydrocarbon combustion reaction. Figure 4.26 to 4.30 presents the radial profile of H_2O mass fraction at five axial locations along with comparison of volume integral averaged values between AGNES, CFD and experimental results. Both CFD and AGNES capture the experimental trends quite well for the upstream axial locations D15 to D45.

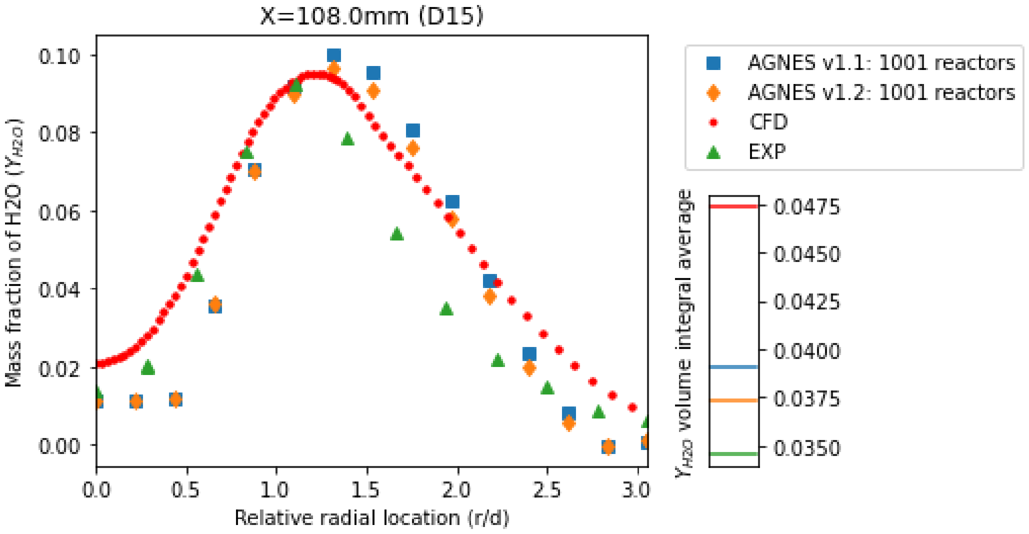

Figure 4.26: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of H_2O mass fraction at axial location D15 (x = 108 mm & x/d$_{jet}$ = 15)

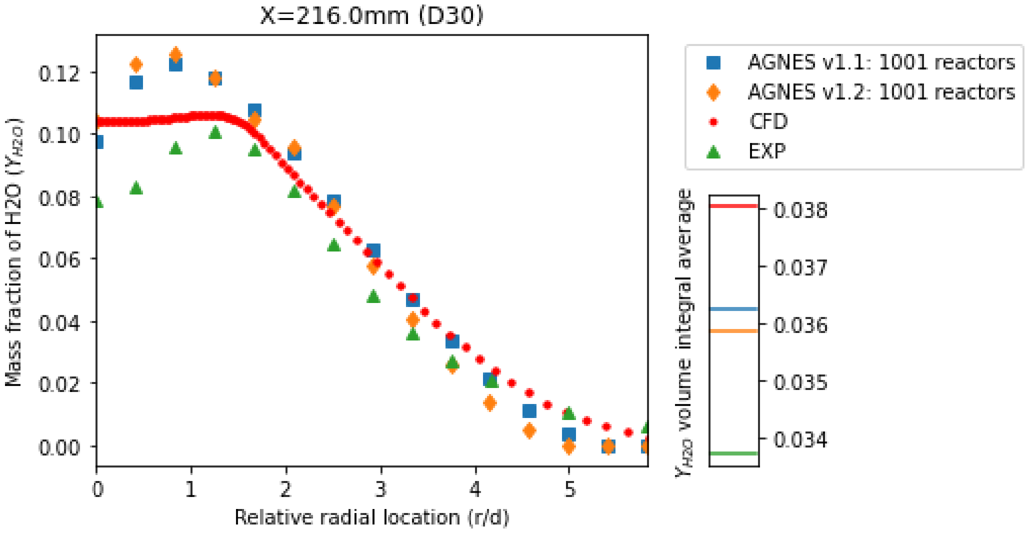

Figure 4.27: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of H_2O mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30)

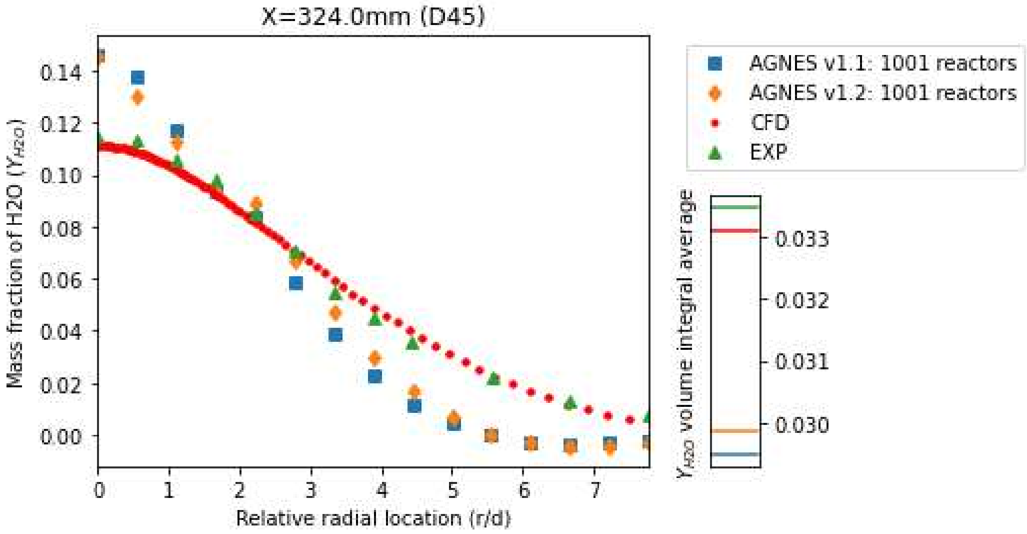

Figure 4.28: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of H_2O mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45)

But after D45 as seen from Figure 4.29 and 4.30, predicted H_2O concentration steeply decrease radially outward. This trend is exactly opposite to that of O_2. This might be explained by the shape and size of the clusters in the CRN mostly at far axial locations. As also pointed by Maaike [**?**] in her thesis, clusters tend to elongate along the axial direction. Along with the use of PSR, species travel in the axial direction is exaggerated. Although the radial profiles are matching with experimental trend, integral values are close enough to the experimental ones.

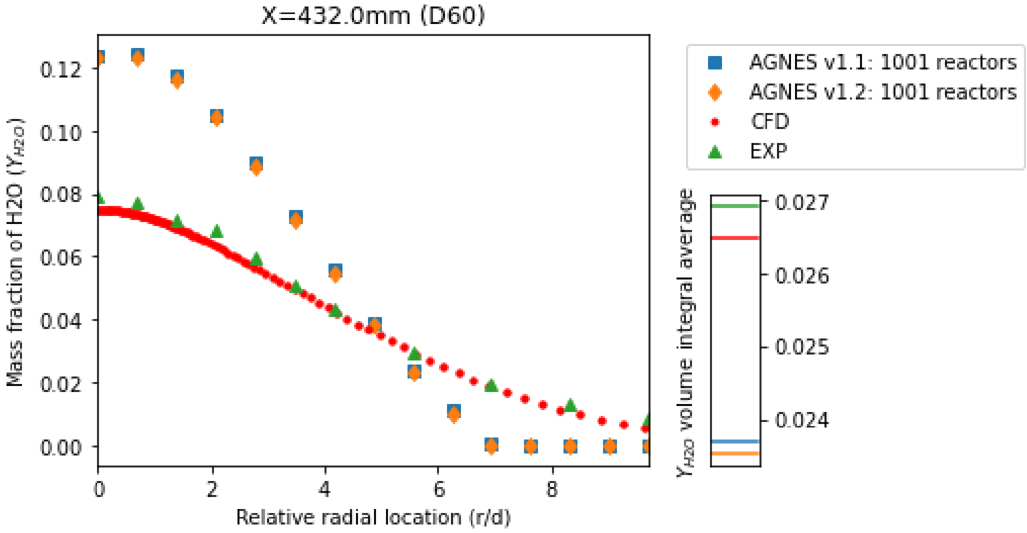

Figure 4.29: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of H_2O mass fraction at axial location D60 (x = 432 mm & x/d$_{jet}$ = 60)

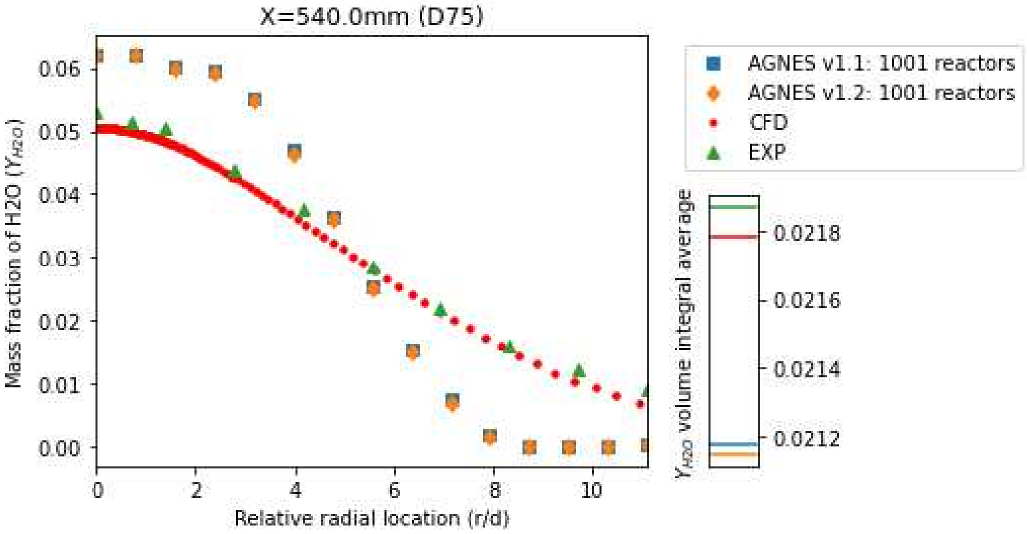

Figure 4.30: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of H_2O mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75)

CARBON DIOXIDE CO_2

CO_2 concentration can provide an indication of completion of reactions and good convergence for any CFD or CRN solution. Again both version of AGNES perform similar to CFD and matches the experimental trend. The same issue as H_2O and O_2 trend continues for downstream location D60 and D75. Locations where O_2 concentration is under predicted compared to experimental data, CO_2 becomes over predicted and vice versa due to O atom conservation.

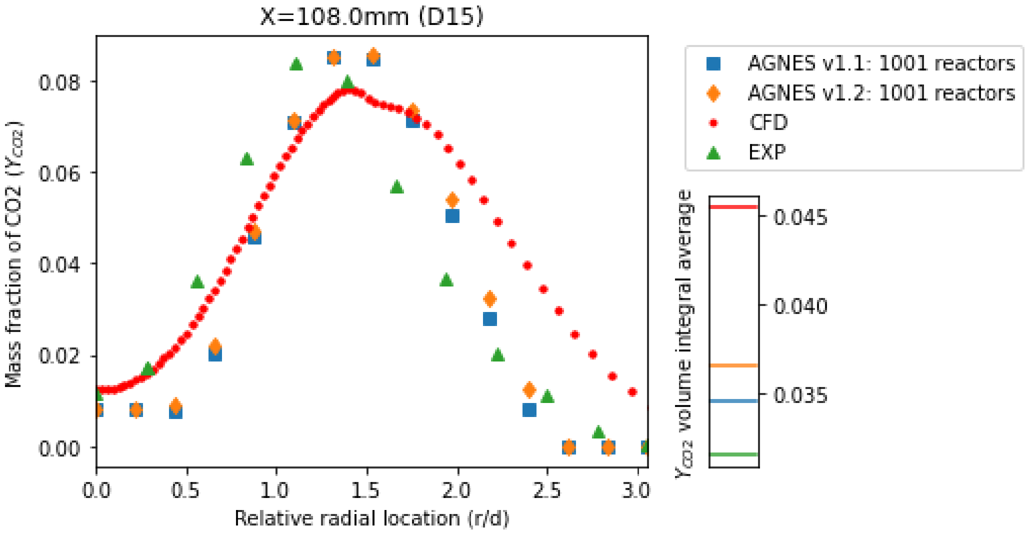

Figure 4.31: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D15 (x = 108 mm & x/d$_{jet}$ = 15)

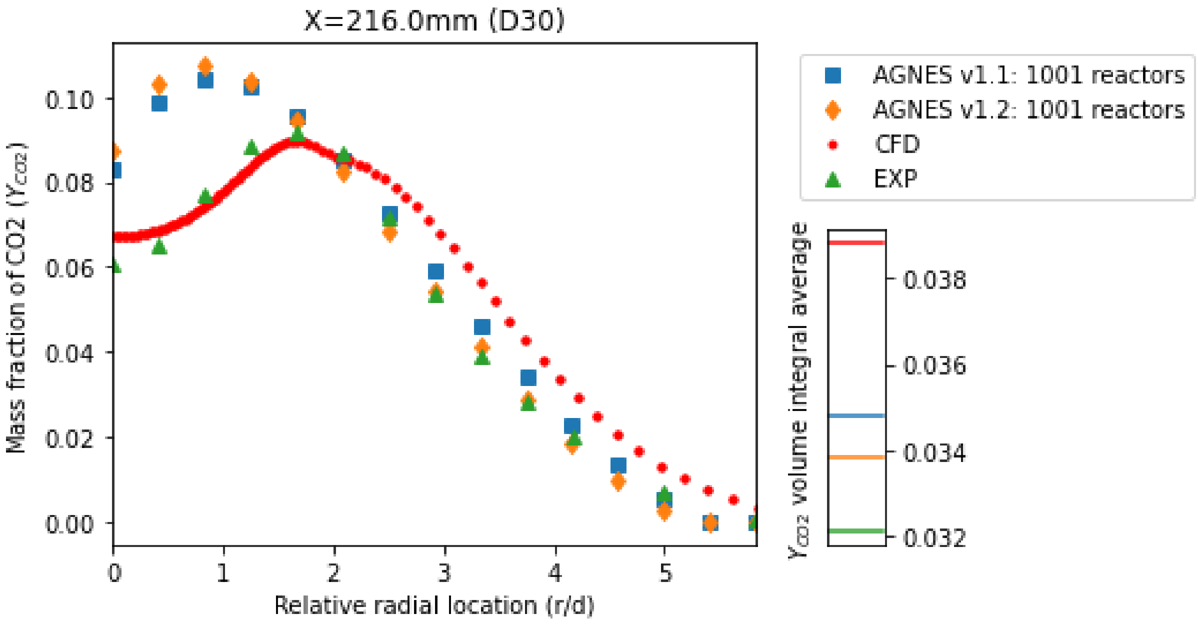

Figure 4.32: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30)

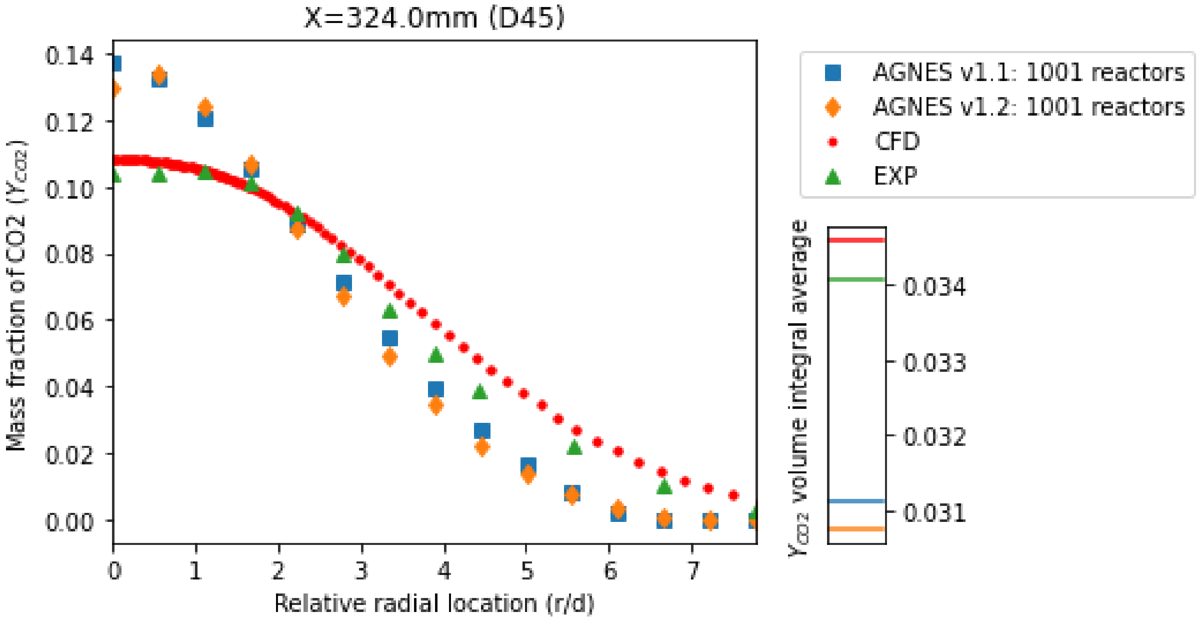

Figure 4.33: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45)

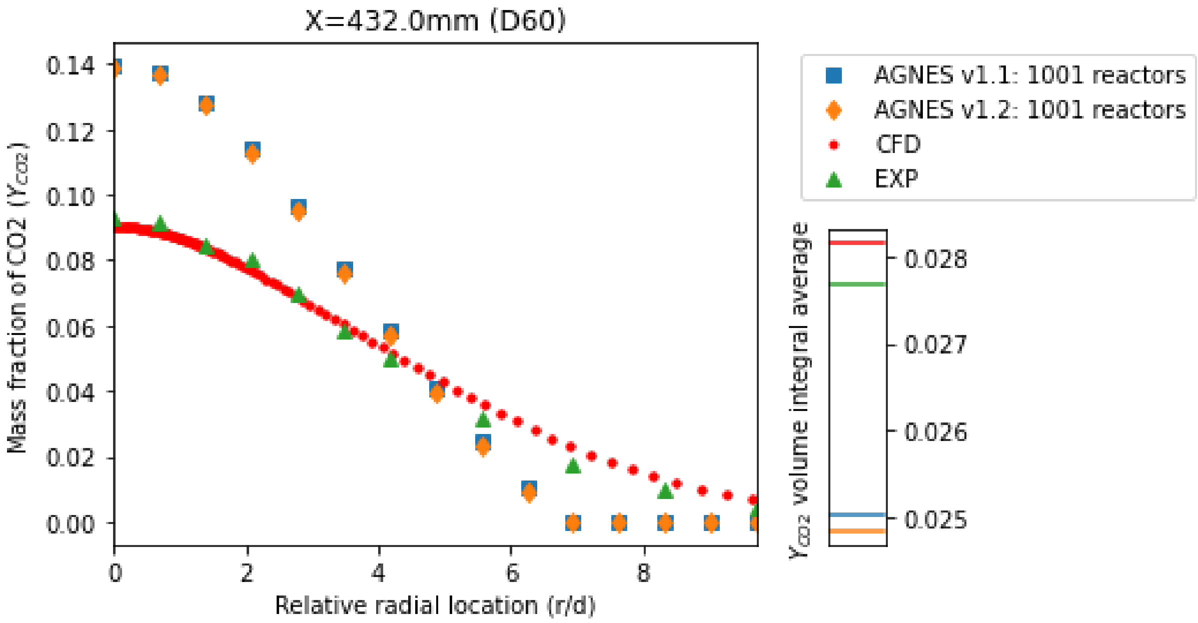

Figure 4.34: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D60 (x = 432 mm & x/d$_{jet}$ = 60)

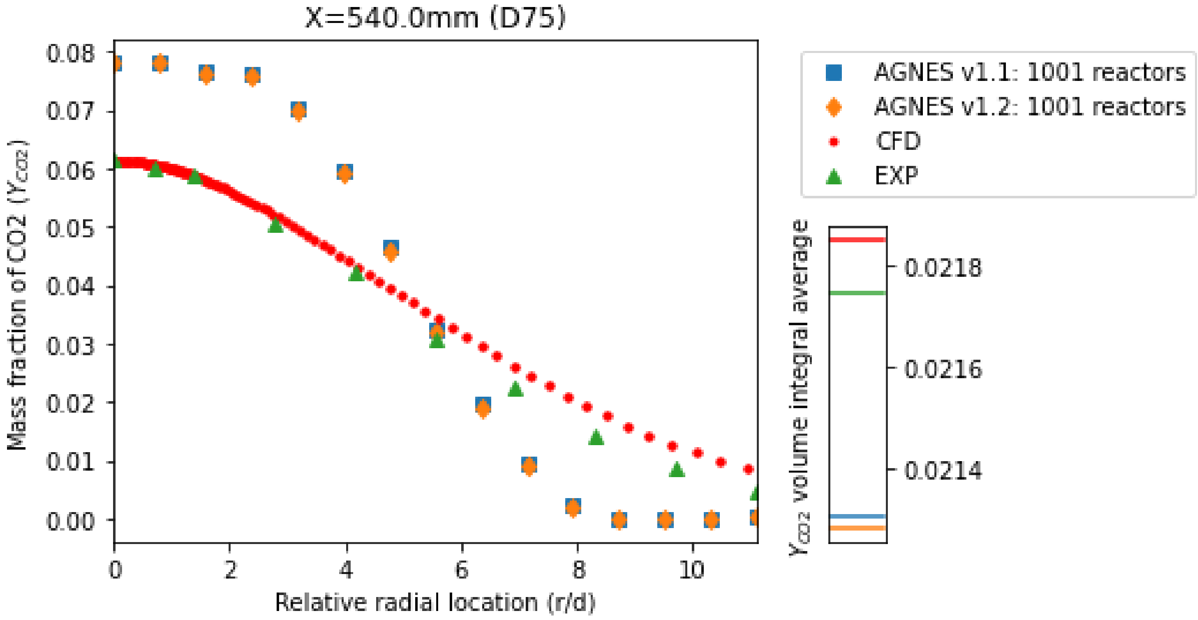

Figure 4.35: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO_2 mass fraction at axial location D75 (x = 540 mm & x/d$_{jet}$ = 75)

4.3.3. MINOR SPECIES CONCENTRATION

AGNES is specifically developed to improve minor species such as NO and CO concentration predictions than what is currently feasible using CFD simulations. Accuracy in predicting these minor species concentration using improved AGNES v1.2 is of paramount importance. In this section, these results are compared with pre-research AGNES v1.1, the experimental data and CFD simulation results.

CARBON MONOXIDE CO

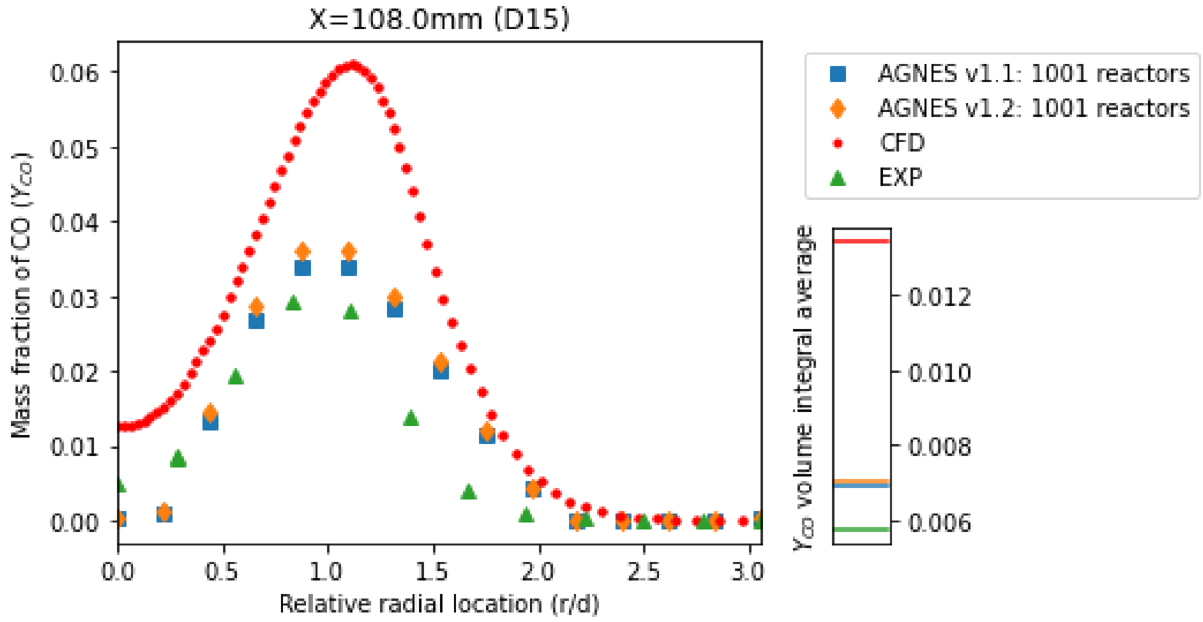

Figure 4.36: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D15 (x = 108 mm & x/d$_{jet}$ = 15)

One of the harmful products of hydrocarbon combustion is carbon monoxide CO. It is formed due to incomplete combustion and poor mixing. CO is produced due to oxidation of hydrocarbon radicals like CH_3 and CH_2O in the preheat and ignition region. From Figure 4.36, for location D15 CO concentration peaks at r/d=1.0 which is similar to both experimental and CFD results. This might be due to high CO production in the high temperature zone. CFD over predicts CO concentration at upstream locations but performs better as we move to downstream axial locations. AGNES on the other hand, has steep radial trends and high predicted CO concentration at D60 and D75 locations which might be due to incomplete oxidation of CO into CO_2. Locations where O_2 concentration is low (shown in Figure 4.24), also shows high CO accumulation. Maybe with molecular diffusion modelled, this over prediction can be corrected.

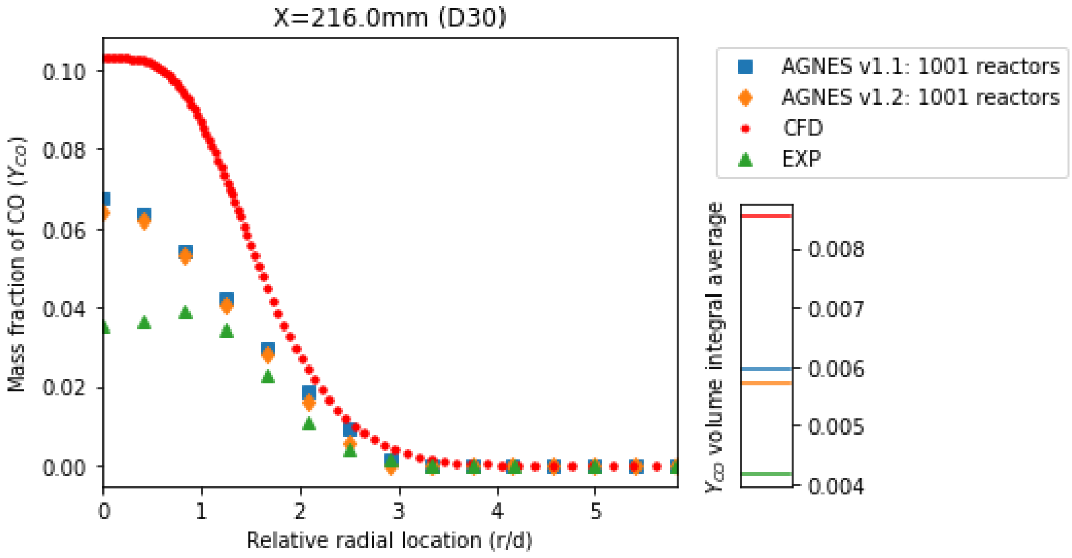

Figure 4.37: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D30 (x = 216 mm & x/d$_{jet}$ = 30)

Figure 4.38 to 4.40 show a decreasing trend in CO concentration, similar to experimental data. This decrease in CO is attributed to its oxidation into CO_2 in presence of O_2 or OH radicals. As we move farther from high temperature zones, CO oxidation is dominated by OH radical due to their elementary reaction's low activation energy. Similar to CO_2, more CO is formed near the symmetric axis (r/d <2) as evident from high consumption of oxygen (seen in 4.24).

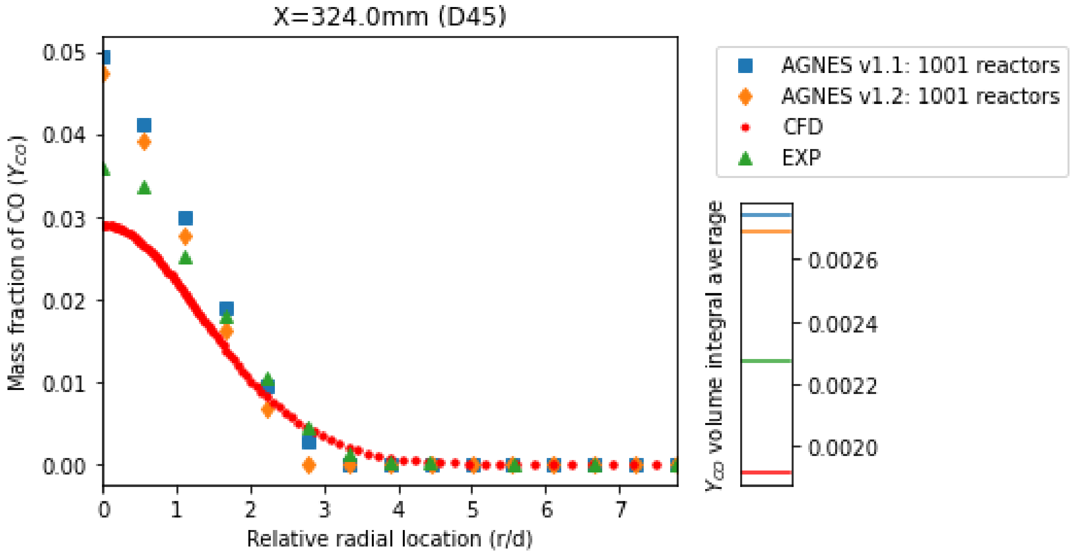

Figure 4.38: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D45 (x = 324 mm & x/d$_{jet}$ = 45)

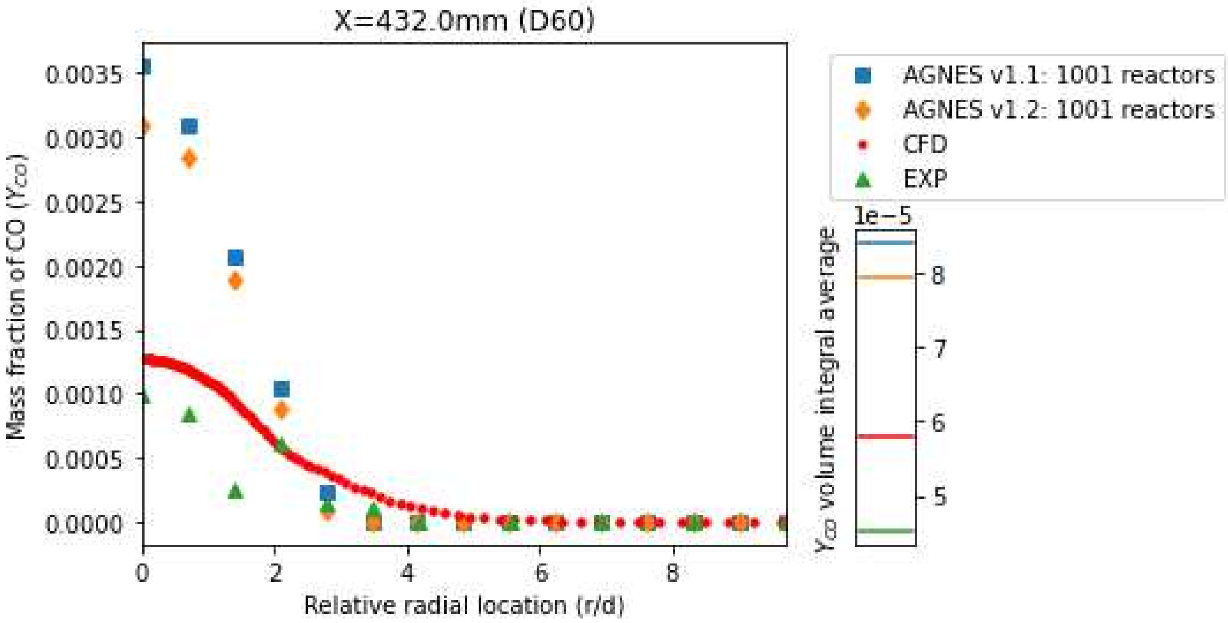

Figure 4.39: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$)

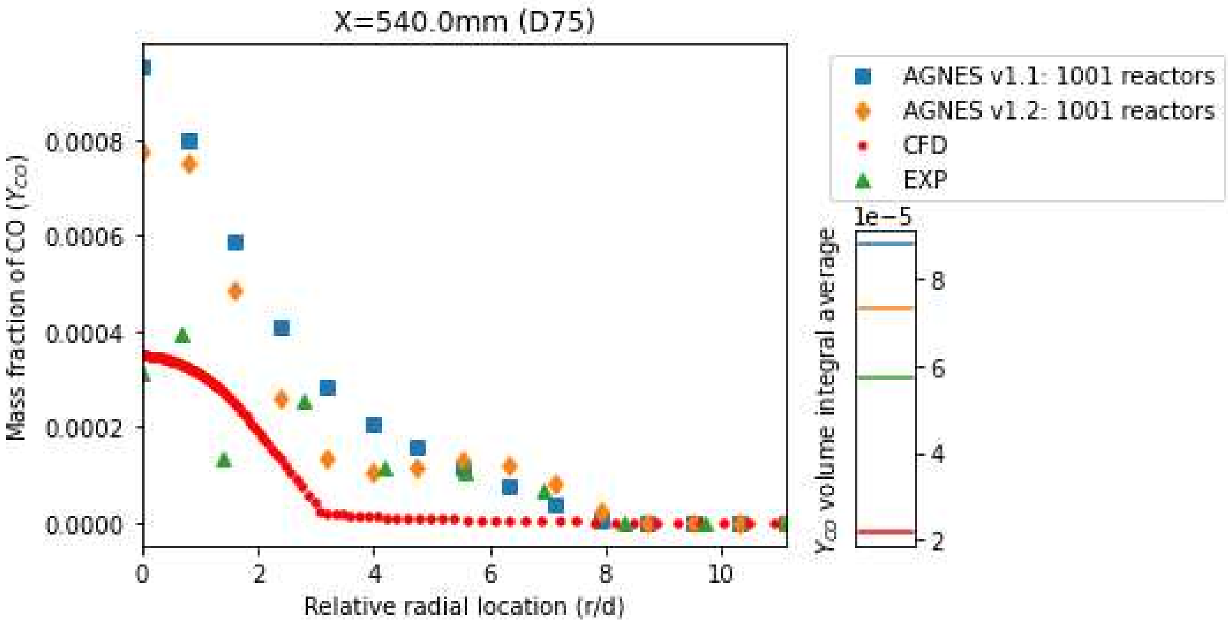

Figure 4.40: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of CO mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$)

NITROGEN OXIDES NO_X

NO_X consists of Nitric oxide NO, Nitrogen dioxide NO_2, Nitrous oxide N_2O. NO_X production is an extremely slow process and follows different pathways based on operating conditions. Moreover, only results for NO is discussed because it is predominantly produced compared other NO_X species. Figure 4.41 to 4.45 shows a marked improvement in NO predictions at all axial locations. As expected, CFD severely over predicts NO concentration which is due to ANSYS NO_X post-processing model.

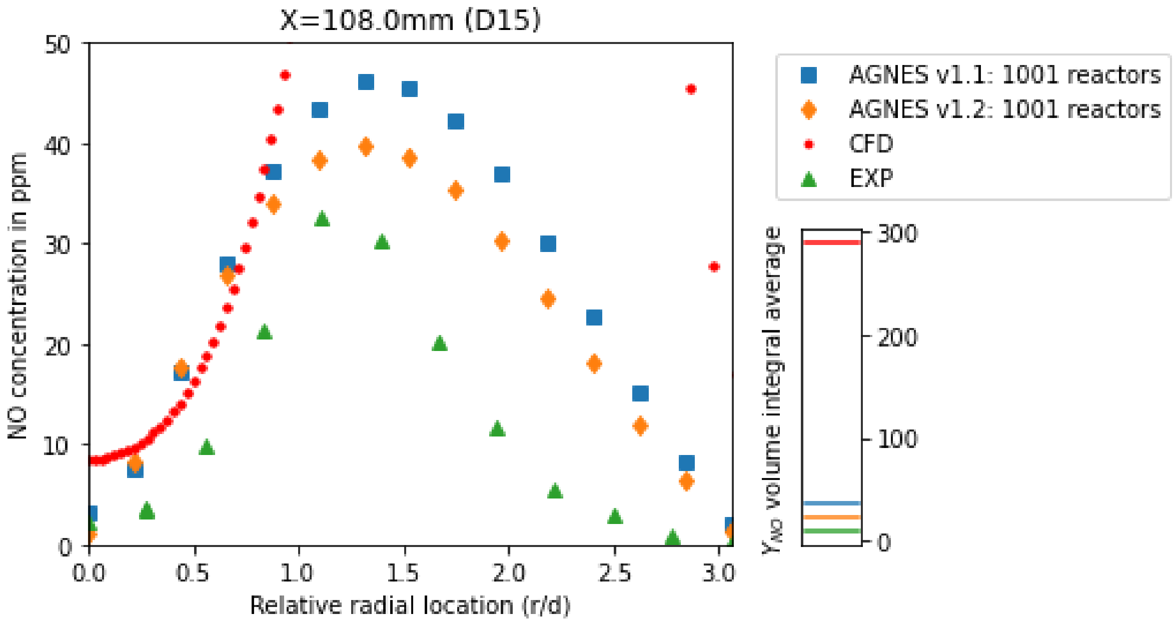

Figure 4.41: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D15 (x = 108 mm & x/d_{jet} = 15)

At axial locations D15 and D30, average temperature is high (around 1800 K) for radial locations close to the symmetric axis (r/d<1.5). Due to this high temperature, high amount of NO is produced due to Zeldovich NO_X pathway. Both AGNES v1.1 and AGNES v1.2 shows similar radial trend as experiments, which is first gradual increase to reach a peak and then decrease after a certain radial location, as shown in Figure 4.41 and 4.42. Overall, NO production is hugely dependent on the temperature calculation, and hence for same CRN solved with energy equation would lead to much better results but will struggle at downstream location due to over predicted temperatures [?].

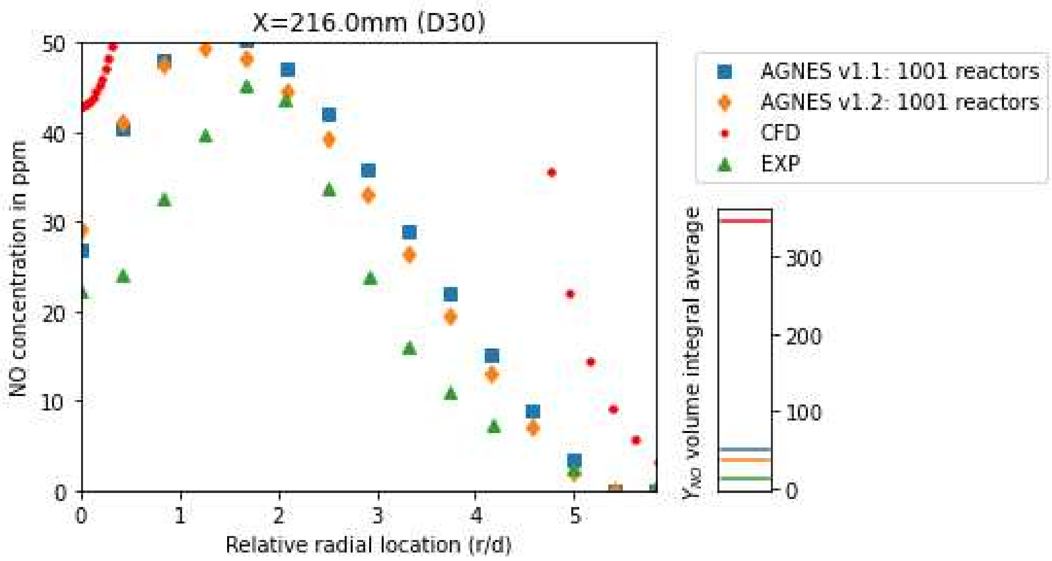

Figure 4.42: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D30 (x = 216 mm & $x/d_{jet} = 30$)

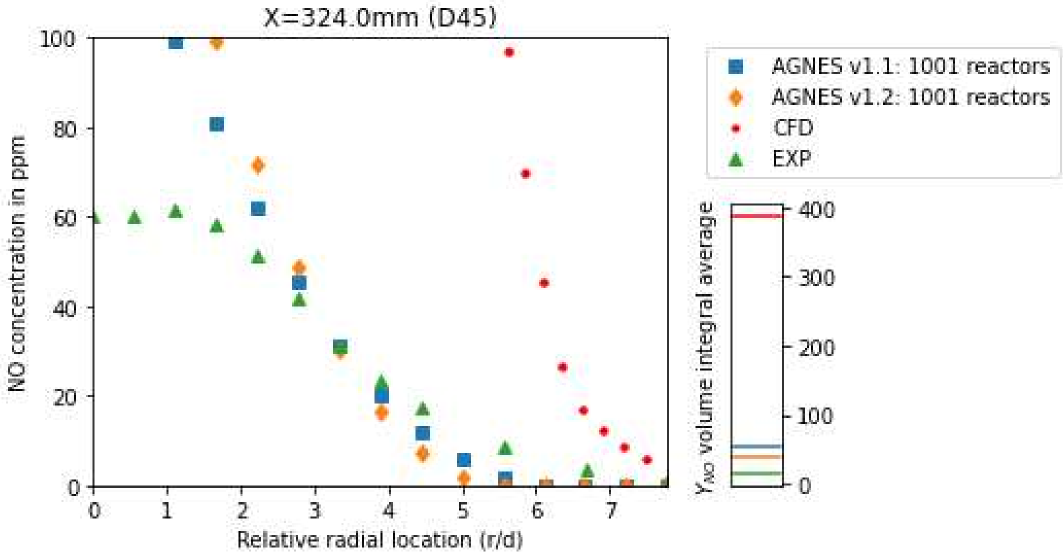

Figure 4.43: Radial profiles for CFD (●), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D45 (x = 324 mm & $x/d_{jet} = 45$)

For the other three locations, temperature is below the thermal NO_X production threshold, hence thermal pathway is suppressed. Therefore, NO is produced due to either NNH and HCN radicals. Again, AGNES radial trends match the experiment almost perfectly along with correct volume integral results. AGNES integral average NO concentration is around 24 ppm compared to CFD 78 ppm and experiments 9 ppm at D75 location. This successfully proves the capabilities of AGNES in accurately predicting minor species like NO.

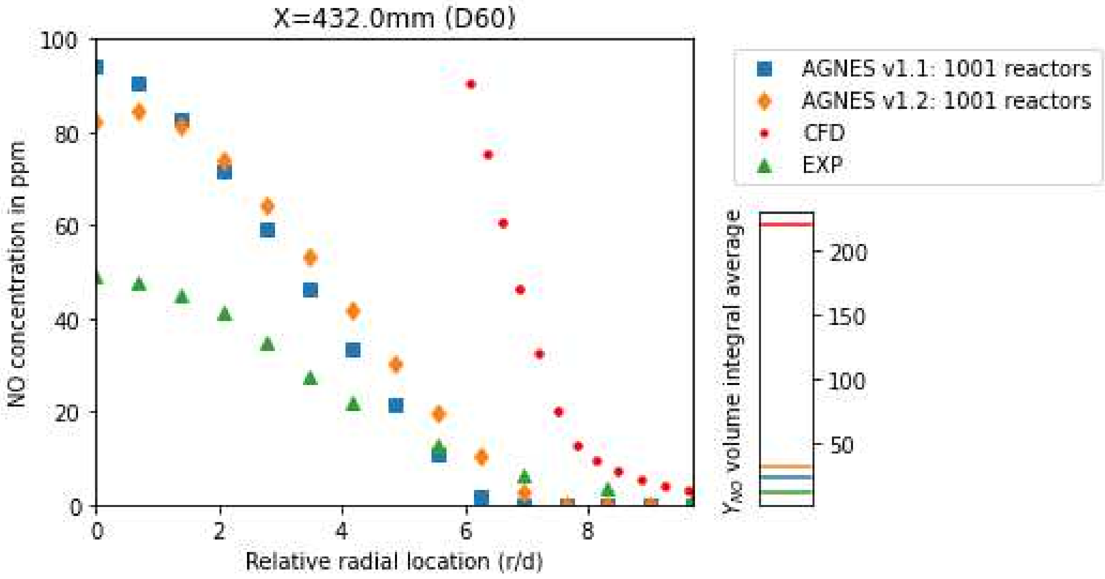

Figure 4.44: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D60 (x = 432 mm & $x/d_{jet} = 60$)

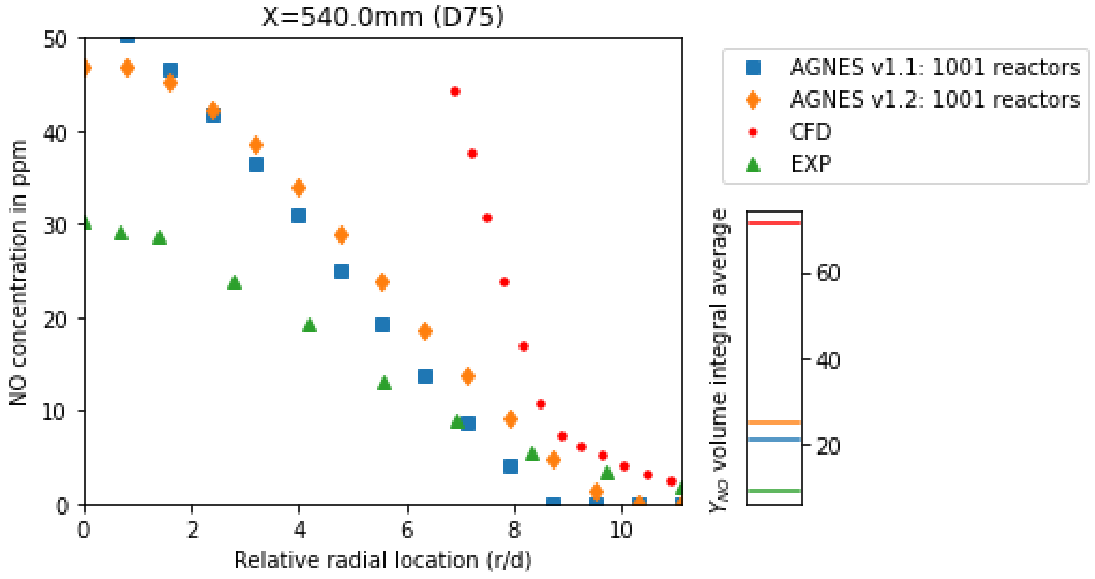

Figure 4.45: Radial profiles for CFD (•), experimental results (▲) and CFD-CRN predicted results (left) using AGNES v1.1 (■) and AGNES v1.2 (♦), and volume integral averages (right) of NO mass fraction at axial location D75 (x = 540 mm & $x/d_{jet} = 75$)

4.3.4. SOLVER PERFORMANCE

The effectiveness of improved solver for AGNES is tested by judging both its solution speed and convergence. The numerical performance of AGNES v1.2 is investigated in terms of execution time of the algorithm in solving CRN with different reactor counts. For this purpose, TU Delft's CentOS based HPC (high performance computing) cluster is used, which is made up of 12 nodes with each node containing 20 Intel Xeon E5-2670v2 (2.5 GHz) CPU cores.

SOLVING TIME

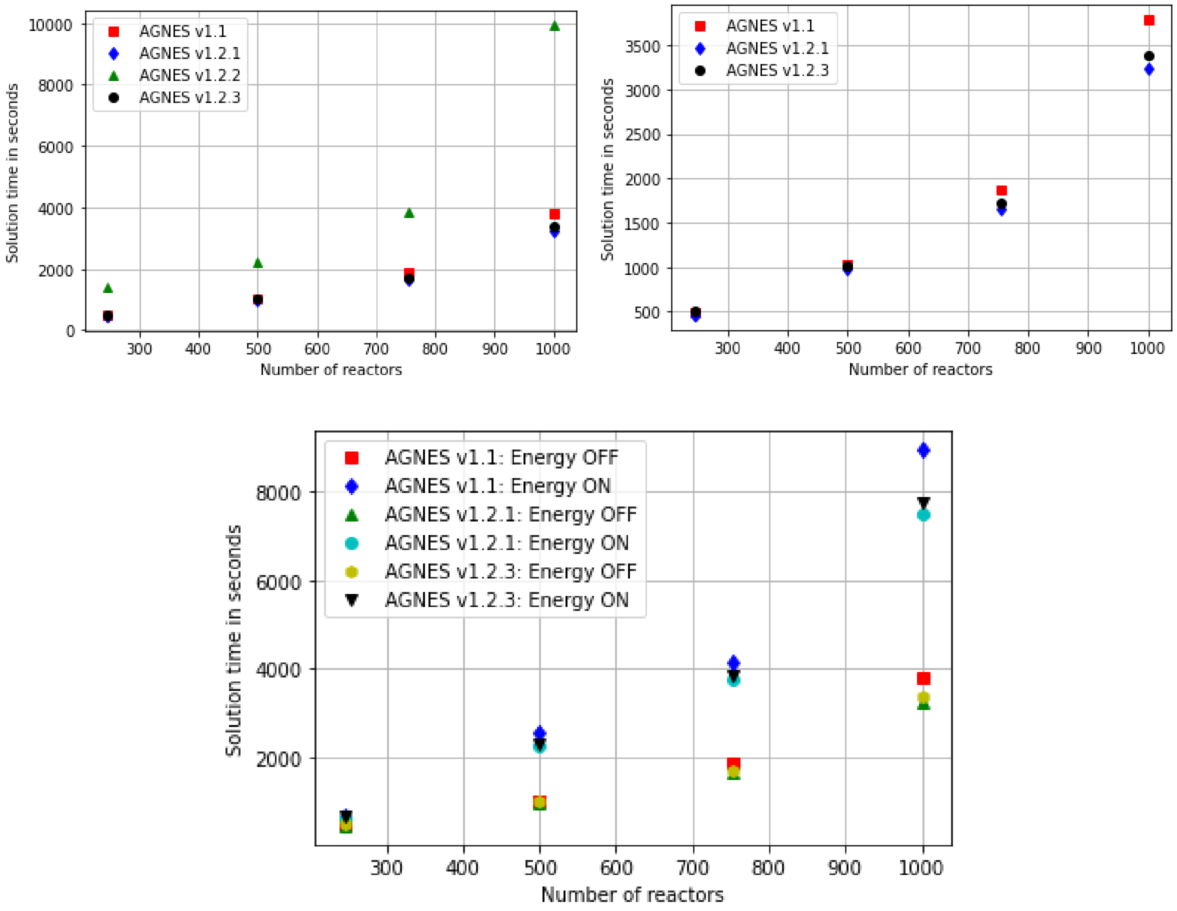

Figure 4.46: Solving time for various reactor size with energy calculation switch on/off using different solver implementations

Solving time of AGNES is mostly dependent on size of the CRN problem i.e number of reactors and species involved. Since the reaction mechanism for this project is kept fixed, the solution time only scales with the reactor count. Furthermore, for cases where temperature is recalculated inside reactors by solving energy equation, non-linearity and number of equations is greatly increased. Both clustering process and CFD post-processing take small fraction of the solution time.

As these factors are not changing for all the test cases, their impact on performance can be assumed to be negligible.

Figure 4.46 is a collection of plots showing solving time in seconds versus the number of reactors for the Sandia Flame D test case using all the previous discussed solver implementations. As expected, there is a non-linear trend in the solving time with increasing reactor number. AGNES v1.2.2 which is a pure time integration approach takes the most amount of computational time and hence, it excluded from further discussion. Both improved AGNES v1.2.1 and AGNES v1.2.3 offers faster solving time as compared to pre-research version. This solving time difference become more apparent with number of reactors as AGNES v1.1 heavy reliance of local solver becomes computational expensive as number of reactors keep on increasing. Although AGNES v1.2.1 uses the same local solver, most of the solution process is governed by global resolution methods, thus boosting its performance. In case of AGNES v1.2.3, the solution process is entirely done using global resolution using a combination of time integration and Newton's method. Even with Energy 'ON' i.e energy equation is solved along with species mass conservation equation, these newly developed implementations perform better compared to the pre-research version. CRN cases with Energy 'ON' become highly non-linear and increase the number of equations needed to be solved. This is evident from their solving time which is almost double of cases with Energy 'OFF'.

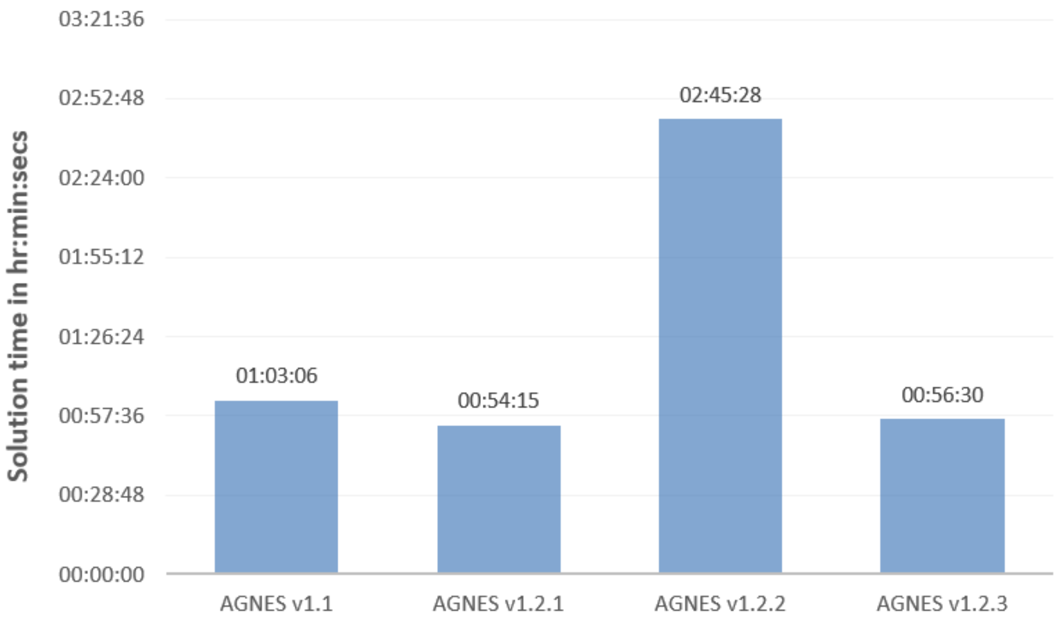

Figure 4.47: Solution time in hr:min:secs for solving a 1000 reactors network using different solver implementations

Figure 4.47 also presents the computational time for solving a 1000 reactors network for the Sandia flame D test case. To quantify this improvement in AGNES performance, the achieved speedup (S_p) is calculated as the ratio of pre-research AGNES solution time to newly developed AGNES v1.2, as shown in Equation 4.4. The achieved speedup is around 1.13 which means there is an improvement in speed by 13%.

$$S_p = \frac{t_{old}}{t_{new}} = \frac{3786}{3378} = 1.13 \tag{4.4}$$

Additionally, this performance can be further enhanced using an efficient way of Jacobian evaluation. As mentioned in Section 2.3, an analytical Jacobian is calculated for every single global step which consists of two sparse matrices: one with only transport contribution J_S and second Jacobian containing only reaction associated terms J_R. As J_S does not change through out the solving process, it can be saved as a constant Compressed Sparse Row (CSR) matrix. This matrix is then passed to the Jacobian evaluation function as an additional argument which is later added to the J_R to get the actual Jacobian. This approach results in saving 0.7-1 second of computational time for each Jacobian evaluation. Figure 4.48 presents the frequency of Jacobian evaluation based on reactor count. Jacobian is an extremely expensive operation but saving even 1 second for each evaluation in the entire solving process could prove to be an big improvement. Figure 4.49 shows the impact of this improved evaluation strategy on the solving time. On average, there is a reduction of 250 seconds in the computational time for a 1000 reactor network case. The final achieved speedup becomes:

$$S_p = 3786/3128 = 1.21 \tag{4.5}$$

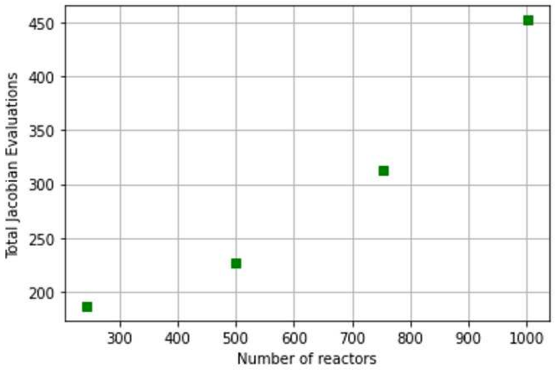

Figure 4.48: Frequency of Jacobian evaluations with increasing number of reactors

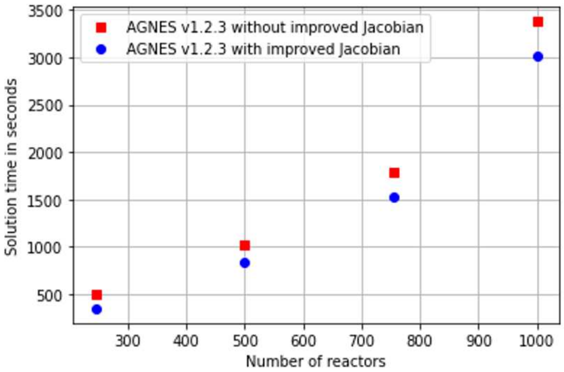

Figure 4.49: Improvement in solving time after using the improved Jacobian evaluation approach

TIME BREAKDOWN

Relative importance of the three resolution methods in terms of computational time is discussed in Figure 4.50. Pre-research version AGNES v1.1 puts more importance on the Cantera based local solver for the initial solving process. Global time integration stays active for less than 10% of the time and remaining solution is taken care by Global Newton's method. Due to the way AGNES v1.1 solver is implemented, global resolution using time integration is not used to its fullest extent. Whenever, the function residual starts increasing i.e the global Newton's method is diverging, global time integration is called. Although this way of alternating between different resolution methods proves to be effective in mitigating convergence issues but it become computationally heavy. Hence, a more optimised routine is developed by taking inspiration from the Stagni et al. KPPSMOKE solution approach [5].

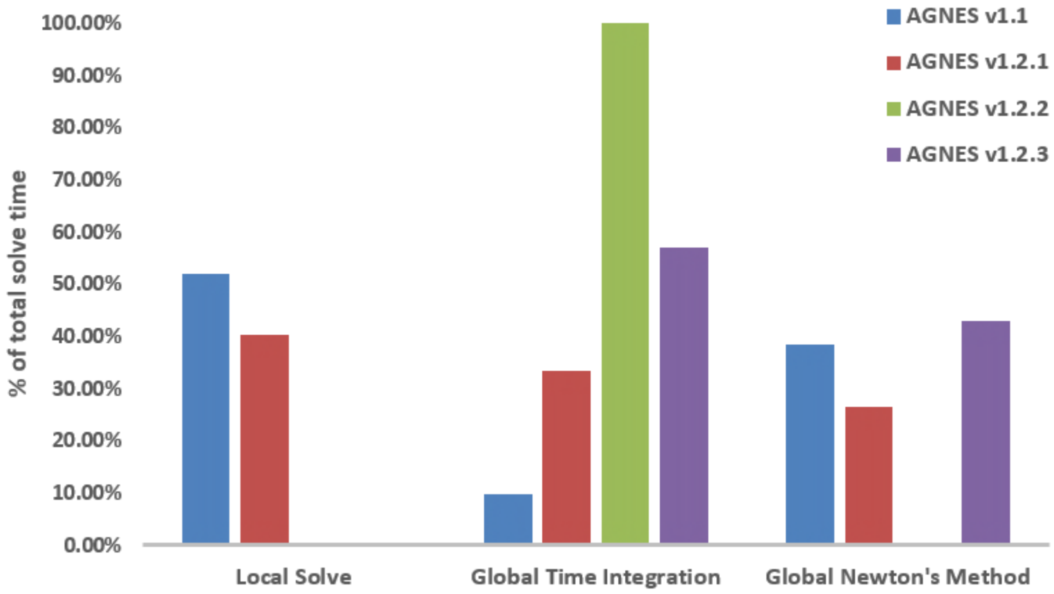

Figure 4.50: Relative importance of three resolution methods in terms of time for different solver implementations

The first improved solver is AGNES v1.2.1 which use similar resolution methods as KPPSMOKE with an identical approach for switching between these methods. Although, these time percentages can be altered by changing the tolerances for each method along with tolerance for the switch criteria, it is quite clear that AGNES v1.2.1 has a much balanced time breakdown. Sequential resolution using local solver is important but it now consumes less of the total computational time. The role of the sequential approach is passed down to the global time integration that ensures a stable approach of the network towards the final solution. The final stage of the solving process is then carried out by the Newton's method. It offers fast convergence but now without any risk of divergence. Overall, this solver implementation is nicely balanced but further improvements are required to get rid of localized resolution. The second version AGNES v1.2.2 uses only global time integration without any need of sequential local resolution or global Newton's advancement. Although, this version offers less programming complexity, it suffers due to its extremely slow convergence due to lack of few iterations of Newton's advancement.

One important observation is made from the previous version: using adaptive time stepping capabilities of PETSc, an alternative to local solver can be designed with the sole purpose of advancing the global system close to the final solution without worrying about the accuracy. This is followed by global Newton's method which will boost the convergence process and in case of divergence, the solving process will move back to the time integration. This version, AGNES v1.2.3, has no local solve requirement and the total solving time is split almost equally between both global resolution methods.

RESIDUAL BEHAVIOUR

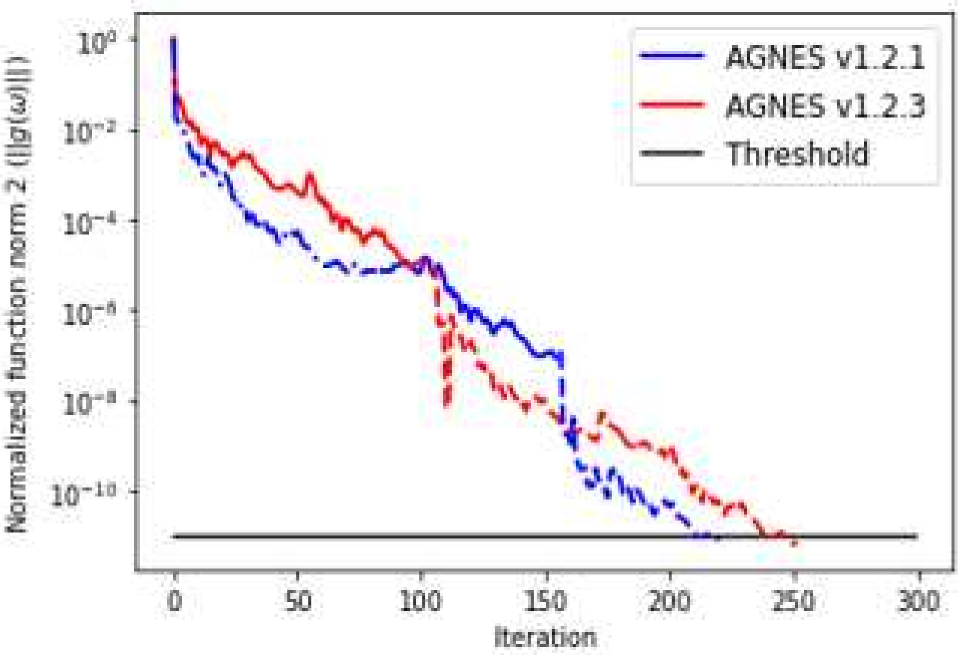

Figure 4.51: Residual function norm 2 trends for AGNES v1.2.1 and AGNES v1.2.3, normalized with respect to the initial value. Dash dot line: resolution through the sequential approach (Cantera based local solver). Solid line: global resolution using time integration. Dashed lines: resolution using globalised non-linear solver

Figure 4.51 shows the variation in function residual (norm 2) with the number of iterations, after normalizing with respect to the initial value, as calculated before starting the solution process. Both AGNES v1.2.1 and v1.2.3 are compared by breaking down the entire solving process based on used resolution methods. For AGNES v1.2.1, this process begins with sequential resolution of the reactor network. As this is based on KPPSMOKE approach [5], local solver is essential in the beginning and later loses its effectiveness over the iterations. After achieving sufficient network resolution, the solution process is handled by time integration for global resolution and as seen from the above figure, it noticeably speeds up the convergence rate. However, using only time integration, it would require more computational steps to reach the final solution. Hence, based on certain convergence threshold, as discussed in Section 2.5.2, the final steps of the solution is done using a globalised Newton's method. This ensures faster convergence and the residuals drop significantly and quickly till the convergence limit.

However, AGNES v1.2.3 uses only global resolution methods which includes an inaccurate time integration (less stringent atol and rtol) as a replacement for sequential local resolution and globalised Newton's method for approaching the final solution. As seen in Figure 4.51, solution process is initialized using time integration which is inaccurate but computationally fast. Within few iterations, the achieved convergence becomes sufficient for the globalised Newton's method to control the solving process and reach the solution. However, in case of divergence, the solver again calls time integration but with slightly reduced atol and rtol (by factor of 10). This process is repeated till the solution process is completed.

AGNES v1.2.3 requires slightly more computational steps and hence longer solution time as compared to AGNES v1.2.1. But this difference might change with increase in reactor count in the network due to limitations of the sequential resolution approach. This approach is needed to be improved further to reduce the computational time by investigating better alternatives for inaccurate time integration.

5

CONCLUSION AND RECOMMENDATIONS

5.1. CONCLUSION

The main objective of this research project was to improve AGNES based on latest developments in the field of CFD-CRN. AGNES CFD-CRN [6][17] approach splits the emission prediction problem into three important steps. First, the flowfield inside the combustion system is modelled using CFD simulation with simplified chemistry. Second, a chemical reactor network is constructed by grouping similar mesh cells by traversing a graph containing all CFD simulation data. Finally, the CRN is solved with detailed chemistry to generate emission predictions. During the literature survey, a research gap was found regarding the choice of resolution methods used in common CFD-CRN software such as KPPSMOKE [5]. It was recommended by Cuoci et al. (2007,2013) [11][4] and Stagni et al. [5] to use some kind of local resolution aiming to resolve every reactor sequentially before even trying a global resolution. Motivation for this approach was to ensure the global Newton's method that offers fast convergence but is quite unstable and prone of divergence if the initial guess is not close to the final solution. However, no explanation was given for why a global time integration cannot be used to achieve similar results as the sequential approach but even faster due to time marching of the entire network. Therefore, this thesis is focused on improving AGNES CRN solver using an optimised combination of different resolution methods, mainly global time integration and Newton's method. This led to the formation of the research questions for this project.

1. **How are the performance parameters, such as computational time and**

convergence rate, affected by implementing a PETSc based solver for hybrid CFD-CRN method?

2. **Does the choice of the time-stepping or nonlinear or linear scheme have any impact on the convergence rate of the CFD-CRN solver?**

The above questions were opted to determine if a global resolution approach without any local solver is feasible at an acceptable computational cost. It was hypothesized that reducing or even removing the local solver or any kind of sequential resolution approach could improve the solution time but at an added risk of divergence. However, spending additional time with global time integration could help the final approach towards the final solution using Newton's method more stable and ensure convergence at any cost. For this project, three solver implementations were tested (detailed descriptions are presented in Section 2.5).

1. AGNES v1.2.1: It is based on Stagni et al. KPPSMOKE approach [5]. This approach utilises Cantera based local solver for sequential resolution of the network, after achieving local convergence global time integration and final solution approach using global Newton's method. Both global methods are based on PETSc library.

2. AGNES v1.2.2: This implementation is a purely global time integration based approach. This was done to test PETSc adaptive time-stepping and convergence rate based on several tolerance values.

3. AGNES v1.2.3: This is a new approach which includes an inaccurate time integration coupled with global Newton's method. Motivation for an inaccurate time integration is to not worry about accuracy but to just push the system close enough to the final solution such that global Newton's method does not face any convergence issue. This method proved to be quite promising in solving large reactor systems but suffered at lower reactor count.

Based on the results discussed in Chapter 4, it was observed that both pre-research AGNES v1.1 and current AGNES v1.2 had similar trends and integral average for all important species including NO and CO. Unfortunately, no improvement in accuracy was seen due to lack of modelled diffusion and temperature fluctuations. It can be concluded that no accuracy was lost due to implementing a new solver approach.

As hypothesized, reducing or removing dependency on local resolution approach proved to improve the solution time. Use of only time integration proved to be quite computationally expensive but opting for an inaccurate time integration with less stringent tolerances resolved the system to a state from which Newton's method could be applied without any risk of divergence. In conclusion, both AGNES v1.2.1 and AGNES v1.2.3 proved to be around 13% faster than pre-research AGNES v1.1. Furthermore, with the help of an improved Jacobian evaluation strategy, where sparse matrix containing only transport contributions was calculated only once and later added to reaction contributions during every Jacobian evaluations, computational time was reduced by 0.7-1 second per Jacobian evaluation. Overall, the achieved speedup is 1.21 i.e a 21% improvement in the solution speed. However, further improvement can be made to improve the time integration using better choice for linear solver and its preconditioner.

In conclusion, the choice for PETSc based solver implementation had a significant impact on AGNES performance. There were no compromises made in terms of accuracy while also achieving similar results as pre-research AGNES version at sufficiently lower computational time. Also, the correct choice of solving schemes and preconditioners proved to be an important factor at achieving faster convergence. However, it is recommended to further explore the capabilities of PETSc and its varied functionalities by re-creating the solver in C language. This will allow for unrestricted access to all latest developments and features in PETSc.

5.2. AGNES IMPROVEMENT RECOMMENDATION

Based on the current AGNES state, few recommendations are made for future improvement of AGNES which can potentially be future MSc thesis topics.

1. **Mass fraction inconsistency correction**: In the current version of AGNES, species mass fraction consistency check is done using Cantera which check if all mass fractions are between 0 to 1 and their sum is equal to 1 inside each reactor. However, this problem can be resolved by selecting an inert species like Argon from the reaction mechanism GRi-Mech 3.0. This approach will lead to reduced number of equations as only $N_S - 1$ species equations are needed to be solved, which speeds up the solving process slightly. Furthermore, mass fraction of Argon will be calculated by subtracting sum of other species mass fraction from 1. This automatically solves the issue of inconsistency. However, this hypothesis is needed to be verified.

2. **Parallel architecture**: The current version of AGNES solves the CRN in sequential manner which leads to high computation time. Cuoci et al. [4] recommended a parallel architecture for KPP algorithm which led to the

development of KPPSMOKE application by Stagni et al. [5] which utilizes faster parallel computation. Parallel computing allows more work to be done at the same time by efficiently utilising the available computational resources. Not all processes can be parallelised but numerical processes such as Jacobian evaluation and solution of reactors especially the linear solves can be done parallel manner. Application of such parallel architecture for solving CRNs in AGNES could lead to improved performance and low computation time. However, if PETSc is selected as the way forward, this requires solver implementation in C or C++ language as more complex functionalities of PETSc are only available in these languages like DMNetwork [29].

3. **Tool flexibility and robustness**: As of now, AGNES is only tested with Sandia Flame D [2] and Verissimo et al. [39] flameless combustion test case. There is lot of work to be done to make AGNES flexible and robust so that any CFD simulation for any geometry can be studied using the tool.
4. **Turbulence**: In the current version of AGNES, temperature fluctuations due to turbulence are not considered. Nevertheless, Cuoci et al. (2013) [4] concluded that use of temperature fluctuations in combustion modelling led to significant improvements in emission predictions. Thus, it is suggested to verify the impact of modelled turbulent fluctuation on the accuracy of emission predictions using different test cases.
5. **Liquid fuels**: Currently, AGNES is only tested with gaseous fuel [17]. However, the use of liquid fuels such as kerosene and gasoline is possible with need minor adjustments like mass-flow corrections accounting for droplet evaporation and inclusion of liquid phase chemistry.
6. **Zoned/Scaled clustering**: Current version of AGNES uses only one clustering tolerance value for the entire domain. This often results in either too high resolution at less interesting locations, poor resolution at important locations or a combination of both. Although, Wit [17] solved this issue by exploring the concept of zoned clustering as proposed by Monaghan et al. [9]. However, further exploration in zoned/ scaled clustering is required for better CRN construction and accuracy.
7. **Complex geometries**: Till date, AGNES is not tried with complex geometries, which could prove to be challenging due to AGNES being designed to handle only a single Fluent fluid zone [6]. It is suggested to enable multiple zones to allow modelling of complex combustion systems.

BIBLIOGRAPHY

[1] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dmitry Karpeyev, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. `https://www.mcs.anl.gov/petsc`, 2019.

[2] Sandia National Laboratories. Sandia / TUD piloted CH_4 / air jet flames. `https://www.sandia.gov/TNF/DataArch/FlameD.html`.

[3] D. Benedetto, S. Pasini, M. Falcitelli, C. La Marca, and L. Tognotti. NO_x emission prediction from 3-D complete modelling to reactor network analysis. *Combustion Science and Technology*, 153:279 – 294, 2000.

[4] Alberto Cuoci, Alessio Frassoldati, Alessandro Stagni, T. Faravelli, Eliseo Ranzi, and G. Buzzi-Ferraris. Numerical modeling of NO_x formation in turbulent flames using a kinetic post-processing technique. *Energy Fuels*, 27:1104–1122, 01 2013.

[5] A. Stagni, A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi. A fully coupled, parallel approach for the post-processing of CFD data through reactor network analysis. *Computers Chemical Engineering*, 60:197 – 212, 2014.

[6] Rishikesh Sampat. Automatic generation of chemical reactor networks for combustion simulations, MSc Thesis, TU Delft (2018). `http://resolver.tudelft.nl/uuid:5297cc7d-cf15-4fc4-9ba4-bce2290cc85c`.

[7] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, William D. Gropp, Dmitry Karpeyev, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.13, Argonne National Laboratory, 2020.

[8] R. Barlow and J. Frank. Piloted CH_4/air flames C, D, E, and F (release 2.1). https://www.sandia.gov/TNF/DataArch/FlameD/SandiaPilotDoc21.pdf.

[9] Rory Monaghan, Råbi Tahir, Alberto Cuoci, Gilles Bourque, Marc Füri, Robert Gordon, Tiziano Faravelli, Alessio Frassoldati, and H.J. Curran. Detailed multi-dimensional study of pollutant formation in a methane diffusion flame. *Energy Fuels*, 26, 04 2012.

[10] T. Faravelli, L. Bua, A. Frassoldati, A Antifora, L. Tognotti, and E. Ranzi. A new procedure for predicting NO_x emissions from furnaces. *Computers & Chemical Engineering*, 25(4):613 – 618, 2001.

[11] Alberto Cuoci, Alessio Frassoldati, G. Ferraris, T Faravelli, and Eliseo Ranzi. The ignition, combustion and flame structure of carbon monoxide/hydrogen mixtures. note 2: Fluid dynamics and kinetic aspects of syngas combustion. *International Journal of Hydrogen Energy*, 32:3486–3500, 10 2007.

[12] Michael Frenklach Nigel W. Moriarty Boris Eiteneer Mikhail Goldenberg C. Thomas Bowman Ronald K. Hanson Soonho Song William C. Gardiner Jr. Vitali V. Lissianski Gregory P. Smith, David M. Golden and Zhiwei Qin. GRI-Mech 3.0. http://www.me.berkeley.edu/gri_mech/.

[13] Craig T. Bowman. Kinetics of pollutant formation and destruction in combustion. *Progress in Energy and Combustion Science*, 1(1):33–45, 1975.

[14] James A. Miller and Craig T. Bowman. Mechanism and modeling of nitrogen chemistry in combustion. *Progress in Energy and Combustion Science*, 15(4):287–338, 1989.

[15] SANJAY M. CORREA. A review of nox formation under gas-turbine combustion conditions. *Combustion Science and Technology*, 87(1-6):329–362, 1993.

[16] Sajjad Yousefian, Gilles Bourque, and Rory Monaghan. Review of hybrid emissions prediction tools and uncertainty quantification methods for gas turbine combustion systems. page V04BT04A005, 06 2017.

[17] Maaike de Wit. The effect of solving the energy equation on combustion simulation using automatically generated chemical reactor networks, MSc Thesis, TU Delft (2019). http://resolver.tudelft.nl/uuid:2c37aa35-f379-4d2a-8d68-a86c76a630c9.

[18] K. Ehrhardt et al. Modeling of NO_x reburning in a pilot scale furnace using detailed reaction kinetics, Combustion Science and Technology 131 (1998).

[19] M. Falcitelli, L. Tognotti, and S. Pasini. An algorithm for extracting chemical reactor network models from CFD simulation of industrial combustion systems. *Combustion Science and Technology*, 174(11-12):27–42, 2002.

[20] A. Frassoldati, S. Frigerio, E. Colombo, F. Inzoli, and T. Faravelli. Determination of NO_x emissions from strong swirling confined flames with an integrated CFD-based procedure. *Chemical Engineering Science*, 60(11):2851–2869, 2005.

[21] T. Nilsson. Development of a simulation methodology for gas turbine combustion, MSc thesis, lund university (2014).

[22] André A. V. Perpignan, Rishikesh Sampat, and Arvind Gangoli Rao. Modeling pollutant emissions of flameless combustion with a joint CFD and chemical reactor network approach. *Frontiers in Mechanical Engineering*, 5:63, 2019.

[23] David G. Goodwin, Raymond L. Speth, Harry K. Moffat, and Bryan W. Weber. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. `https://www.cantera.org`, 2021. Version 2.5.1.

[24] Alan Hindmarsh, Peter Brown, Keith Eric Grant, Steven Lee, Radu Serban, Dan Shumaker, and Carol Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31:363–396, 11 2004.

[25] The Trilinos Project Team. *The Trilinos Project Website*.

[26] MPI: A Message-Passing Interface Standard Version 3.1. *Message Passing Interface Forum*, June 4, 2015.

[27] Scott D. Cohen, Alan C. Hindmarsh, and Paul F. Dubois. Cvode, a stiff/nonstiff ode solver in c. *Computers in Physics*, 10(2):138–143, 1996.

[28] Lisandro D. Dalcin, Rodrigo R. Paz, Pablo A. Kler, and Alejandro Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011. New Computational Methods and Software Tools.

[29] Shrirang Abhyankar, Getnet Betrie, Daniel Adrian Maldonado, Lois C. Mcinnes, Barry Smith, and Hong Zhang. PETSc DMNetwork: A library for scalable network PDE-based multiphysics simulations. *ACM Trans. Math. Softw.*, 46(1), April 2020.

[30] Shrirang Abhyankar, Jed Brown, Emil M Constantinescu, Debojyoti Ghosh, Barry F Smith, and Hong Zhang. PETSc/TS: A modern scalable ODE/DAE solver library. *arXiv preprint arXiv:1806.01437*, 2018.

[31] Andrew Christieb, Sigal Gottlieb, Zachary Grant, and David Seal. Explicit strong stability preserving multistage two-derivative time-stepping schemes. *Journal of Scientific Computing*, 68, 04 2015.

[32] Kenneth Jansen, Christian Whiting, and Gregory Hulbert. Generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190, 11 2000.

[33] Rory Monaghan, Råbi Tahir, Alberto Cuoci, Gilles Bourque, Marc Füri, Robert Gordon, Tiziano Faravelli, Alessio Frassoldati, and H.J. Curran. Detailed multi-dimensional study of pollutant formation in a methane diffusion flame. *Energy Fuels*, 26, 04 2012.

[34] Rory Monaghan, Råbi Tahir, Gilles Bourque, Robert Gordon, Alberto Cuoci, Tiziano Faravelli, Alessio Frassoldati, and H.J. Curran. Detailed emissions prediction for a turbulent swirling nonpremixed flame. *Energy Fuels*, In press, 02 2014.

[35] Ali Habibi, Bart Merci, and D. Roekaerts. Turbulence radiation interaction in Reynolds-averaged Navier-Stokes simulations of nonpremixed piloted turbulent laboratory-scale flames. *Combustion and Flame*, 151:303–320, 10 2007.

[36] Bart Merci, Erik Dick, Jan Vierendeels, D. Roekaerts, and Tim Peeters. Application of a new cubic turbulence model to piloted and bluff-body diffusion flames. *Combustion and Flame*, 126:1533–1556, 07 2001.

[37] Jeroen van Oijen and Philip Goey. Modelling of premixed laminar flames using flamelet-generated manifolds. *Combustion Science and Technology - COMBUST SCI TECHNOL*, 161:113–137, 12 2000.

[38] Warren D. Seider and Charles W. White III. Chemical reaction equilibrium analysis: Theory and algorithms by william r. smith and ronald w. missen, 364 pp., john wiley, 1983. *AIChE Journal*, 31(1):176–176, 1985.

[39] A. S. Veríssimo, A. M. A. Rocha, and M. Costa. Operational, combustion, and emission characteristics of a small-scale combustor. *Energy & Fuels*, 25(6):2469–2480, 2011.