Delft University of Technology

Document Version
Final published version

Citation (APA)
El Abbassi, M. (2026). *Mathematical Modelling of Turbulent Combustion and Conjugate Heat Transfer in Rotary Kilns*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:89a50b29-bc28-4e3f-8b67-2dc9436bbf88

Mathematical Modelling of Turbulent Combustion and Conjugate Heat Transfer in Rotary Kilns

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathrm{U} = 0$$

$$\rho \left(\frac{\partial \mathrm{U}}{\partial t} + \mathrm{U} \cdot \nabla \mathrm{U} \right) = -\nabla p + \nabla \cdot \mu \nabla \mathrm{U}$$

$$\frac{\partial (\rho Y_\theta)}{\partial t} + \nabla \cdot (\rho \mathrm{U} Y_\theta) = \nabla \cdot \rho \Gamma \nabla Y_\theta + R_\theta$$

$$\frac{\partial (\rho h_s)}{\partial t} + \nabla \cdot (\rho \mathrm{U} h_s) = \frac{Dp}{Dt} + \nabla \cdot \rho \alpha \nabla h_s + Q_c + Q_r$$

$$-\nabla \cdot \left(\frac{1}{3\kappa} \nabla G \right) = \kappa (4 \sigma T^4 - G)$$

$$\frac{\partial (\bar{\rho} h)}{\partial t} = \nabla \cdot (\lambda \nabla T)$$

Mohamed el Abbassi

MATHEMATICAL MODELLING OF TURBULENT COMBUSTION AND CONJUGATE HEAT TRANSFER IN ROTARY KILNS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. H. Bijl,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 3 februari 2026 om 10:00 uur

door

Mohamed EL ABBASSI

Dit proefschrift is goedgekeurd door de

promotor: prof.dr.ir. C. Vuik,
promotor: prof.dr. H.M. Schuttelaars,
copromotor: dr. D.J.P. Lahaye

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. C. Vuik,	Technische Universiteit Delft
Prof.dr. H.M. Schuttelaars,	Technische Universiteit Delft
Dr. D.J.P. Lahaye,	Technische Universiteit Delft
Onafhankelijke leden:	
Prof.dr.ir. N.G. Deen,	Technische Universiteit Eindhoven
Prof.dr.ir. M.B. van Gijzen,	Technische Universiteit Delft
Dr.ir. G.H. Keetels,	Technische Universiteit Delft
Em.prof.dr. D.J.E.M Roekaerts,	Technische Universiteit Delft
Dr.ir. M.I. Gerritsma	Technische Universiteit Delft, *reservelid*

This research was sponsored by Almatis B.V.

Keywords: Non-premixed combustion, Conjugate heat transfer, OpenFOAM, Iterative methods, Rotary kiln, Industrial furnace, NO_x formation.

ISBN 978-94-6384-903-6

Printed by: www.proefschriftmaken.nl
An electronic version of this dissertation is available at http://repository.tudelft.nl/

Take as small steps as you can,
but don’t stop.

Kees Vuik

SUMMARY

This dissertation presents the development, validation, and application of an integrated Computational Fluid Dynamics (CFD) solver, which is built upon the open-source OpenFOAM framework, to predict and ultimately reduce thermal nitric oxide (NO) formation in industrial rotary kilns without sacrificing process productivity. The work is structured into three parts: theoretical foundations, solver implementation and validation through increasingly complex case studies, and final conclusions and recommendations.

The theoretical part of the dissertation established a comprehensive mathematical framework for simulating the complex interactions within rotary kilns, focusing on turbulent combustion, conjugate heat transfer (CHT), and thermal NO formation. Each part of this multi-physics problem is highlighted with a discussion of one or more available models to resolve it, which is often a trade-off between computational speed and accuracy.

The numerical treatment employs the Finite Volume Method for spatial discretization and the PISO/SIMPLE families for pressure–velocity coupling. A systematic study of linear solvers revealed that, for non-reacting flows on a 2.3 million-cell kiln mesh, a Conjugate Gradient (CG) solver with the Generalized Geometric Algebraic Multigrid (GAMG) as a preconditioner accelerates convergence by up to 7× compared to basic solvers. In reactive simulations, however, the chemistry solver becomes the dominant cost, underscoring the necessity of chemistry-acceleration techniques. Chemical kinetics are introduced via both global (single-step and two-step) and detailed (e.g., GRI-3.0) methane/air mechanisms. Due to the stiffness and computational cost of detailed mechanisms, the Tabulation of Dynamic Adaptive Chemistry (TDAC) method was implemented, achieving speed-ups of approximately 160× for the GRI-3.0 scheme with minimal impact on NO predictions.

At the heart of the solver lie the Reynolds-Averaged Navier–Stokes (RANS) equations, which are augmented with transport equations for turbulence, chemical species, and energy (of both fluid and solid). The solver couples both the standard combustion and CHT solvers of OpenFOAM. Thermal NO formation is computed in the post-processing stage based on the extended Zeldovich mechanism, which uses the solved temperature and radical fields to evaluate NO production efficiently.

Chapter 4 validates the coupled combustion–CHT solver against the Sandia Flame D benchmark, supplemented by a cylindrical refractory wall. The results of the new solver, with conjugate heat transfer turned off, are identical to the standard combustion solver of OpenFOAM, and good agreement is shown with experiments when using the full GRI mechanism. With conjugate heat transfer switched on, good qualitative and quantitative agreements are shown with the results generated by ANSYS Fluent. In Chapter 5, the impact of CHT and external radiative heat loss is assessed in an axisymmetric kiln model at thermal inputs of 10 MW and 40 MW. Comparisons with fixed-wall-temperature studies showed that neglecting CHT can misrepresent temperature distributions by several hundred Kelvin, highlighting the critical role of CHT modeling in rotary-kiln simulations.

Chapter 6 applies the solver to a full-scale 3D kiln at Almatis B.V., exploring three strategies for NO mitigation. The first strategy modifies the secondary air-inlet geometry by replacing a rectangular inlet with an annular one, which produces a more uniform recirculation zone, reduces local hot spots, and lowers thermal NO formation by up to 35 percent. The second strategy varies the air–fuel ratio (AFR) from fuel-rich to fuel-lean conditions. As the higher-Reynolds-number runs exhibited convergence challenges due to vortex-stretching instabilities, this study continued in a simplified 2D kiln. The study showed that the lean conditions significantly reduce both temperature and NO formation, at the potential expense of clinker sintering. However, experimental tests on the actual kiln revealed that NO formation increases with AFR. The third strategy introduces exhaust gas recirculation (EGR), again in 2D; up to 7 percent EGR attenuates peak flame temperatures, and the increased radiative emissivity of the recirculated flue gases maintains wall heating while achieving thermal NO reductions up to 99 percent. Of the strategies tested, EGR emerges as the most effective NO-control measure that preserves kiln wall heating, positioning it as a strong candidate for industrial deployment.

Limitations of this work include the absence of full detailed chemistry in 3D simulations at higher flow rates than the benchmark case, omission of kiln rotation and granular bed dynamics, and reliance on steady RANS. Future developments should extend to running unsteady RANS simulations in 3D space to capture unsteady phenomena such as vortex-stretching and flame lift-off, and implement state-of-the-art detailed combustion models and accelerators. Furthermore, by integrating kiln rotation and granular bed dynamics via Discrete Element Method (DEM) coupling will also account for clinker quality and even more accurate flow predictions, which will support the advancement of more sustainable and efficient rotary-kiln operations. With the currently available computational resources and the new modular-based architecture of OpenFOAM, the desired outcome is closer than ever.

SAMENVATTING

Dit proefschrift presenteert de ontwikkeling, validatie en toepassing van een geïntegreerde oplosser voor numerieke stromingsleer (Computational Fluid Dynamics, CFD), gebaseerd op het open-source framework van OpenFOAM. De oplosser is ontwikkeld om de vorming van thermisch stikstofoxide (NO) in industriële draaiovens te voorspellen en uiteindelijk te reduceren, zonder dat dit ten koste gaat van de procesproductiviteit. Het werk is opgebouwd uit drie delen: de theoretische grondslagen, de implementatie en validatie van de oplosser aan de hand van steeds complexere casestudies, en ten slotte de conclusies en aanbevelingen.

Het theoretische deel van dit proefschrift beschrijft een uitgebreid wiskundig raamwerk voor het simuleren van de complexe interacties in draaiovens, met de nadruk op turbulente verbranding, gekoppelde warmteoverdracht (Conjugate Heat Transfer, CHT) en thermische NO-vorming. Elk onderdeel van dit multifysische probleem wordt behandeld aan de hand van één of meerdere beschikbare modellen, waarbij doorgaans een afweging moet worden gemaakt tussen rekensnelheid en nauwkeurigheid.

Voor de numerieke aanpak wordt de eindige volumemethode (Finite Volume Method, FVM) toegepast voor de ruimtelijke discretisatie, en worden de PISO- en SIMPLE-algoritmen gebruikt voor de koppeling tussen druk en snelheid. Een systematische studie van lineaire oplossers liet zien dat, voor niet-reagerende stromingen op een rekenrooster met 2,3 miljoen cellen, een Conjugate Gradient (CG)-oplosser met Generalized Algebraic Multigrid (GAMG) als preconditioner de convergentie tot zeven keer versnelt ten opzichte van standaardoplossers. Bij reactieve simulaties daarentegen vormen de chemische berekeningen de grootste rekenkundige belasting, wat de noodzaak onderstreept van versnellingsmethoden voor chemie. Chemische kinetiek is gemodelleerd met zowel globale mechanismen (éénstaps- en tweestaps) als gedetailleerde mechanismen (bijv. GRI-3.0 voor methaan/lucht). Vanwege de stijfheid en hoge rekenkosten van gedetailleerde mechanismen is de Tabulation of Dynamic Adaptive Chemistry (TDAC)-methode geïmplementeerd. Hiermee werd voor het GRI-3.0-mechanisme een snelheidswinst van circa 160 keer bereikt, met minimale invloed op de NO-voorspellingen.

Centraal in de oplosser staan de Reynolds-gemiddelde Navier–Stokes (RANS)-vergelijkingen, uitgebreid met transportvergelijkingen voor turbulentie, chemische soorten en energie (zowel in de fluïde- als vaste fase). De oplosser koppelt de standaard verbrandings -en CHT-oplossers van OpenFOAM. De thermische NO-vorming wordt in de naverwerking berekend op basis van het uitgebreide Zeldovich-mechanisme, waarbij de opgeloste temperatuur- en radicaalvelden worden gebruikt om de NO-productie efficiënt te evalueren.

Hoofdstuk 4 valideert de gekoppelde verbrandings–CHT-oplosser met de Sandia-vlam D-benchmark, uitgebreid met een cilindrische vuurvaste wand. De resultaten van de nieuwe oplosser, met CHT uitgeschakeld, zijn identiek aan die van de standaard verbrandingsoplosser van OpenFOAM, en bij gebruik van het volledige GRI-mechanisme wordt goede overeenstemming bereikt met experimentele data. Met CHT ingeschakeld komen de resultaten zowel kwalitatief als kwantitatief goed overeen met die van ANSYS Fluent. In Hoofdstuk 5 wordt de invloed van CHT en externe stralingsverliezen onderzocht in een axiaalsymmetrisch ovenmodel bij thermische vermogens van 10 MW en 40 MW. Vergelijkingen met simulaties met een vaste wandtemperatuur tonen aan dat het verwaarlozen van CHT kan leiden tot temperatuursverschillen van enkele honderden Kelvin, wat de cruciale rol van CHT-modellering in simulaties van draaiovens benadrukt.

Hoofdstuk 6 past de oplosser toe op een industriële driedimensionale draaioven bij Almatis B.V., waarbij drie strategieën voor NO-reductie worden onderzocht. De eerste strategie verandert de geometrie van de secundaire luchtinlaat door een rechthoekige opening te vervangen door een ringvormige. Dit creëert een meer uniforme recirculatiezone, vermindert lokale hotspots en verlaagt de thermische NO-vorming met maximaal 35 procent. De tweede strategie onderzoekt variaties in de lucht–brandstofverhouding (Air–Fuel Ratio, AFR), van brandstofrijk naar brandstofarm. Omdat simulaties met hogere Reynoldsgetallen last hadden van convergentieproblemen door vortex-stretching-instabiliteiten, is deze studie voortgezet in een vereenvoudigd tweedimensionaal ovenmodel. De resultaten tonen aan dat brandstofarme condities zowel de temperatuur als de NO-vorming aanzienlijk reduceren, zij het met een mogelijk negatieve invloed op het sinteren van klinker. Experimentele metingen in de werkelijke oven lieten echter zien dat de NO-vorming toeneemt met de AFR. De derde strategie introduceert rookgasrecirculatie (Exhaust Gas Recirculation, EGR), eveneens in 2D. Tot 7 procent EGR verlaagt de piekvlamtemperaturen, terwijl de verhoogde stralingsemissiviteit van de gerecirculeerde rookgassen de wandverwarming behoudt en thermische NO-reducties tot 99 procent mogelijk maakt. Van de onderzochte strategieën blijkt EGR de meest effectieve maatregel voor NO-beheersing, met behoud van wandverwarming, waardoor het een veelbelovende optie is voor industriële toepassing.

De beperkingen van dit onderzoek omvatten het ontbreken van volledig gedetailleerde chemie in 3D-simulaties bij hogere debieten dan de benchmark, het niet meenemen van ovenrotatie en de dynamica van het granulair bed, en de afhankelijkheid van stationaire RANS-berekeningen. Toekomstig onderzoek zou zich moeten richten op het uitvoeren van tijdsafhankelijke RANS-simulaties in 3D om instationaire fenomenen zoals vortex-stretching en vlamafsplitsing vast te leggen, evenals de implementatie van geavanceerde verbrandingsmodellen en chemieversnellers. Bovendien zal het integreren van ovenrotatie en granulair beddynamica via koppeling met de Discrete Element Method (DEM) bijdragen aan een betere voorspelling van klinkerkwaliteit en nauwkeuriger stromingsgedrag, waarmee duurzamere en efficiëntere ovenprocessen ondersteund worden. Met de huidige rekenkracht en de modulaire architectuur van OpenFOAM komt dit doel dichterbij dan ooit.

Contents

1

INTRODUCTION

1.1. ROTARY KILN AND ITS APPLICATIONS

Rotary kilns are long rotating cylindrical furnaces that are slightly inclined horizontally, with the burner located at the lower end, as it is shown in Fig. 1.1. The raw (mostly solid) mixture enters from the higher end of the kiln and gradually moves towards the burner due to the inclination and rotation. Rotary kilns can handle different kinds of feedstock, from slumped and granular solids to wet slurry. Therefore, its applications in the industry are diverse, ranging from drying and calcination to metallurgical processing [1].

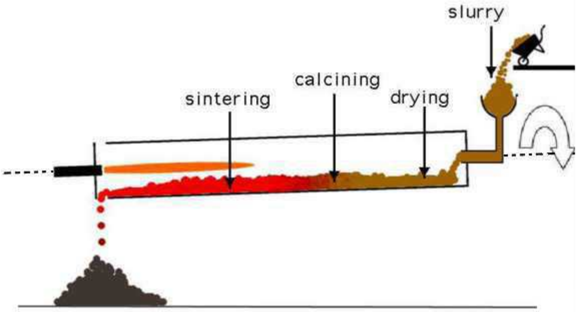

Figure 1.1: Cross section of a rotary kiln. [2]

The cement industry is the major material processing industry that makes use of the kiln. During the process, the raw mixture of limestones gradually moves down towards the lower end while being stirred due to the kiln's rotation. At the lower end, fuel in the form of gas, oil, or pulverised solid fuel is blown in through the burner pipe, producing a large concentric flame in the lower part of the kiln tube. Due to heat provided by the flame, the temperature of the incoming raw mixture gradually increases, and depending on the length of the kiln and preprocessing of the raw mix, it may go through different stages of chemical reactions. Under the assumption that no chemical preprocessing occurs, the first stage is the drying process in which the water in the slurry evaporates. This is followed by the calcination of the ores, in which, e.g., calcium carbonate (limestone) decomposes to calcium oxide and carbon dioxide. In the final stage, the temperature is high enough for partial melt to occur, whereby different oxides fuse together. This process is known as 'sintering'.

The kiln is considered the heart of the cement manufacturing process. However, it is also the main energy-consuming, greenhouse gas- and pollutant-emitting stage. Cement is the source of about 8% of the world's carbon dioxide emissions [3]. Roughly half of the total carbon dioxide is formed by the combustion process. The other half is due to calcination. Therefore, improvement of kiln efficiency and mitigation of pollutant formation have been the central concern of cement manufacturing technology.

1.2. POLLUTANT FORMATION AND PRODUCTION CONCERNS

Carbon dioxide and water are the products formed at ideal combustion of hydrocarbon fuels and can only be reduced by improving the efficiency of the kiln. However, combustion is never ideal, and undesired byproducts are formed as well, such as carbon monoxide (CO), nitric oxide (NO), nitrogen dioxide (NO_2), sulphur dioxide (SO_2), unburned hydrocarbons (UHC) and soot particles. These are called pollutants due to their harmful effects on the environment. Pollutant formation can be reduced by improving the combustion technology.

SO_2 is a byproduct from fuels that contain sulphur. CO, UHC and soot are produced by intermediate reactions that are killed due to quenching or lack of oxygen.

Most of the NO oxidises further to NO_2, and are collectively known as NO_x. NO primarily results from two sources [4]

- Molecular N_2 contained in air,
- Fuel-bound nitrogen,

of which the latter is usually present in fuels such as heavy oil and coal. It is a major contributor of photochemical smog and ozone in the urban area [5]. NO_x is formed either at high temperatures where N_2 reacts with different radicals or as a combustion product of nitrogen-bound fuel. NO is far more important than NO_2 in combustion processes because NO_2 quickly reacts with O or H radicals to form

NO, unless this reaction is quenched. Due to the significantly high operating temperatures in the kiln, $NO_x$are the major concerning pollutants that are being formed. With more stringent global NO_x emission standards, predicting NO_x formation in rotary kilns has become a priority. By 2050, the European Union expects that NO_x emissions in the cement industry will reduce by 40% as compared to 2020 levels [6].

Another important concern of the cement producer is how the combustion and resulting heat transfer influence the sintering of the limestones. As thermal NO mitigation techniques usually aim at lowering the flame's temperature, it is important to study the consequence of the resulting heat release.

1.3. PRESENT CONTRIBUTIONS IN CFD MODELLING

Due to the extreme temperatures and the rotation of the kiln, it is nearly impossible to conduct measurements inside the kiln. The (remote) measurement devices will be destroyed either by the flame or by the sloshing clinkers. Moreover, the thick refractory lining and steel shell provide no optical access. Due to these difficulties, rotary kilns undoubtedly benefit from the use of Computational Fluid Dynamics (CFD) as a predictive tool, which is a more cost-efficient and relatively safer procedure compared to small-scale laboratory experiments. However, accurate predictions by computational models are not always guaranteed. More detailed geometries and computational models will lead to more accurate predictions, but they are usually less robust, and they always come with a higher computational cost. Hence, the level of detail of the applied models is bound with the sheer size of the kiln and the complexity of the occurring physical phenomena. Modelling simplifications are therefore unavoidable.

Similar challenges arise in other high-temperature industrial furnaces, such as steam cracking furnaces and glass melting furnaces. These systems also operate at extreme temperatures, involve turbulent combustion, radiative heat transfer, and conjugate heat transfer through refractory linings, and suffer from limited experimental accessibility. CFD has therefore been extensively applied in these fields to study combustion behaviour, pollutant formation, and heat transfer performance. Studies on steam cracking furnaces have focused on turbulent combustion, NO_x formation, and burner configuration effects [7, 8], while glass furnace simulations have demonstrated the importance of coupling combustion processes with heat transfer and material flow [9]. These studies highlight the need for consistent multiphysics modelling approaches that balance model detail, numerical robustness, and computational efficiency.

Recent CFD studies related to rotary kilns include the modelling of the aerodynamics [10], combustion [11, 12, 13, 14, 15], multiphase flow [16], coupling with a 1D material bed model [17], heat transfer coupled with DEM for granular flow [18, 19], and combinations hereof [20, 21, 22]. More recent work has addressed conjugate heat transfer and large-scale kiln simulations [23, 24, 25, 26]. Together, these studies confirm that as more physical processes are included, the mathematical models are often simplified to maintain numerical stability and acceptable computational cost.

In this dissertation, the following physical phenomena, which occur inside the rotary kiln, are considered:

- turbulent flow and mixing of natural gas, preheated air and combustion products using an (unsteady) Reynolds-Averaged Navier Stokes model;
- chemical reactions between fuel and oxygen producing heat and combustion products;
- transfer of heat generated by the flame through the freeboard space filled by the mixture of fuel, preheated air and combustion products;
- conductive heat transfer within the refractory lining and steel shell as well as the radiative heat loss to the ambient surroundings.

In this work, heat transfer between the reacting flow and the kiln walls is modelled using the conjugate heat transfer (CHT) approach. This approach resolves conduction in the solid region and convection in the fluid region in a fully coupled manner, which is done by enforcing continuity of temperature and heat flux at the fluid–solid interface. CHT is implemented in many popular CFD codes. There are several publications available on furnace models where combustion and CHT are coupled. For example, in [27] the prediction of the furnace wall heat distribution was made with CD-Adapco's STAR-CCM+. ANSYS CFX was used in [28] to model the heat distribution, while ANSYS Fluent was the CFD tool for the works of other researchers [29, 30, 31].

Before this research started, there were no publications on coupling turbulent combustion and CHT with the open source CFD-toolbox OpenFOAM. OpenFOAM sets a structured object-oriented framework and includes numerous applications to solve different kinds of CFD-related problems. The source code is fully accessible and allows building new or modified applications while making use of existing libraries, models and utilities to link them. OpenFOAM also allows high-performance computing using e.g., MPI and GPUs that do not require any licence costs and hence may lead to significant savings for large and complex problems. There are numerous studies in which combustion solvers of OpenFOAM were benchmarked against different experiments and other solvers (e.g., [32, 33, 34, 35]) and some include thermal radiation for the heat transfer (e.g., [36]). The capabilities of OpenFOAM's CHT solver have also been studied extensively, and some investigations into this matter (with and without radiative heat transfer) can be found in [37, 38, 39, 40].

1.4. OBJECTIVES

The aim of this research is to contribute to the development and application of a multiphysics CFD solver for high temperature furnaces, with focus on rotary kilns. The solver is intended to model the coupled behavior of turbulent flow, combustion, heat transfer, and thermal NO formation in a single numerical environment, and is developed within the open-source toolbox OpenFOAM, enabling flexible model selection and efficient large-scale simulations.

The objectives of this work are:

- to develop an application in OpenFOAM that couples the combustion solver `reactingFoam` with the conjugate heat transfer solver `chtMultiRegionFoam`,
- to develop an application in OpenFOAM for predicting thermal NO formation in a postprocessing stage for when the species is not already included in the reaction mechanism,
- to assess the influence of numerical solution methods on the convergence and demonstrate the capabilities of the solver with benchmark studies,
- and to gain physical insight into the interaction between turbulence, combustion, (wall) heat transfer, and NO formation.

In addition to these scientific objectives, the developed solver is applied to an industrial rotary kiln as a representative application case. In this context, thermal NO formation is an important environmental concern. Sensitivity studies are performed to investigate the influence of operating and design parameters, and promising NO mitigation strategies are identified that can guide future experimental and numerical investigations.

1.5. THESIS OUTLINE

With the focus on building a fully integrated mathematical solver covering all necessary physics, this dissertation is divided into three parts: theory, applications and conclusions.

The theoretical part consists of two chapters. In Chapter 2 we start with defining the mathematical problem, followed by the elaboration of all the required physical models from top to bottom which are needed to simulate the combustion and heat transfer process of the rotary kiln (or industrial furnace in general), including the pros and cons of certain used models. Chapter 3 discusses the numerical implementation of OpenFOAM, which starts with the finite volume method of discretization of the partial differential equations, followed by the coupling algorithms and several solution methods required to solve the linear systems of equations.

In the second part, which consists of three chapters, different applications of the built mathematical solver are shown in ascending order of geometric complexity. Chapter 4 reports the verification of the solver, where a cylindrical refractory wall is added to the prominent turbulent combustion benchmark case from the Sandia laboratory. Also, the combustion module of the solver is validated. The chapter continues with applying the solver on a more complex application. The burner flow reactor adds recirculation to the problem, which occurs in almost every combustion application in the industry. The conjugate heat transfer module is studied and compared with ANSYS Fluent. The importance of conjugate heat transfer for rotary kilns is shown in Chapter 5. In here, the solver is benchmarked against a CFD study of a comparable kiln which only imposes a fixed wall temperature at the wall.

The final application in Chapter 6 is the actual rotary kiln of Almatis, which faces the challenge of reducing the thermal NO emissions. In here, we propose three possible solutions for the non-premixed combustion in order to reduce hot spots by modifying one of the following:

- change to higher air-fuel ratios;
- modify the air inlet geometry;
- apply exhaust gas recirculation.

In the final part, Chapter 7, the main conclusions and findings of this dissertation are summarised, and recommendations are given for future studies.

The outline is summarised in the diagram below.

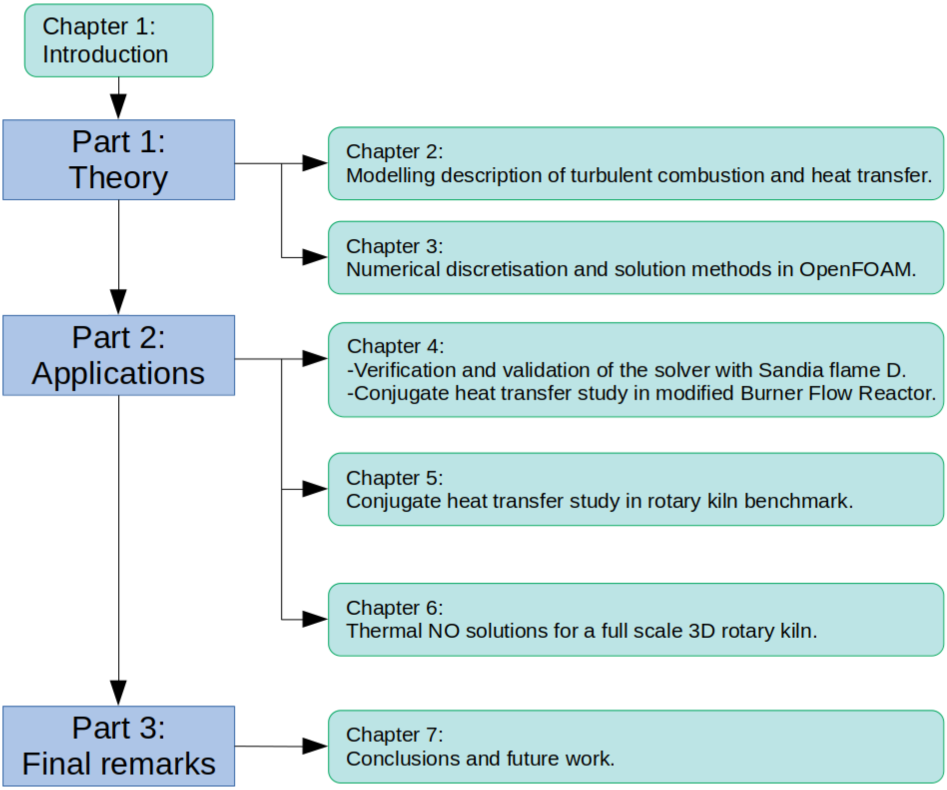

Figure 1.2: Outline of this dissertation.

I

THEORY

2

MODELLING TURBULENT COMBUSTION AND HEAT TRANSFER

2.1. INTRODUCTION

The heating process that the raw mix undergoes inside a rotary kiln can be predicted mathematically by modelling the occurring multi-physical phenomena and their interaction. These physical phenomena can be split into three parts: turbulent flow, combustion and heat transfer.

The interaction of the turbulent flow, combustion and heat transfer can be described in a loop form. The turbulent flow carries the reactant species, allowing them to mix so that reaction occurs. Heat is released by the reaction, which is partially absorbed and reflected by the raw mix and the kiln's refractory wall. This leads to a volume expansion and alters the turbulent flow. Thus, the model should be capable of solving the transport equations for mass, momentum, chemical species and energy of the reactive flow in the freeboard (i.e., the 'empty' part of the kiln), as well as the heat transfer between the freeboard and the refractory wall.

This chapter starts with presenting the governing transport equations in section 2.2, which align with OpenFOAM practice. This is followed by the basics of chemical kinetics in section 2.3, which is required for combustion modelling. The post-processing of the thermal NO formation is described separately in section 2.4. Section 2.5 explains how the unknown terms due to averaging of the turbulent fields are modeled, and the chapter ends with an elaboration of the heat transfer between the gas and solid in section 2.6.

2.2. GOVERNING EQUATIONS

We start with an overview of the basic conservation laws that set the mathematical framework for the above-mentioned multi-physical problem that we want to solve. In general these partial differential equations describe the evolution of some physical property that is carried along by the flow, say an arbitrary scalar or vector variable ϕ, and is written in the following differential form:

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{transient}}+\underbrace{\nabla\cdot(\rho\boldsymbol{U}\phi)}_{\text{advection}}=\underbrace{\nabla\cdot\Gamma\nabla\phi}_{\text{diffusion}}+\underbrace{Q_\phi}_{\text{source}}, \tag{2.1}$$

where ρ is the fluid density, $\boldsymbol{U}$ the fluid velocity (vectors are always in boldface), Γ a diffusivity parameter and Q_ϕ a source (or sink) of ϕ . The conservation laws may consist of some or all of the following terms related to the flow's characteristics: time rate of change, advection, diffusion and source. The advection term can be recognised by the divergence of the velocity and the diffusion term by the Laplace operator. Other recurring names for these types of equations are balance equations, advection-diffusion-reaction equations or transport equations, and the designations are often used interchangeably throughout this dissertation.

2.2.1. CONTINUITY

The first basic physical law that is considered is the conservation of mass, where it is stated that mass can not be generated nor destroyed. The rate of change of mass in a given volume fixed in space (or control volume) must be equal to the total difference between incoming and outgoing mass fluxes through the entire surface of the considered volume. This equation is also known as the continuity equation

$$\frac{\partial\rho}{\partial t}+\nabla\cdot\rho\boldsymbol{U}=0. \tag{2.2}$$

The first term represents the time rate of change of mass within the control volume, also known as the transient term, while the second term shows the net flux of the mass flowing into or out of the control volume and is called the advection term.

For steady-state simulations, the transient term is omitted, and if incompressible and non-reacting flows are considered, the density can be excluded from the equation, which leads to

$$\nabla \cdot \boldsymbol{U} = 0. \tag{2.3}$$

2.2.2. CONSERVATION OF MOMENTUM

The conservation of momentum is based on Newton's second law, and it describes how the flow propagates. It is a balance between the acceleration of mass (i.e., the time rate of change of momentum) of the flow within the control volume, and the net forces applied on the control volume. Two types of forces can be distinguished:

- surface forces: pressure and viscous shear stress,
- body forces: e.g. gravity.

However, due to strong advection, the gravity is neglected in this dissertation, and only the pressure and viscous stresses are applicable. Putting the mass acceleration term on the left-hand side (LHS) and the forces on the right-hand side (RHS) leads to

$$\rho\left(\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{U} \cdot \nabla \boldsymbol{U}\right) = -\nabla p + \nabla \cdot \tau, \tag{2.4}$$

where p denotes the fluid pressure, and τ is the deviatoric viscous stress tensor, which for compressible flows is defined as [41]:

$$\tau = \mu \left[\nabla \boldsymbol{U} + (\nabla \boldsymbol{U})^{\mathsf{T}} - \frac{2}{3}(\nabla \cdot \boldsymbol{U})\mathbf{I} \right], \tag{2.5}$$

where μ is the dynamic viscosity of the fluid, the superscript T denotes tensor transposition, and $\mathbf{I}$ is the identity tensor. The resulting momentum balance equations (Eq. 2.4) combined with the mass balance equation (Eq. 2.2) are known as the Navier-Stokes equations.

In case of non-thermal flows, the viscosity is constant. For thermal and reacting flows, the viscosity increases with temperature and their relation is given by the Sutherland model as follows:

$$\mu = \frac{A_s \sqrt{T}}{1 + T_s / T}, \tag{2.6}$$

where A_s is the Sutherland coefficient and T_s the Sutherland temperature. These are constant values and depend on the composition of the fluid. The temperature T is determined by the energy conservation law, described in section 2.2.4.

For incompressible, non-reacting and steady-state flows, the transient term can be excluded from Eq 2.4, while the viscous stress term is drastically simplified, and the kinematic viscosity ν is used instead of the dynamic viscosity. Then, division by the density on both sides leads to

$$\boldsymbol{U} \cdot \nabla \boldsymbol{U} = -\nabla \frac{p}{\rho} + \nabla \cdot \nu \nabla \boldsymbol{U}. \tag{2.7}$$

We can see now the resemblance of the diffusion terms in Eqs. 2.1 and 2.7 more clearly. Note that the first RHS term is known as the kinematic pressure (in OpenFOAM still named p but with different unit), and that the density here is not a variable but a reference value.

By using Eq. 2.3, the velocity and pressure in Eq. 2.7 can be decoupled in two separate equations, which are solved iteratively. More on these algorithms is presented in section 3.2.5.

2.2.3. SPECIES TRANSPORT

The combustion process involves the transport (and reaction) of chemical species. Given that there are N chemical species in the reacting mixture, the species are characterised through their mass fractions Y_θ , where $\theta = 1, 2, \ldots , N$. The resulting advection-diffusion-reaction equation for the chemical species θ is given as:

$$\frac{\partial(\rho Y_\theta)}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} Y_\theta) = \nabla \cdot \rho \Gamma \nabla Y_\theta + R_\theta. \tag{2.8}$$

The first RHS term is the molecular diffusion flux that is approximated by Fick's law, which is a simplified assumption that the molecular diffusion is proportional to the gradient of the mass fraction of the species θ, and a diffusion constant Γ [42].

The second RHS term is the source term of species θ, represented by its reaction rate R_θ. The elaboration of the chemical source term is presented in subsection 2.3.

The species transport equations are not conservation laws by definition, as species can be created or destroyed, though the total amount of each element is conserved. These equations are excluded for all single gas cases in this dissertation.

2.2.4. CONSERVATION OF ENERGY

According to the first law of thermodynamics, the total energy within an isolated system equals the sum of the internal energy and kinetic energy. The total energy cannot be created or destroyed but can only be transformed from one form to another. The conservation of energy requires some attention, as it can be written in different forms and consists of many terms that are often neglected depending on the application.

The most commonly used energy variables are the specific internal energy e and specific enthalpy h, which is the sum of internal energy and kinematic pressure:

$$h = e + \frac{p}{\rho}. \tag{2.9}$$

As a change of enthalpy directly relates to heat addition or removal in constant pressure processes, the enthalpy is the variable of choice for many combustion systems, including rotary kilns. The full conservation of energy in terms of kinetic energy and specific enthalpy is then written as follows [41]:

$$\frac{\partial(\rho h)}{\partial t} + \nabla\cdot(\rho \boldsymbol{U} h) + \frac{\partial(\rho K)}{\partial t} + \nabla\cdot(\rho \boldsymbol{U} K) = \underbrace{\frac{\partial p}{\partial t} + \boldsymbol{U}\cdot\nabla p + \nabla\cdot(\tau\cdot\boldsymbol{U}) + \rho\mathbf{g}\cdot\boldsymbol{U}}_{work} + \underbrace{\nabla\cdot\frac{\lambda}{c_p}\nabla h + Q}_{heat}, \tag{2.10}$$

where $K \equiv |\boldsymbol{U}|^2/2$ is the specific kinetic energy, λ the thermal conductivity, and c_p the specific heat capacity at constant pressure.

The RHS terms of Eq. 2.10 are distinguished between the net work being done on the surroundings, and heat addition via conduction and other sources. As gravity $\mathbf{g}$ is already neglected in Eq. 2.4, the change in potential energy $\rho\mathbf{g}\cdot\boldsymbol{U}$ can be excluded. The contribution of viscous heating $\nabla\cdot(\tau\cdot\boldsymbol{U})$ is also very small in combustion cases and only significant at very high velocities. The only diffusive flux that is considered is Fourier's law, whereas molecular species diffusion is only important at very low speeds and therefore neglected [43] . The diffusion term can also be expressed via the thermal diffusivity α:

$$\alpha = \frac{\lambda}{\rho c_p}, \tag{2.11}$$

which changes the diffusion term into $\nabla\cdot\rho\alpha\nabla h$. The enthalpy h in its absolute form includes the heat of formation:

$$h = h_s + \sum_{\theta=1}^{N} \Delta h^o_{f,\theta} Y_\theta, \tag{2.12}$$

where $\Delta h^o_{f,\theta}$ and Y_θ are respectively the formation enthalpy and mass fraction of species θ. h_s is the *sensible* enthalpy, which is a function of the temperature

$$h_s = \int_{T_{\text{ref}}}^{T} c_p(T)dT. \tag{2.13}$$

As we want to account for the energy change due to combustion in a separate term, it makes more sense to use the sensible enthalpy as the transported variable for the energy conservation.

Although the kinetic energy terms are often removed in the combustion literature, it is still included by default in the combustion solvers of OpenFOAM, as it better predicts the temperature change across a normal shock [44]. This may occur in the fuel nozzles of the kiln's burner, but not in the freeboard where the velocities are very low. For the sake of focus in further derivations, the eventual energy balance equation reduces to

$$\frac{\partial(\rho h_s)}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} h_s) = \frac{Dp}{Dt} + \nabla \cdot \rho\alpha\nabla h_s + Q_c + Q_r, \tag{2.14}$$

where the pressure work $\frac{\partial p}{\partial t} + \boldsymbol{U} \cdot \nabla p$ in Eq. 2.10 is written in the material derivative form $\frac{Dp}{Dt}$. Two heat source terms appear on the RHS. Q_c is the net *rate of* heat release by combustion that is separated from the absolute enthalpy

$$Q_c = -\sum_{\theta=1}^{N} \Delta h^o_{f,\theta} R_\theta, \tag{2.15}$$

and Q_r is the source term due to thermal radiation, which will be elaborated on in section 2.5.4.

2.2.5. THERMODYNAMIC EQUATION OF STATE

To simulate the multi-physical interactions in the combustion process, the transport equations are linked through several parameters, which are highlighted here.

The variable that occurs in every equation (except for incompressible flows) is the fluid density. For ideal gases, the dependence of density on pressure and temperature is expressed via the equation of state:

$$\rho = \frac{1}{RT} p, \tag{2.16}$$

where R is the specific gas constant. For the implemented compressible thermophysical models, the density is updated or corrected using the compressibility parameter ψ (due to high speed or temperature change):

$$\rho = \psi p, \tag{2.17}$$

where $\psi = (RT)^{-1}$.

The viscosity and conductivity are related via the Prandtl number Pr, which is the ratio of momentum diffusivity to thermal diffusivity:

$$\mathrm{Pr} = \frac{\nu}{\alpha} = \frac{c_p \mu}{\lambda}. \tag{2.18}$$

A typical value for Pr is about 0.71 for air and many gases. The specific heat capacity is an important thermodynamic property which comes back in the conservation of energy, and where other properties are derived from. In the case of non-reacting and isothermal flows, c_p is constant. For reacting flows, the `JANAF` model is used, which calculates c_p as a function of temperature from a set of coefficients taken from `JANAF` thermochemical tables [45]:

$$c_{p,\theta} = R((((a_{4,\theta} T + a_{3,\theta}) T + a_{2,\theta}) T + a_{1,\theta}) T + a_{0,\theta}), \tag{2.19}$$

where the constants $a_{0,\theta}$ to $a_{4,\theta}$ are given for every chemical species. c_p is the mass-weighted sum of the heat capacities of every species $c_{p,\theta}$:

$$c_p = \sum_{\theta=1}^{N} c_{p,\theta} Y_\theta. \tag{2.20}$$

Using Eqs. 2.13 and 2.19, the temperature is calculated iteratively (using the Newton-Raphson method) from the solved sensible enthalpy field:

$$h_s = \int_{T_{ref}}^{T} \left(\sum_{\theta=1}^{N} c_{p,\theta} Y_\theta \right) dT. \tag{2.21}$$

2.3. CHEMICAL KINETICS

Combustion can be described as a reaction between fuel and oxygen. In this dissertation, the main fuel is methane (CH_4) and the oxidant is the oxygen (O_2) contained in air. Assuming that nitrogen in air is inert, the combustion ultimately produces carbon dioxide (CO_2) and water (H_2O):

$$\mathrm{CH_4 + 2\,O_2 + 7.52\,N_2 \xrightarrow{k_1} CO_2 + 2\,H_2O + 7.52\,N_2} \tag{2.22}$$

This is also known as the global methane/air combustion mechanism. The rate of reaction is determined by the reaction rate constant k_1. In reality, the combustion of methane consists of many intermediate reactions, where many species are created and destroyed at different rates. The study of reaction mechanisms and reaction rates is what chemical kinetics is about.

To illustrate the chemical kinetics, the two-step Westbrook and Dryer (WD) reaction mechanism of methane combustion is considered [46]:

$$\mathrm{CH_4 + 1.5\,O_2 \xrightarrow{k_1} CO + 2\,H_2O} \tag{2.23}$$

$$\mathrm{CO \;\; + 0.5\,O_2 \xleftrightarrow{k_2} CO_2} \tag{2.24}$$

This is another reaction mechanism that is used in this dissertation, and it is the simplest of the more detailed mechanisms. The two-step reaction mechanism involves an irreversible step of formation of the intermediate species carbon monoxide (CO) from incomplete combustion of methane, and a reversible step of carbon monoxide oxidation to carbon dioxide. To determine the chemical source term, we take the formation and consumption of CO as an example.

If we do not involve turbulence yet, then the net source of chemical species θ due to reaction is computed as the sum of the Arrhenius reaction sources over the N reactions that the particular species participates in:

$$R_\theta = W_\theta \sum_{j=1}^{N} \frac{\partial [C_\theta]}{\partial t}, \tag{2.25}$$

where W_θ is the molecular weight of species θ in g/mole, and $[C_\theta]$ the molar concentration of species θ .

Next is to define the reaction rate progress r of reaction step j of the mechanism, which involves M species, and is given by:

$$r_j = k_{j,f} \prod_{i=1}^{M} [C_{i,j}]^{\eta'_{\ell,j}} - k_{j,b} \prod_{i=1}^{M} [C_{i,j}]^{\eta''_{m,j}}, \tag{2.26}$$

where $C_{i,j}$ is the concentration of species i in reaction j, $\eta'_{\ell,j}$ is the stoichiometric coefficient of reactant ℓ in reaction j, and $\eta''_{m,j}$ is the stoichiometric coefficient for product m in reaction j. The forward and backward rate constants, $k_{j,f}$ and $k_{j,b}$, are given by Arrhenius expressions for reaction j

$$k_j = A_j T^{\beta_j} e^{\frac{-E_j}{\bar{R}T}} \tag{2.27}$$

where A is the empirical Arrhenius constant, T is the temperature at which the reaction occurs, β is the temperature exponent, E is the activation energy and $\bar{R}$ is the universal gas constant. Arrhenius expressions are simple and powerful because they allow one to define the detailed chemical kinetics accurately.

Applying this in the two-steps mechanism, the reaction rate of step 1 is

$$r_1 = k_{1,f}[\mathrm{CH_4}][\mathrm{O_2}]^{1.5}, \tag{2.28}$$

and that of step 2 is

$$r_2 = k_{2,f}[\mathrm{CO}][\mathrm{O_2}]^{0.5} - k_{2,b}[\mathrm{CO_2}]. \tag{2.29}$$

Taking CO as an example, this species is treated as the product in the first step and a reactant in the second step. Hence, the molar rate of change of species CO is calculated as follows:

$$\frac{\partial[\mathrm{CO}]}{\partial t} = k_{1,f}[\mathrm{CH_4}][\mathrm{O_2}]^{1.5} - k_{2,f}[\mathrm{CO}][\mathrm{O_2}]^{0.5} + k_{2,b}[\mathrm{CO_2}]. \tag{2.30}$$

Finally, the source term of the transport equation of CO is:

$$R_{\mathrm{CO}} = W_{\mathrm{CO}}\frac{\partial[\mathrm{CO}]}{\partial t} = \frac{\partial}{\partial t}\left(\rho Y_{\mathrm{CO}}\right). \tag{2.31}$$

The most widely used detailed reaction mechanism of methane/air combustion is the GRI-3.0 mechanism [47] and it involves 53 species and 325 reactions, which leads to very realistic predictions of flame temperature and emissions. However, this means that 53 additional species transport equations have to be resolved. Moreover, a very large and extremely stiff linear system of ordinary differential equations (ODE) of the chemical source terms is formed, which describes the chemical kinetics (such as Eq. 2.30). The reaction timescales often span many orders of magnitude (fast radical reactions versus slow oxidation steps). This wide separation of timescales produces Jacobian matrices with very large eigenvalue spreads, requiring fully implicit ODE solvers for numerical stability, as shown in [48].

To reduce the computational costs, the reaction mechanism can be simplified. The mean species source term can be calculated relatively fast when only the global or two-step reaction mechanism is used, but these are very crude simplifications of the actual chemical kinetics occurring in the real combustion process. This simplification often overestimates the rate of the reaction, and can give inaccurate predictions of the formation of certain species which are affected by other species that are left out. For thermal predictions, they are often acceptable.

The chemistry calculations with a very large mechanism can also be accelerated without losing information by using **in situ Adaptive Tabulation** (ISAT) [49]. The general idea is that during these calculations, the ISAT algorithm stores the thermochemistry variables at each thermochemical state variable $\phi = [Y_1, Y_\theta, ..., Y_N, T, p]$, where each reacted state $\phi'(t_0 + \Delta t)$ is a unique function of the unreacted state $\phi(t_o)$ and time-step Δt. Each unique reaction source term value then only has to be calculated once, and the resulting table can be interpolated to a certain accuracy, if it is more efficient than calculating the chemistry directly.

Another approach to reduce the computational resources without significantly losing information is to reduce the reaction mechanism in every time step by eliminating species and reactions that have small impact on the overall system. The **dynamic adaptive chemistry** (DAC) reduction scheme [50] is one of such algorithms that use this approach. The **tabulation of dynamic adaptive chemistry** (TDAC) is a thermochemistry accelerator which is developed in OpenFOAM [51] and combines both ISAT and DAC, where a speed-up factor of above 300 is obtained for a detailed 857-species iso-octane mechanism. In this dissertation a speed-up of 160 is shown for the much smaller GRI-3.0 mechanism.

This method provides exact solutions for both laminar and turbulent flames. For turbulent flames however, direct numerical simulation has too high computational cost and is replaced by the solution of averaged equations in this dissertation. This method leads to closure problems, due to the highly non-linear dependence of the Arrhenius chemical kinetics on temperature. Therefore, turbulent finite-rate combustion models also incorporate additional models for the effect of turbulence when solving the detailed Arrhenius chemical kinetics, which will be elaborated on in section 2.5.3.

2.4. THERMAL NO POST-PROCESSING

Thermal NO is the dominant component of NO_x formed at high temperature as a result of the reaction between the nitrogen and oxygen in the air stream. The formation increases exponentially, becomes significant above 1800K [52], and is determined by a set of highly temperature-dependent chemical reactions known as the extended Zeldovich mechanism:

$$\mathrm{O} + \mathrm{N_2} \xleftrightarrow{\mathrm{k_1}} \mathrm{NO} + \mathrm{N}, \tag{2.32}$$

$$\mathrm{N} + \mathrm{O_2} \xleftrightarrow{\mathrm{k_2}} \mathrm{NO} + \mathrm{O}, \tag{2.33}$$

$$\mathrm{N} + \mathrm{OH} \xleftrightarrow{\mathrm{k_3}} \mathrm{NO} + \mathrm{H}. \tag{2.34}$$

The forward and backward reaction rate constants have been measured in numerous studies, of which the prominent ones are given as [53]:

$$k_{1,f} = 1.8 \times 10^8 e^{-38370/T} \quad k_{1,b} = 3.8 \times 10^7 e^{-425/T}, \tag{2.35}$$

$$k_{2,f} = 1.8 \times 10^4 T e^{-4680/T} \quad k_{2,b} = 3.8 \times 10^3 T e^{-20820/T}, \tag{2.36}$$

$$k_{3,f} = 7.1 \times 10^7 e^{-450/T} \quad k_{3,b} = 1.7 \times 10^8 e^{-24560/T}. \tag{2.37}$$

Notice the very low forward rate constant $k_{1,f}$ (Eq. 2.35), which is due to the extraordinarily high activation energy that is required to break the strong N_2 triple bond for the first reaction 2.32. The first reaction is the slowest reaction in the Zeldovich mechanism, and therefore it is the rate-limiting reaction in the formation of NO. The reaction will only become sufficiently fast at high temperatures; hence, it is only important above 1800K. Because it requires capturing the microscopic hot spots in the turbulent flame, the value of NO formation cannot be accurately predicted without a model for the temperature fluctuations, though trends can be fairly predicted.

Thermal NO formation can be predicted by using the full GRI-3.0 mechanism, which includes the Zeldovich mechanism. If one is mainly interested in the formation of $\mathrm{NO_x}$, this leads to unnecessarily long computational costs. The kinetics of thermal NO has shown that its formation is much slower than the other species, which allows the Zeldovich mechanism to be decoupled from the main combustion process and gives the ability to use a much shorter reaction mechanism. The NO formation is then post-processed from a solved reacting field.

A thermal NO post-processor is created for OpenFOAM in [54] which is inspired by the $\mathrm{NO_x}$ post-processor of ANSYS Fluent [55].

The NO transport equation is given by:

$$\frac{\partial(\rho Y_{\mathrm{NO}})}{\partial t} + \nabla \cdot (\rho \boldsymbol{U} Y_{\mathrm{NO}}) = \nabla \cdot \rho \Gamma \nabla Y_{\mathrm{NO}} + R_{\mathrm{NO}}, \tag{2.38}$$

and similarly to CO in section 2.3, by using the Zeldovich mechanism and corresponding reaction rate constants, the rate of formation of NO is given by:

$$\begin{aligned}\frac{d[\mathrm{NO}]}{dt} =& k_{1,f}[\mathrm{O}][\mathrm{N_2}] + k_{2,f}[\mathrm{N}][\mathrm{O_2}] + k_{3,f}[\mathrm{N}][\mathrm{OH}] - \\ & k_{1,b}[\mathrm{NO}][\mathrm{N}] - k_{2,b}[\mathrm{NO}][\mathrm{O}] - k_{3,b}[\mathrm{NO}][\mathrm{H}].\end{aligned} \tag{2.39}$$

For most combustion cases, except in extremely fuel-rich conditions, the rate of formation as is described in Eq. 2.39 can be simplified by making *Quasi-Steady Assumption* for the radical species N:

$$\frac{d[\mathrm{NO}]}{dt} = k_{1,f}[\mathrm{O}][\mathrm{N_2}]\frac{\left(1 - \frac{k_{1,b}k_{2,b}[\mathrm{NO}]^2}{k_{1,f}k_{2,f}[\mathrm{N_2}][\mathrm{O_2}]}\right)}{\left(1 + \frac{k_{1,b}[\mathrm{NO}]}{k_{2,f}[\mathrm{O_2}]+k_{3,f}[\mathrm{OH}]}\right)}, \tag{2.40}$$

which now only requires calculating the O and OH radicals in addition to the main reactants. Eq. 2.40 shows that NO formation is very sensitive to temperature and O concentration. There are several ways to determine the concentrations of O and OH. Their transport can be directly modelled if their reactions are included in the mechanism. This is the most accurate approach. If not, then another way is to calculate the concentrations of O and OH using a partial equilibrium approach:

$$[\mathrm{O}] = 36.64\,T^{1/2}e^{-27123/T}[\mathrm{O_2}]^{1/2}, \tag{2.41}$$

$$[\mathrm{OH}] = 2.129\cdot 10^2\,T^{-0.57}e^{-4595/T}[\mathrm{O_2}]^{1/2}[\mathrm{H_2O}]^{1/2}. \tag{2.42}$$

The application of the thermal NO post-processor [56] will be shown in chapter 6. The main limitation of this code is that it does not include a turbulence-chemistry interaction model, and therefore the fluctuating temperature and species concentrations are not captured, which are the major source of NO formation. Room for improvement is to include the effects of temporal fluctuations using a presumed *Probability Density Function* (PDF) approach [57].

2.5. TURBULENCE CLOSURE MODELS

Turbulence is chaotic in nature, and to resolve all the fluctuating components for a large industrial application is prohibitive. A necessary simplification is to average the velocity and the other flow-dependent variables in time. This process is known as Reynolds averaging for incompressible flows, and density-weighted or Favre averaging for variable density flows. The averaging is meant to cancel out the fluctuating components, so that the transport equations only have to resolve the mean flow. However, the averaging also causes unknown terms to appear which have to be modelled in some way in order to close the system of equations.

In the fluid domain, the Favre-averaged transport equations of mass, momentum, chemical species and sensible enthalpy [58][42] are respectively described by:

$$\begin{aligned}
\frac{\partial(\bar{\rho})}{\partial t} + \nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}) &= 0, && (2.43)\\
\frac{\partial(\bar{\rho}\widetilde{\boldsymbol{U}})}{\partial t} + \nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{U}}) &= -\nabla\bar{p} + \nabla\cdot\widetilde{\tau} - \nabla\cdot\bar{\rho}\widetilde{U''U''}, && (2.44)\\
\frac{\partial(\bar{\rho}\widetilde{Y_\theta})}{\partial t} + \nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}\widetilde{Y_\theta}) &= \nabla\cdot\bar{\rho}\Gamma\nabla\widetilde{Y_\theta} - \nabla\cdot\bar{\rho}\widetilde{Y''_\theta U''} + \widetilde{R}_\theta, && (2.45)\\
\frac{\partial(\bar{\rho}\widetilde{h}_s)}{\partial t} + \nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}\widetilde{h}_s) &= \frac{D\bar{p}}{Dt} + \nabla\cdot\bar{\rho}\alpha\nabla\widetilde{h}_s - \nabla\cdot\bar{\rho}\widetilde{h''_s U''} + \widetilde{Q}_c + \widetilde{Q}_r, && (2.46)
\end{aligned}$$

where the over-bar and tilde notations respectively stand for the Reynolds and Favre average values, while the double quotation marks denote the fluctuating components relative to the Favre average. The modelling of the fluctuating components and mean source terms is explained in this section.

2.5.1. REYNOLDS STRESSES

The unknown Reynolds stresses (last term of Eq. 2.44) are solved by employing the Boussinesq hypothesis that is based on the assumption that in turbulent flows, the relation between the Reynolds stress and viscosity is similar to that of the stress tensor in laminar flows (Eq. 2.5), but with increased (turbulent) viscosity:

$$-\bar{\rho}\widetilde{U''_i U''_j} = \mu_t\left(\frac{\partial\widetilde{U}_i}{\partial x_j} + \frac{\partial\widetilde{U}_j}{\partial x_i}\right) - \frac{2}{3}\rho k\delta_{ij}, \qquad (2.47)$$

where the proportionality factor μ_t is the turbulent dynamic viscosity, δ_{ij} is the Kronecker delta, and k is the turbulent kinetic energy (not to be confused with the reaction rate constant k_j), defined as:

$$k = \sum\frac{1}{2}\widetilde{U''_i U''_i}. \qquad (2.48)$$

The Reynolds stresses in this dissertation are closed with the Realizable k-ε turbulence model [59]. In the validation of [13], this turbulence model was compared with the other two k-ε models and showed to have the best fit with experimental data of a piloted turbulent diffusion flame. The model solves two additional transport equations, one for the turbulent kinetic energy k, and the other for its dissipation rate ε:

$$\frac{\partial(\bar{\rho}k)}{\partial t}+\nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}k)=\nabla\cdot\left[\left(\mu+\frac{\mu_t}{\Theta_k}\right)\nabla k\right]+P_k-\bar{\rho}\varepsilon, \tag{2.49}$$

$$\frac{\partial(\bar{\rho}\varepsilon)}{\partial t}+\nabla\cdot(\bar{\rho}\widetilde{\boldsymbol{U}}\varepsilon)=\nabla\cdot\left[\left(\mu+\frac{\mu_t}{\Theta_\varepsilon}\right)\nabla\varepsilon\right]+\bar{\rho}c_1S\varepsilon-\bar{\rho}c_2\frac{\varepsilon^2}{k+\sqrt{\nu\varepsilon}}, \tag{2.50}$$

where P_k is the production term which represents the rate at which turbulent kinetic energy is generated in the flow due to mean velocity gradients. It can be expressed as:

$$P_k=\mu_t S^2, \tag{2.51}$$

where S is the modulus of the mean strain rate tensor:

$$S\equiv\sqrt{2S_{ij}S_{ij}}, \qquad S_{ij}=\frac{1}{2}\left(\frac{\partial\widetilde{U}_i}{\partial x_j}+\frac{\partial\widetilde{U}_j}{\partial x_i}\right). \tag{2.52}$$

Θ_k,Θ_ε and c_2 are constants, and c_1 is a function of k, ε and S. The effect of buoyancy and other source terms are neglected in Eqs. 2.49 and 2.50.

The turbulent viscosity can be determined with k and ε by the following relation:

$$\mu_t=\bar{\rho}c_\mu\frac{k^2}{\varepsilon}, \tag{2.53}$$

where in the Realizable k-ε model, c_μ is a function of k, ε, S, and the mean rotation rate. This is one of the major differences compared to other k-ε models where c_μ is a constant.

2.5.2. TURBULENT SCALAR FLUXES

The turbulent scalar fluxes $\bar{\rho}\widetilde{\phi''u''}$ for the scalars Y_θ and h_s (both denoted as ϕ) are closed with the Gradient diffusion assumption

$$-\bar{\rho}\widetilde{\phi''U''}=\bar{\rho}\Gamma_t\nabla\widetilde{\phi}, \tag{2.54}$$

where Γ_t is the turbulent diffusivity determined by the turbulent kinematic viscosity ν_t and the turbulent Prandtl number Pr_t for the enthalpy transport, or the turbulent Schmidt number Sc_t for the species transport:

$$\Gamma_{t,h} = \frac{\nu_t}{\mathrm{Pr}_t}, \tag{2.55}$$

$$\Gamma_{t,\theta} = \frac{\nu_t}{\mathrm{Sc}_t}. \tag{2.56}$$

When the turbulent Lewis number, $\mathrm{Le}_t = \mathrm{Sc}_t/\mathrm{Pr}_t$, is assumed to be equal to unity, the turbulent diffusivity is the same in both the energy and species transport equations.

With the defined turbulent diffusivities, the turbulent scalar fluxes are combined with the laminar diffusion terms in the transport equations. The thermal diffusivity in the averaged transport of the specific sensible enthalpy (Eq. 2.46) is replaced by the effective diffusivity α_{eff}. From Eqs 2.54 and 2.55, α_{eff} is defined as

$$\alpha_{\mathrm{eff}} = \frac{\nu}{\mathrm{Pr}} + \frac{\nu_t}{\mathrm{Pr}_t}, \tag{2.57}$$

where the turbulent Prandtl number, from experimental data, has an average value of 0.85.

The mass diffusivity in the chemical species transport is given by the laminar diffusion constant Γ_θ, which is the molecular diffusivity of species θ in the mixture. In the same way as with the enthalpy transport, the averaged species transport equation (Eq. 2.45) combines the laminar and turbulent diffusion constants:

$$\Gamma_{\mathrm{eff}} = \frac{\nu}{\mathrm{Sc}} + \frac{\nu_t}{\mathrm{Sc}_t}. \tag{2.58}$$

The laminar viscosity ν is the variable for the laminar molecular diffusivity in OpenFOAM. This can be constant or temperature dependent (e.g., Sutherland model). The molecular diffusivities of most gases are in the same order of magnitude as the laminar viscosity of the mixture at room temperature, but can increase significantly at higher temperatures and with certain species. In turbulent combustion however, the laminar diffusion is generally much smaller than the turbulent diffusion, and the specification of detailed laminar diffusion properties in turbulent flows is generally not necessary [55].

In most of the CFD studies, the default value of Sc_t is between 0.7 and 0.9; however, the optimum values are widely distributed in the range of 0.2–1.3, and the specific value selected has a significant effect on the prediction results [60]. Sc_t is assumed to be unity in OpenFOAM and not adjustable, but can be modified in the source code.

2.5.3. TURBULENCE-CHEMISTRY INTERACTION

The mean heat release due to combustion $\widetilde{Q_c}$ (Eq. 2.46) is related to the mean chemical source term $\widetilde{R}_\theta$

$$\widetilde{Q_c} = -\sum_{\theta=1}^{N} \Delta h^o_{f,\theta} \widetilde{R_\theta}. \tag{2.59}$$

Several combustion models exist to determine the value of the chemical source term, from finite rate models that account for detailed chemical reactions determined by the Arrhenius expressions (as discussed in section 2.3), to infinitely fast models that consider the turbulent mixing as the dominant factor of the chemical reactions. They will be elaborated here.

The **Partially Stirred Reactor** (PaSR) model [61] allows for the detailed Arrhenius chemical kinetics to be incorporated in turbulent reacting flows. It assumes that each cell is divided into a non-reacting zone and a reaction zone that is treated as a perfectly stirred reactor. The fraction of the reaction zone is proportional to the ratio of the chemical reaction time t_c to the total conversion time $t_c + t_{\mathrm{mix}}$:

$$\gamma = \frac{t_c}{t_c + t_{\mathrm{mix}}}. \tag{2.60}$$

The turbulence mixing time t_{mix} characterizes the exchange process between the reacting and non-reacting mixture, and is determined via the k-ε model as

$$t_{\mathrm{mix}} = c_{\mathrm{mix}} \sqrt{\frac{\mu_{\mathrm{eff}}}{\bar{\rho}\varepsilon}}, \tag{2.61}$$

where c_{mix} is a constant and μ_{eff} is the sum of the laminar and turbulent dynamic viscosity. Then the mean source term is calculated as $\widetilde{R}_\theta = \gamma R_\theta$, where R_θ is the laminar reaction rate of species θ (Eq. 2.25).

The chemical time scale can be determined with the following relation:

$$\frac{1}{t_c} = \max\left(\frac{-\partial R_\theta}{\partial Y_\theta} \frac{1}{\bar{\rho}}\right). \tag{2.62}$$

The PaSR model is a detailed combustion model which can be used to solve the full reaction mechanism, but due to the expensive ODE calculations of the source terms, it is a slow model, especially when the chemical time scale is orders of magnitude smaller than the global flow time scale, which is handled via *Operator-Splitting* [62, 63]. The model can be accelerated by utilising a reduced version of the reaction mechanism (at the cost of accurate emission prediction) or by making use of tabulation and adaptive chemistry reduction with the TDAC method.

As the detailed chemical kinetics require the source terms to be integrated over time with the implicit ODE solver, the PaSR model is only applicable for transient or semi-transient simulations. For very large industrial applications where solving the flow is already very costly, this is a major drawback, even with chemistry accelerators.

For steady-state simulations, we therefore implement the **Eddy Dissipation Model** (EDM) [64] by modifying the `infinitelyFastChemistry` model in the standard library of OpenFOAM. The EDM is an attractive model due to its computational efficiency and reasonable accuracy. The rate of reaction is taken as the minimum of the three rates for fuel, oxidant and products

$$\widetilde{R}_\theta = -A\bar{\rho}\frac{\varepsilon}{k}\min\left(\bar{Y}_F, \frac{\bar{Y}_O}{\eta}, B\frac{\bar{Y}_P}{1+\eta}\right), \tag{2.63}$$

where A and B are user-defined model constants that by default are respectively equal to 4 and 0.5, and η is the stoichiometric coefficient. The time step in the `infinitelyFastChemistry` model is replaced by the turbulence mixing time scale t_{mix}, which characterises the largest eddies:

$$t_{\text{mix}} = \frac{k}{\varepsilon}. \tag{2.64}$$

This definition is different to t_{mix} in the PaSR (Eq. 2.61), which characterises the fine structures. Being a 'mixed-is-burned' model type, the EDM excludes the effects of chemical kinetics and is based on the aspects of combustion occurring at high Reynolds and Damköhler numbers, where the chemical time scales are much smaller than the turbulent time scales [42]. Therefore, the underlying assumption for choosing this model is that in rotary kilns the combustion is indeed mainly controlled by turbulent mixing. The model is validated in [32] where it is shown that the flame temperatures are over-predicted, as can be expected from 1-step reaction mechanisms with infinitely fast reaction rates. However, the profiles are well captured. The overpredictions are lower when radiation is included.

Although not used in this dissertation, worth mentioning is the **Flamelet Generated Manifold** (FGM) [65], which is another reduced and tabulated chemistry approach that is developed for OpenFOAM in [66, 67]. The basic idea is that for flames with thin flame fronts and very fast chemistry, the most important reactions occur normal to the flame front, which are represented by one-dimensional flame structures called *flamelets*. Each flamelet has a unique thermochemical property (in terms of temperature and species mass fractions), and all flamelets are stored in low-dimensional manifolds that are a function of two or more control variables. Usually those control variables are the mixture fraction Z, which indicates the location between the fuel and oxidiser, and the progress variables PV, which denote the reacted state of the composition. So instead of solving the mean energy balance equation and transport equations of all the mean chemical species, only the transport of mean Z and mean PV have to be modelled. However, to account for turbulence using the PDF approach [68] , also the variances of Z and PV have to be included as control variables, leading to a total of four transport equations in addition to the Navier-Stokes equations. The manifolds can be extended with additional control variables to include more influences on the flame's characteristics, such as strain, enthalpy loss, and pressure [67].

The FGM model is superior to the EDM as it includes detailed chemistry, and the thermal NO post-processor will not be necessary. Compared with the PaSR, the stiff ODEs are avoided and (much) fewer transport equations are resolved, which provides for better numerical robustness and computational efficiency.

2.5.4. RADIATIVE HEAT TRANSFER MODELLING

The content of this section mainly follows references [69] and [70]. Thermal radiation describes the thermal energy transport by electromagnetic waves in the infrared spectrum, and accounts for over 90% of the heat transfer in rotary kilns [1]. It is therefore essential to obtain the mean radiation source term $\widetilde{Q_r}$ for the enthalpy transport equation.

To describe thermal radiation as a source term, the spectral radiation intensity is used, which is the energy per unit wavelength crossing a surface orthogonal to the direction of propagation of the beam per unit area. If we follow the thermal radiation in a specific direction and at a specific frequency, it can be noticed that when the beam passes through a volume, the intensity may be augmented or attenuated through emission, absorption, and scattering. This is mathematically described in the Radiative Transfer Equation (RTE). In combustion systems where the fuel is a gas, scattering can be neglected; hence, the scattering terms are left out, and for an emitting and absorbing non-grey medium the RTE is as follows:

$$\frac{dI_\chi(\mathbf{r},\mathbf{s})}{ds} = \underbrace{-\kappa_\chi I_\chi(\mathbf{r},\mathbf{s})}_{\text{absorption}} + \underbrace{\kappa_\chi I_{b\chi}(\mathbf{r})}_{\text{emission}}, \tag{2.65}$$

where for each wavelength χ, I_χ is the spectral radiation intensity at point $\mathbf{r}$ propagating along direction $\mathbf{s}$, and κ is the absorption coefficient. The black-body intensity $I_{b\chi}$ is given by Planck's law:

$$I_{b\chi} = \frac{c_1}{\pi\chi^5(exp(c_2/(\chi T) - 1)}, \tag{2.66}$$

where c_1 and c_2 are constants.

To obtain the divergence of the radiative heat flux $\nabla \cdot q_R$ as the source term for the enthalpy transport equation, the RTE is integrated in both the spectral variable and the solid angle of 4π:

$$\mathbf{q}_R = \int_0^\infty \int_{4\pi} \mathbf{s} I_\chi(\mathbf{r}, \mathbf{s}) d\Omega d\chi, \tag{2.67}$$

where $d\Omega$ is the differential solid angle.

The total incident radiative intensity is not known and if the temperature of the medium is unknown, the black-body radiative intensity is also unknown. Therefore, the RTE has to be solved together with the flow equations. Furthermore, to solve Eq. 2.65, the local absorption coefficient of the gas mixture has to be determined, which is a function of gas composition, temperature, wavelength, and pressure. Within the radiation spectrum, individual species absorb and emit through thousands of wavelengths, which makes it too expensive to calculate for all of them. Although models exist that average the amount of the wavelength lines up to a handful of broad bands (Wide Band Model), we chose to apply the grey gas assumption. A property of grey bodies is that they are independent of the spectral variable (e.g. wavelength), and the absorption coefficient is an average over the whole spectrum. This is a crude simplification and may lead to significant errors. Nevertheless, the model still accounts for the gas composition, temperature and pressure so that the absorption coefficient can be calculated with a polynomial for each species, given by

$$\kappa_\theta = \sum_{p=1}^{6} b_{p,\theta} \cdot T^{p-1}, \tag{2.68}$$

where T is the local gas temperature and the polynomial coefficients $b_{p,\theta}$ are specified for species θ at a certain pressure.

The grey gas polynomial coefficients used in this dissertation for the main species O_2, CH_4, N_2, H_2O and CO_2 are found in `reactingFoam`'s example tutorial file [71].

In the present work, turbulence-radiation interaction is neglected. All radiative quantities are evaluated using mean temperature, mean species concentrations, and mean absorption coefficients obtained from the Reynolds-averaged flow field. Under this assumption, the RTE may be interpreted as an equation for ensemble-averaged quantities. This approach is widely used in practical CFD simulations of turbulent combustion, but it introduces limitations that should be kept in mind.

Many solution methods have been developed to solve the radiative heat transfer problems. Two popular methods are the **P1 spherical harmonics** approximation and the **Discrete Ordinates Method** (DOM).

The **P1** approximation is a computationally cheap model with reasonable accuracy, which solves the following partial differential equation for a non-scattering medium:

$$-\nabla \cdot \left(\frac{1}{3\kappa} \nabla G \right) = \kappa (4\sigma T^4 - G), \tag{2.69}$$

where the radiation source term appears on the LHS of the equation, under the assumption of isotropic scattering. σ is the Stefan–Boltzmann constant 5.670373×10^{-8} $\mathrm{Wm^{-2}K^{-4}}$, and G is the total incident radiation, defined (for a grey medium) as:

$$G = \int_{4\pi} I(\mathbf{r}, \mathbf{s})\, d\Omega. \tag{2.70}$$

Here it is clear that by neglecting turbulence-radiation interaction it is assumed that $\widetilde{T}^4 \approx \widetilde{T^4}$. Since temperature fluctuations are significant in turbulent flames, this assumption generally leads to an underestimation of the mean radiative emissive power, hence to an over prediction of local flame temperatures. This also has a direct impact on thermal NO formation, which depends exponentially on temperature, in addition to the errors introduced by the grey gas assumptions. The P1 approximation loses its accuracy in optically thin media and geometries with high aspect ratios, such as a rotary kiln, but it is a decent model for initial studies.

The more accurate and more expensive **DOM** solves the RTE for a set of discrete directions (discretized in polar and azimuth angles) that span the total solid angle range of 4π around a point in space. The integrals over solid angles are approximated using a numerical quadrature rule. Therefore, the RTE may be written as follows for direction $\vec{s}^{m}$:

$$\frac{dI^m}{ds} = -\kappa I^m + \kappa I_b, \tag{2.71}$$

where the superscript m ($1 \leq m \leq M$) denotes the m-th direction and M is the total number of discrete directions.

Radiation has to be treated differently when considering solids. In combustion systems, solid boundaries are generally opaque and may be assumed to be grey and diffuse. If the solid is diffuse, its radiative properties are also independent of direction, so that emission and reflection happen diffusely (neglecting specular reflection). The boundary condition at the solid wall in the case of the P1 approximation is:

$$-\frac{1}{3\kappa}\mathbf{n}\cdot\nabla G = -\frac{\epsilon_w}{2(2-\epsilon_w)}(4\sigma T_w^4 - G_w), \tag{2.72}$$

where the subscript w denotes the wall surface, and ϵ_w is the wall emissivity. For the DOM, the wall boundary condition is [72]:

$$I_w(\vec{s}) = \epsilon_w I_{b,w} + \frac{1-\epsilon_w}{\pi}\int_{\vec{n}\cdot\vec{w}<0} I_w(\vec{w})|\vec{n}\cdot\vec{w}|d\Omega, \tag{2.73}$$

where the second RHS term is the reflected incoming radiation. The wall emissivity ranges from zero to one, where the value 1 represents a black body and the value 0 represents a perfect mirror. Note that for the P1 approximation, ϵ_w cannot be equal to zero.

2.6. CONJUGATE HEAT TRANSFER

Next to thermal radiation, the freeboard gas and refractory wall of the kiln exchange heat through conduction and convection as well, of which the latter two require coupling of the thermal energy transport between the fluid and solid domains. Two important conditions are therefore required at the interface of the domains to ensure continuity of both the temperature and heat flux:

$$T_{f,\text{int}} = T_{s,\text{int}} \tag{2.74}$$

and

$$\lambda_{\text{eff}}\frac{\partial T_f}{\partial n} = -\lambda_s\frac{\partial T_s}{\partial n} \tag{2.75}$$

where the subscripts f, s and int respectively stand for fluid, solid and interface. $\frac{\partial T}{\partial n}$ is the temperature gradient normal to the interface, and $\lambda_{\mathbf{eff}}$ is the *effective* conductivity which combines the laminar and turbulent components:

$$\lambda_{\text{eff}} = \rho c_p \alpha_{\text{eff}}. \tag{2.76}$$

How to determine the LHS and RHS terms of Eqs. 2.74 and 2.75 is elaborated here.

2.6.1. CONDUCTION

For solid regions, only the energy transfer needs to be solved, and therefore, the equation of enthalpy for solids, which is the following heat equation, has to be added to the list of transport equations (Eqs. 2.43-2.46):

$$\frac{\partial(\bar{\rho} h)}{\partial t} = \nabla \cdot (\lambda_s \nabla T). \tag{2.77}$$

Determining the temperature and heat flux at the interface boundary from the solid side simply requires solving the heat equation.

2.6.2. CONVECTIVE HEAT TRANSFER

Looking back at Eq. 2.75, the LHS term is the convective heat transfer. This can be calculated when the effective fluid conductivity λ_{eff} and the temperature gradient at the wall from the fluid side are determined. However, this is not as straightforward as calculating the heat flux at the solid side, which is pure conduction. On the fluid side, the influence of the boundary layer of the flow needs to be incorporated. The no-slip condition leads to very steep velocity gradients that affect the turbulent kinetic energy, the dissipation rate, the turbulent (or 'eddy') viscosity, the temperature and the thermal conductivity.

Certain low Reynolds turbulence models, such as the k-ω model, incorporate the wall shear stress and are able to resolve the boundary layer region of the velocity and turbulence fields, on the condition that the wall-adjacent cells are well within the viscous sub-layer, where the velocity profile is linear. As the variation across a cell is linear (2nd order accurate), the velocity gradients will be captured. This would require excessive mesh refinement and may lead to a substantial increase in required computational power to solve the resulting flow fields.

The k-ε models, that are primarily used in this dissertation, do not account for the wall damping effects. Instead of resolving the boundary layer directly, it is modelled by employing wall functions.

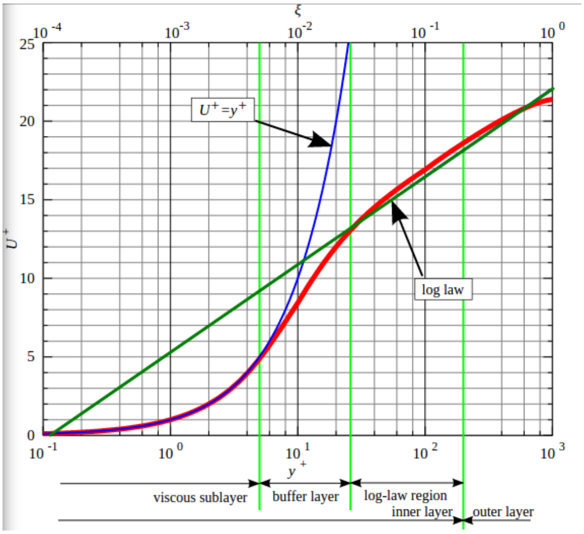

Figure 2.1: Law of the wall, horizontal velocity near the wall with the mixing length model [73].

Near wall treatment using wall functions

The wall functions make use of the universal behaviour of the flow near the wall [74]. These are empirical functions determined from experiments of parallel plane channel flows and are used to calculate the velocity, the turbulence properties and the thermal diffusivity in the boundary layer. These parameters depend on each other, and therefore, the wall functions need to be defined as wall boundary conditions for the velocity, temperature and turbulence fields.

Velocity Fig. 2.1 shows the familiar velocity profile near the wall. It is a plot of the non-dimensional flow velocity u^+ plotted against the non-dimensional distance normal to the wall y^+. Their relation as a function of the actual velocity u and wall distance y, respectively, are given by

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \qquad y^+ = \frac{y u_\tau}{\nu} \qquad u^+ = \frac{u}{u_\tau}, \tag{2.78}$$

where u_τ is the friction velocity, τ_w is the wall shear stress, and ν is the kinematic viscosity. The near wall region is divided into three regions:

- The viscous sub-layer ($y^+ < 5$), where the shear stress is equal to the wall shear stress, which dampens out the velocity fluctuations. Therefore the viscous stresses are the main contributor, leading to a linear velocity profile which is given by

$$u^+ = y^+. \tag{2.79}$$

- The log-law region ($30 < y^+ < 200$), where viscous stresses are negligible compared to Reynolds stresses, resulting in a logarithmic velocity profile, given by the log law of von Kármán [75]

  $$u^+ = \frac{1}{\zeta}\ln(Ey^+), \tag{2.80}$$

  where the constant ζ is the Von Kármán constant and, based from experiments, is equal to $\zeta \approx 0.41$. The wall roughness parameter E is equal to $E \approx 9.8$ for smooth walls, and for rough walls, other values can be assigned.

- The buffer layer, lying in between the other layers, for which the velocity profile is not well defined.

The red curve in Fig. 2.1 is the actual flow velocity from a DNS simulation, which is the profile that needs to be captured. The blue and green curves are the standard wall functions. Notice that the horizontal y^+ axis is put in a logarithmic scale, hence the blue curve is actually linear, and the green line is actually logarithmic. The green curve, which represents the log-law wall function, proposed by Spalding, allows the wall-adjacent cell centroids to be in the log-law region when computational resources are very limited. The downside is that this is not accurate for all flow types, especially with large adverse pressure gradients leading to flow separation [76] . Another disadvantage is when the wall-adjacent cell centroid is in the viscous sublayer, the log-law wall functions lead to large errors. As for $y^+ < 5$, the velocity gradient is linear, and the variation across a cell is linear; the velocity gradient can be computed directly. Many CFD codes that apply the basic standard wall functions, such as in OpenFOAM, switch automatically between the functions where the blue and green curves intersect. This intersection is defined in OpenFOAM as y^+_{lam}, which is approximately at $y^+ = 11.5$.

As can be noticed in Fig. 2.1, even when switching between the wall functions at where they coincide, there is still a significant discrepancy between the prediction and the red curve when the wall-adjacent cells are in the buffer region. For a complex mesh, it is difficult and not guaranteed to have all the wall-adjacent cells outside the buffer region. A more accurate alternative is to deploy Spalding's wall function, which fits the Direct Numerical Simulation (DNS) curve through the entire y^+-range, and is given by:

$$y^+ = u^+ + \frac{1}{E}[e^{\zeta u^+} - 1 - \zeta u^+ - \frac{1}{2}(\zeta u^+)^2 - \frac{1}{6}(\zeta u^+)^3]. \tag{2.81}$$

The wall functions for the velocity profile are set in the wall boundary conditions for the eddy viscosity ν_t. The reason for that is that OpenFOAM actually does not modify the velocity at the wall adjacent cell, but makes a correction for the eddy viscosity when the cell centroid is outside the linear viscous sub-layer, in order to achieve the same wall shear stress, given by

$$\tau_w = \nu \frac{\partial u}{\partial y}\Big|_{\text{wall}}. \tag{2.82}$$

Numerically, the wall shear stress is calculated as follows:

$$\tau_w = \nu \frac{u_P - u_w}{y_P}, \tag{2.83}$$

where u_w is the wall velocity, which is equal to zero, and the subscript P denotes the centroid of the cell adjacent to the wall. Since the velocity profile is non-linear outside the viscous sub-layer, equation 2.83 will not capture the steep gradient in equation 2.82 if the cell centroid is in the log-law region. In order to achieve the same wall shear stress, relations 2.82 and 2.83 are equated. Substituting the relation of $y+$ (Eq. 2.78) and Eq. 2.80 into Eq. 2.83, will lead to

$$\tau_w = \nu \frac{\partial u}{\partial y}\Big|_{\text{wall}} = \frac{u_\tau u_P}{\frac{1}{\kappa}\ln(Ey^+)}. \tag{2.84}$$

By replacing ∂u and ∂y with u_P and y_P, the corrected viscosity can be found

$$\nu_w = \frac{u_\tau y_P}{\frac{1}{\zeta}\ln(Ey^+)}, \tag{2.85}$$

where ν_w is the corrected viscosity near the wall, which is the sum of the laminar viscosity ν (given) and the eddy viscosity ν_t.

Spalding's wall function is implemented in OpenFOAM's `nutUSpaldingWallFunction`, where ν_t is equal to zero in the viscous sub-layer and calculated as shown above outside the viscous sub-layer. With the calculated value for ν_t, and using Eq. 2.57, also α_{eff} is found.

Temperature The universal behaviour of the temperature distribution near the wall can also be derived from the log law, which can be used for heat transfer calculations. This thermal equivalent of Eq. 2.80 is

$$T^{+} \equiv -\frac{(T - T_w) c_p \rho u_\tau}{q_w} = Pr_t \left(u^{+} + P \left[\frac{Pr_t}{Pr} \right] \right), \tag{2.86}$$

where T_w is the wall temperature, and q_w is the wall heat flux. The variable P is the Jayatilleke pee-function, which is a correction function dependent on the ratio of laminar to turbulent Prandtl numbers and is determined from experimental data [77]

$$P = 9.24 \left(\left[\frac{Pr}{Pr_t} \right]^{3/4} - 1 \right) \left(1 + 0.28 e^{-0.007 Pr/Pr_t} \right). \tag{2.87}$$

The pee-function can also be derived analytically with the assumption of a certain distribution of turbulent viscosity and Pr_t over the viscous region [78]

$$P = 9.24 \left(\left[\frac{Pr}{Pr_t} \right]^{3/4} - \left[\frac{Pr}{Pr_t} \right]^{1/4} \right). \tag{2.88}$$

Thus, by knowing either the heat flux or temperature gradient normal to the wall, the other can be calculated.

Kinetic energy The kinetic energy at the wall boundary is found with OpenFOAM's `kqRWallFunction`. This function is actually a wrapper for the Neumann boundary condition, where its gradient is equal to zero. Fig. 2.2 graphically shows the development of the terms in the transport equation of the kinetic energy (Eq. 2.49) in the log-law region. We see that the production and dissipation of k balance each other, and the convective and diffusive terms are negligible. Therefore, it makes sense to restrict any additional fluxes at the wall to disrupt this balance and assign a zero gradient boundary condition for k.

However, in the viscous sub-layer, this theory is not correct, and therefore the `kqRWallFunction` requires that the wall-adjacent cell centroid to be in the log-law region. In the viscous sub-layer, the kinetic energy production diminishes, which does not balance the turbulence dissipation. Therefore, if the cell centroid falls within the sub-layer, k has to be computed.

Kalitzin et al. [80] proposed two equations that fit the DNS data k more accurately both in the sub-layer and in the log region, including a large part in the buffer region, as shown in Fig. 2.3. This adaptive wall function is implemented in OpenFOAM's `kLowReWallFunction`, where low Reynolds refers to the viscous sub-layer. The equations are as follows:

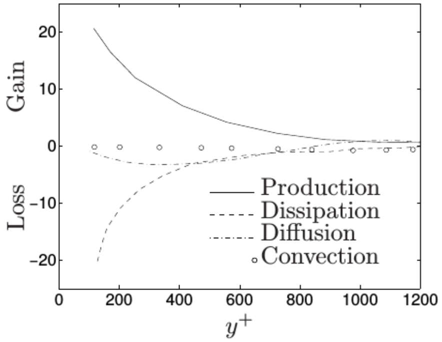

Figure 2.2: Boundary along a flat plate. Energy balance in k equation [79].

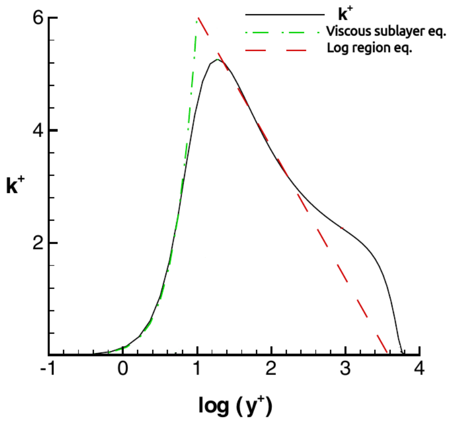

Figure 2.3: Numerical solution for k^+ vs. theoretical solution [80]

$$k^+ = \frac{2400}{C_{\varepsilon 2}^2}\left[\frac{1}{(y^+ + C)^2} + \frac{2y^+}{C^3} - \frac{1}{C^2}\right] \qquad y^+ \leq y^+_{\text{lam}}, \tag{2.89}$$

$$k^+ = \frac{C_k}{\zeta\sigma_k}\log(y^+) + B_k \qquad y^+ > y^+_{\text{lam}}, \tag{2.90}$$

where C, $C_{\varepsilon 2}$, C_k, σ_k, and B_k are model constants to fit the DNS-curve.

The turbulence based friction velocity (not to confuse with the shear-stress based friction velocity in Eq. 2.78!) is obtained by,

$$u_\tau = C_\mu^{1/4} k_P^{1/2}, \tag{2.91}$$

and finally, the value for k at the wall is then assigned as:

$$k_w = k^+ u_\tau^2. \tag{2.92}$$

Energy dissipation rate Unlike u, T or k, the wall function for the energy dissipation rate ε calculates the value for the wall-adjacent cell centroid rather than the face value at the wall. There is only one epsilon wall function, which by default is adaptive and contains two equations that fit the DNS-data both in the viscous sub-layer and the log region:

$$\varepsilon_P = \frac{2k_P\nu}{y^2} \qquad y^+ \leq y^+_{\text{lam}}, \tag{2.93}$$

$$\varepsilon_P = \frac{C_\mu^{3/4} k_P^{3/2}}{\zeta y_P} \qquad y^+ > y^+_{\text{lam}}. \tag{2.94}$$

2.6.3. CONJUGATION

To perform *conjugation* between the solution domains, two popular methods are mentioned in [81] :

1. The direct approach is grounded in solving the complete set of governing equations of both solid and fluid simultaneously, replacing both domains with one large single. To accelerate computation, an example is given of Pantakar's approach [82] where all dependent variables are described by a scalar ϕ, thus solving a general transport equation:

   $$\frac{\partial(\rho\phi)}{\partial t} + \nabla\cdot(\rho\boldsymbol{U}\phi) = \nabla\cdot(\Gamma\nabla\phi) + Q. \tag{2.95}$$

   To ensure that the velocity is zero at the wall and in the solid, one puts for the velocity field very large values of Γ in the grid points of the solid domain, and the value of the real fluid viscosity as Γ in the fluid domain. For the energy field, the real values of Γ are prescribed for both the fluid and solid. This ensures that the interface conditions (Eqs. 2.74 and 2.75) are met.

2. Another approach is the iterative method, where the equations of the fluid and solid domains are solved separately. In this approach, each domain solution provides a boundary condition along the interface for the other. The procedure begins by guessing a Dirichlet boundary condition for one of the conjugate quantities (typically the temperature) at the interface and solving the equations in the first domain (e.g., the fluid). This solution yields an updated value for the complementary interface condition (the heat flux), which is then imposed as the boundary condition for the second domain (the solid). Solving the solid domain with this flux produces a new estimate of the interface temperature, and the process is repeated. If the iteration converges, the procedure continues until the desired accuracy is achieved. However, the rate of convergence strongly depends on the initial guess, especially in transient simulations.

The conjugation in OpenFOAM is based on the iterative approach when using the `turbulentTemperatureRadCoupledMixed` boundary condition at the interface. This condition treats the interface as two coupled patches, e.g., one in the fluid region and one in the solid region. The solver will solve the fluid region first, after which the temperature field on the fluid interface patch is mapped on the solid interface patch and is updated using the heat flux:

$$T_{s,\text{int}} = T_{f,\text{int}} + q\frac{\delta}{\lambda_s}, \tag{2.96}$$

where δ is the *effective* numerical thickness used to approximate the temperature gradient across the interface. If there is no explicit thermal resistance or interface thickness (`thicknessLayers`) defined for δ in the boundary condition, OpenFOAM will use the following to ensure numerical stability:

$$\delta = \frac{1}{2}(\Delta x_f + \Delta x_s), \tag{2.97}$$

where x_f and x_s are respectively the distances from the interface face to the nearest cell centroids in the fluid region and solid region. Otherwise, $q = -\lambda \Delta T/\delta$ will be undefined as the physical distance between the interface patches is zero.

The final step of the iterative conjugate heat transfer process is to solve the heat equation in the solid region, after which the updated temperatures at the interface are mapped back into the fluid domain. This process is repeated until convergence.

2.6.4. INTERFACE CONDITION WITH THERMAL RADIATION

When adding radiation to the problem, the interface condition (Eq. 2.75) is altered using the `turbulentTemperatureRadCoupledMixed` condition:

$$-\lambda_f \frac{\partial T_f}{\partial n} + q_{r,\text{in}} + q_{r,\text{out}} = \lambda_s \frac{\partial T_s}{\partial n}, \tag{2.98}$$

where $q_{r,\text{in}}$ is the incident radiative heat flux (determined with the P1 or DOM model),

$$q_{r,\text{in}} = \int_{\vec{n}\cdot\vec{s}<0} I_s(\vec{s})|\vec{n}\cdot\vec{s}|d\Omega, \tag{2.99}$$

and $q_{r,\text{out}}$ is the radiative heat flux emitted and reflected by the solid surface

$$q_{r,\text{out}} = \frac{1}{\pi}\left[\epsilon_s \sigma T_s^4 + (1-\epsilon_s) q_{r,\text{in}}\right]. \tag{2.100}$$

2.6.5. EXTERNAL WALL HEAT LOSS

In order to have a more realistic thermal boundary condition at the outer wall surface, we incorporate heat loss to the environment by introducing a Robin boundary condition on the outer wall surface, using OpenFOAM's `externalWallHeat-FluxTemperature` function. It is assumed that there is no wind so that heat loss due to forced advection can be neglected (note that there is freedom to incorporate this as well, using the convective heat transfer coefficient). Referring to Fig. 2.4, the radiative heat loss is calculated as follows.

We define the inner wall temperature at y=0 as $T_{w,\mathrm{i}}$:

$$T|_{y=0} = T_{w,\mathrm{i}}. \tag{2.101}$$

At the outer wall surface (where y=1) we use the Stefan-Boltzmann law to determine the flux, such that

$$\lambda_s \frac{\partial T}{\partial y}\bigg|_{y=1} = q_{r,\mathrm{ext}} = \epsilon_w \sigma (T^4_{w,\mathrm{o}} - T^4_\infty), \tag{2.102}$$

where T_∞ is the ambient temperature, which is set to 288.15 K (ISA). Using Eqs. 2.101 and 2.102 yields the following relation for the outer wall surface temperature:

$$T_{w,\mathrm{o}} = \frac{\epsilon_w \sigma}{\lambda_s}(T^4_{w,\mathrm{o}} - T^4_\infty) y_1 + T_{w,\mathrm{i}}, \tag{2.103}$$

where y_1 is equal to the thickness of the refractory wall. Eq. 2.103 is valid when the interior surface area of the wall is equal to its outer surface area. In case of a rotary kiln, this is a valid assumption.

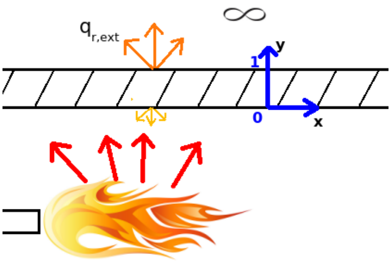

Figure 2.4: Picture of a confined turbulent flame with radiative heat loss to the environment.

3

NUMERICAL DISCRETISATION AND SOLUTION METHODS IN OPENFOAM

3.1. INTRODUCTION

The theory in this chapter partially follows Refs. [83, 84, 72, 85, 86]. Computational Fluid Dynamics (CFD) is a part of fluid mechanics that uses numerical analysis to solve advection-diffusion-reaction type equations related to a wide range of applications involving fluid flow and heat and mass transfer.

Discretisation is a technique in numerical analysis that transforms a continuous problem into a discrete counterpart, which can be solved numerically by means of numerical solution methods. It first requires that the geometry is divided into small, non-overlapping finite elements (or finite volumes), usually called cells. The discretisation method then transforms the partial differential equations into discrete algebraic equations, which are integrated over these discrete cells. The Finite Volume Method (FVM) is the most popular discretisation technique in CFD due to its main advantage of being strictly conservative. This is also the technique that is implemented in OpenFOAM. More about this in Sec. 3.2.

After the discretisation of each transport equation (e.g. Eqs. 2.43-2.46) the resulting set of discrete equations need to be linearised, which leads to a linear system of equations in the form of $\mathbf{Au} = \mathbf{b}$. The resulting systems of equations can be solved numerically by linear solvers, which can be grouped in general into direct or iterative methods. Sec. 3.3 describes the solution methods that are used in Open-

FOAM.

The time it takes for a solution to converge depends on the set of solution methods that is chosen for the linear systems of equations. The performance of the solution methods in OpenFOAM are studied in 3.4

Within OpenFOAM, many solvers are implemented to solve different sets of partial differential equations that the user wishes to model. At the beginning of the research, there was no single solver that could solve all the necessary physics described in Chapter 2. Therefore, a new solver is proposed that is a combination of two already existing solvers: one for turbulent combustion and the other for conjugate heat transfer. This is presented in Sec. 3.5.

3.2. FINITE VOLUME DISCRETISATION METHOD

The idea of the FVM is to construct a set of small sub-domains, called cells, which are treated as control volumes, and every governing equation is valid for each control volume. This set is often called a grid or mesh and is treated as a set of matrices on the computer. Control volumes can have arbitrary shapes, from tetrahedral to polyhedral. Each control volume in the mesh has values for computed fields (such as pressure, velocity, and temperature) at its centroid, which communicates with the centroid values of the neighbouring cells through the connecting faces. This is known as a cell-centerd finite volume method.

Recall from Chapter 2 the general form of the transport equation for an arbitrary scalar or vector variable ϕ , given by:

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{transient}} + \underbrace{\nabla\cdot(\rho\boldsymbol{U}\phi)}_{\text{advection}} = \underbrace{\nabla\cdot\Gamma\nabla\phi}_{\text{diffusion}} + \underbrace{Q_\phi}_{\text{source}} . \tag{3.1}$$

When using the FVM, the integral form of the transport equations within each control volume is considered. The integration covers the entire volume (V) of the control volume, and for transient equations, the entire time domain:

$$\frac{\partial}{\partial t}\int_V [\rho\phi]\, dV + \int_V [\nabla\cdot(\rho\boldsymbol{U}\phi)]\, dV = \int_V [\nabla\cdot\Gamma\nabla\phi]\, dV + \int_V [Q_\phi]\, dV. \tag{3.2}$$

Let’s first consider the steady-state case where the transient term is cancelled. The volume integrals of the advection, diffusion, and source terms are treated in the following subsections.

3.2.1. SOURCE TERM

The source term can be some constant Q or scalar function of e.g. ϕ, which may or may not depend (non-)linearly on the velocity. Then the approximation of the finite volume integration is as simple as multiplying its value at the cell centroid with the volume of the cell.

$$\int_V [Q]\, dV \approx Q_P V_P, \tag{3.3}$$

$$\int_V [Q_\phi]\, dV \approx Q_{\phi P} V_P, \tag{3.4}$$

$$\int_V [Q_\phi \boldsymbol{U}]\, dV \approx Q_{\phi P} \boldsymbol{U}_P V_P, \tag{3.5}$$

where P denotes the centroid of the local cell.

3.2.2. DIFFUSION TERM

Since the diffusion term is a flux acting on the surface of a control volume, the volume integral is transformed to a surface integral by using the Gaussian divergence theorem which states that

$$\int_V \nabla \cdot \boldsymbol{\Phi} dV = \int_S \boldsymbol{\Phi} \cdot \hat{\mathbf{n}} dS, \tag{3.6}$$

where $\boldsymbol{\Phi}$ is an arbitrary flux and $\hat{\mathbf{n}}$ the outward pointing normal vector at each point of the surface S. The surface integral is discretised into a sum over all faces,

$$\int_S \boldsymbol{\Phi} \cdot \hat{\mathbf{n}} dS = \sum_{f=1}^{m} \boldsymbol{\Phi}_f \cdot \mathbf{n}_f S_f, \tag{3.7}$$

where the subscript f indicates the faces of the cell consisting of m faces. The discretised diffusion term of Equation 3.2 is given as

$$\int_V \nabla \cdot (\Gamma \nabla \phi) dV = \int_S (\Gamma \nabla \phi) \cdot \hat{\mathbf{n}} dS = \sum_{f=1}^{m} \Gamma_f \nabla \phi_f \cdot \mathbf{n}_f S_f. \tag{3.8}$$

SPATIAL DISCRETISATION

Since we are integrating along the surface, we are interested in the value at the face centers, while the values of ϕ are stored at the cell centroids P and N (see Fig. 3.1).

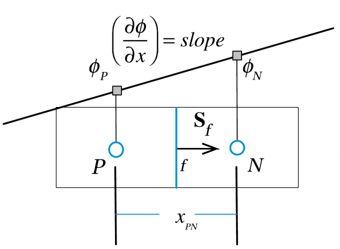

Figure 3.1: Interpolation profile at face f and slope of the gradient [85]

Therefore, the values at the face centers are obtained by linear interpolation. For uniform orthogonal cells, the face value ϕ_f is given by

$$\phi_f = \Psi\phi_N + (1-\Psi)\phi_P \qquad \Psi = \frac{x_{Pf}}{x_{PN}}, \tag{3.9}$$

where ϕ_P and ϕ_N denote the values of ϕ at the center points of respectively the local cell P and neighbouring cell N on the other side of the face, x_{PN} is the distance between those two cell centroids, x_{Pf} is the distance between the local cell centroid and the face center; and Ψ is the weighting factor. This is called **linear (or central) differencing** and it is the appropriate interpolation scheme for the gradient at the cell surface, which is linear by itself. This also applies for gradient terms such as the pressure gradient in the conserved momentum equation. Via a Taylor expansion, it can be shown that this interpolation scheme is second-order accurate, due to neglecting 2^{nd} and higher-order terms, which is considered very accurate for most CFD applications. In OpenFOAM this scheme is called Gauss linear, referring to the Gauss divergence theorem that was used to convert the volume integral of the diffusion term into a surface integral.

The diffusivity Γ at the cell surface center is also interpolated via the Gauss linear scheme:

$$\Gamma_f = \Psi\Gamma_N + (1-\Psi)\Gamma_P. \tag{3.10}$$

3.2.3. ADVECTION TERM

Similarly to the diffusion term in the previous section, the divergence theorem is applied to the advection term, which also acts on the control volume's surface. This leads to the discretised form

$$\int_V \nabla\cdot(\rho\mathbf{u}\phi)\,dV = \int_S (\rho\mathbf{u}\phi)\cdot\hat{\mathbf{n}}\,dS = \sum_{f=1}^{m}\phi_f(\rho\mathbf{u})_f\cdot\mathbf{n}_f S_f = \sum_{f=1}^{m}\phi_f\mathbf{F}_f, \tag{3.11}$$

where $\mathbf{F}_f = (\rho \boldsymbol{U})_f \cdot \mathbf{n}_f S_f$, is the advective flux at face f_i .

SPATIAL DISCRETISATION

The treatment of advective terms is one of the major challenges in CFD numerics and so the options are more extensive compared to the other terms.

THE CENTRAL DIFFERENCE SCHEME

At first sight, the obvious approach would be a linear interpolation profile similar to the one used for the diffusion term. This accurate scheme, however, is also **unbounded**, meaning that the interpolated face values are not guaranteed to remain within the range defined by the neighbouring cell-center values. As a result, overshoots and undershoots may occur, which can lead to oscillations, also known as numerical dispersion. This behaviour is illustrated in Fig. 3.2, where the value at the cell center exceeds the values of the neighbouring cells. In flows that are strongly diffusive this is not a problem, but when diffusion is overwhelmed by advection, the flow has a preferred direction, and the unbounded downstream cells lead to unphysical results, instability and even divergence in RANS simulations.

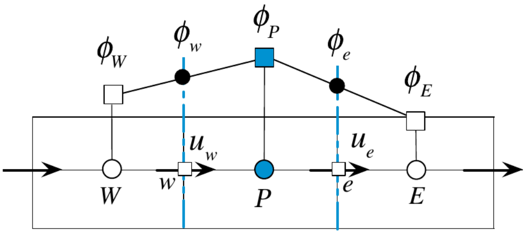

Figure 3.2: Profile of the central difference scheme, where E and W stands for the neighbouring cells at the east and west side of the local cell P [85]

THE UPWIND SCHEME

A scheme that is more compatible with the advection process is the upwind scheme. The upwind scheme basically mimics the basic physics of advection in the sense that the value at cell face f between cell centroids P and N depends on the direction of the mass flux ρu_f:

$$\phi_f = \begin{cases} \phi_P & \rho u_f > 0 \\ \phi_N & \rho u_f < 0 \end{cases} \tag{3.12}$$

As can be seen in Fig. 3.3, the value at the face is equal to the upwind nodal value, which for this case is local cell centroid P as the flow is moving from left to right.

The upwind scheme is considered the most stable advective discretisation scheme. However, since the value of ϕ no longer varies linearly between the cell centroid and face center, but is constant, this scheme is only first-order accurate. As a result, the flow is 'smeared out' by the upwind scheme, also known as numerical diffusion, and we lose a lot of resolution, which leads to inaccurate results.

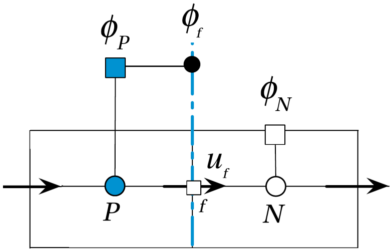

Figure 3.3: Profile of the upwind scheme [85]

Nevertheless, the upwind scheme is very robust and is often recommended as the initial scheme to get an initial solution before switching to a higher-order scheme.

THE LINEAR UPWIND SCHEME

The previous sections discussed that both the upwind and central difference schemes have severe limitations, the former because of its poor accuracy due to numerical diffusion, and the latter because of its instability, also known as numerical dispersion error. A lot of research had been done to improve the accuracy and stability of advection schemes by using higher-order upwind biased interpolation profiles. These higher-order schemes aim at producing at least a second-order accurate solutions, while being unconditionally stable.

The linear upwind scheme is one such type of scheme and one of the most popular ones. It has a linear profile, as was the case with the central difference scheme. However, instead of a symmetric profile, an upwind biased stencil is used,

$$\phi_f = \begin{cases} \phi_P + \nabla\phi_P \cdot r & \rho u_f > 0 \\ \phi_N + \nabla\phi_N \cdot r & \rho u_f < 0 \end{cases} \tag{3.13}$$

where r is the distance between the cell centroid P (or neighbour centroid N) and cell face f. As depicted in Fig. 3.4, the linear profile is constructed by employing the ϕ values at nodes P and W to obtain $\nabla\phi_P$. Therefore, the value at the face is actually calculated by extrapolation rather than interpolation.

Although the linear upwind scheme is an accurate scheme, it is still prone to instability when very high values of velocity gradients occur in the flow or when the

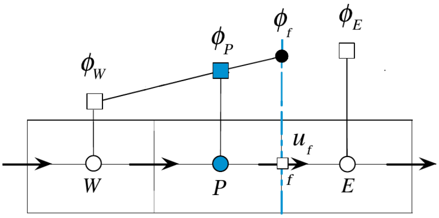

Figure 3.4: Profile of the upwind scheme [85]

initial internal flow field is far from the solution. As mentioned before, the remedy for this is to run simulations from a flow field that is already solved with the upwind scheme as a starting guess. The second solution is to apply a gradient limiter to limit $\nabla\phi_P$.

Other higher-order upwind-biased schemes that can be used in OpenFOAM are the limited linear scheme, the linear-upwind stabilised transport scheme, the QUICK scheme [87] and others [86], which either utilise a limiter or blending of the linear and upwind schemes, but these fall out of the scope of this dissertation.

3.2.4. TRANSIENT TERM

Transient simulations add a new dimension to the problem, and therefore the discretisation of the transient term is required. By substituting all the spatial discretised terms into the general transport equation (Eq. 3.1) and rearranging such that the transient term is on the LHS, the time integration of the transport equation becomes:

$$\int_{t-\Delta t/2}^{t+\Delta t/2} \left(\frac{\partial \rho\phi}{\partial t}\right) V_P \, dt = \\ - \int_{t-\Delta t/2}^{t+\Delta t/2} \left(\sum_{f=1}^{N} \phi_f \mathbf{F}_f - \sum_{f=1}^{N} \Gamma_f \nabla\phi_f \cdot \mathbf{n}_f S_f - Q_{\phi P} V_P \right) dt, \quad (3.14)$$

where the RHS term can be expressed by a spatial discretisation operator $L(\phi_C^t)$ at some time t, such that that Eq. 3.14 becomes

$$\int_{t-\Delta t/2}^{t+\Delta t/2} \left(\frac{\partial \rho\phi}{\partial t}\right) V_P \, dt = - \int_{t-\Delta t/2}^{t+\Delta t/2} L(\phi_C^t) \, dt. \quad (3.15)$$

FIRST-ORDER EULER IMPLICIT SCHEME

The transient first-order implicit Euler scheme is obtained similarly as the first-order upwind scheme of the advective term. The difference is that the transient term is integrated over the temporal domain, which is visualised vertically in Fig. 3.5. The temporal grid does not represent a geometry; rather, the temporal cell centroids represent the stored spatial field variables per time step. By using a first-order Euler implicit interpolation profile, the value of $\rho\phi$ at the temporal element face is set equal to the value at the centroid of the upwind element. Which means that $(\rho_P\phi_P)^{t+\Delta t/2}$ is set to $(\rho_P\phi_P)^t$, and $(\rho_P\phi_P)^{t-\Delta t/2}$ is set to $(\rho_P\phi_P)^{t-\Delta t}$. After integrating Eq. 3.15, the time discretisation becomes:

$$\frac{(\rho_P\phi_P)^t - (\rho_P\phi_P)^{t-\Delta t}}{\Delta t} = -L(\phi_P^t), \tag{3.16}$$

where the spatial operator is evaluated at the current time step t.

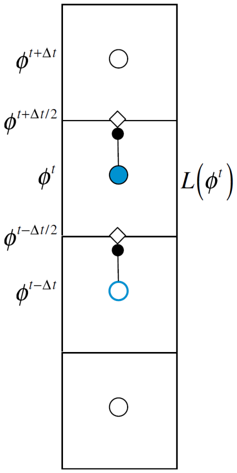

Figure 3.5: Profile of the first-order Euler implicit scheme along the time 'grid' [85]

NUMERICAL DIFFUSION

The implicit Euler scheme is unconditionally stable, but as it is first-order accurate, it generates numerical diffusion that scales with the time step.

Just like the central difference scheme for the advective term, the temporal discretisation can be obtained with a second-order accurate linear profile between the cell centroids of the previous and following time step. This is known as the **Crank-**

Nicolson scheme and leads to following temporal discretisation

$$\frac{(\rho_P\phi_P)^t - (\rho_P\phi_P)^{t-\Delta t}}{\Delta t} = -\frac{1}{2}\left[L(\phi_P^t) + L(\phi_P^{t-\Delta t})\right]. \tag{3.17}$$

This is a more accurate discretisation, but its stability is constrained by the low time step. Therefore, similar to the central difference scheme, it is advisable to run a simulation with the Euler implicit scheme first, followed by the Crank-Nicolson. In OpenFOAM, the Crank-Nicolson method is typically blended with the Euler implicit with a blending factor from zero to one, where the value 1 corresponds to pure Crank-Nicolson.

Most of the simulations in this thesis are steady-state simulations, where the transient term is excluded. However, problems that involve combustion with detailed chemistry need to be resolved in time as well.

CONSTRAINED BY THE LOW CFL CONDITION

In order to have a stable transient simulation of a reacting flow it is necessary to have the maximum Courant number stay far below 1, as chemical time scales are much lower than flow time scales. The Courant number in its general form is defined as

$$Co = \frac{U \cdot \Delta t}{\Delta x} \tag{3.18}$$

where Δx is the cell size. This value determines how many time steps are required for the flow to pass through a cell. Assigning a maximum Courant number is known as the Courant–Friedrichs–Lewy (CFL) condition. OpenFOAM allows adjusting the global time step continuously, which is determined by the maximum Courant number at the most critical cell:

$$\Delta t^{n+1} = \min\left(\frac{Co_{max} \cdot \Delta x_i^n}{U_i^n}\right), \tag{3.19}$$

where i is the index of the local cell. The global time step is used throughout the entire computational domain for the actual iteration. The main disadvantage of this approach is that the smallest cell with the highest velocity determines the speed of the simulation's progress.

For rotary kiln simulations this is problematic due to the sheer size of the kiln, where the flow residence times can be longer than 30 seconds. A CFL-condition of <0.2 requires that the time steps should be of the order 10^{-8} seconds, due to the limitation of the fuel inlets where the inlet speed reaches Mach 0.6 and the cell size is less than one mm. This leads to extremely long run times, even for URANS.

LOCAL TIME STEPPING (LTS)

Instead of calculating the global maximum time step, the LTS method assigns at each cell a local time step by using the global maximum Courant number

$$\Delta t_i^{n+1} = \frac{Co_{max} \cdot \Delta x_i^n}{u_i^n}, \tag{3.20}$$

so that a pseudo time advancement can be achieved that is not constrained by the CFL condition in the smaller cell. This way, the solution is accelerated to steady state because a time step "field" is computed, so each cell has its own time step instead of one global time step. Information in the flow field is now transported much faster through the domain. The LTS approach can be used if a steady-state solution exists and if you are not interested in the intermediate steps. Before reaching the steady-state the solution is most likely invalid and possibly even unphysical [86].

The time step differences across neighbouring cells can be large and cause instability. So there are several smoothing and damping parameters in OpenFOAM to manage the LTS procedure in order to maintain stability.

ALPHATEMP

This parameter limits the heat release in a cell. Similar to an under-relaxation factor for the energy variable. A value of 1.0 sets the new value completely; a value of 0.5 blends between 50% of the previous value and 50% of the new value. This is a useful parameter in simulations with chemical reactions.

ALPHAY

Similarly to alphaTemp, alphaY limits the chemical reaction rate. It also uses a value between 0 and 1 to blend the previous and new values.

RDELTATSMOOTHINGCOEFF

This parameter spatially smooths the time scale field by ensuring the values in neighbouring cells are at least rDeltaTSmoothingCoeff multiplied with the cell value. This is to avoid abrupt transitions from small local time steps to large local time steps in neighbouring cells in the field which will lead to spikes. This is in particular useful for meshes with large local differences in cell sizes. A value of 1.0 does not perform any smoothing, so each cell has its own time step fitting the maximum Courant number. A value of zero will use a unique time step for the complete domain.

RDELTATDAMPINGCOEFF

While rDeltaTSmoothingCoeff is a spatial limiter, the rDeltaTDampingCoeff limits the time rate of change. The changes of the time step field between iterations are damped so that time increases only with a fraction of the old time step.

MAXDELTAT

This parameter limits the maximum allowed time step. In cells where the flow hardly moves, the time step can rise to infinity, so a limitation is required. However, this value should be really large to take the best out of the LTS approach.

Due to the high numbers of parameters that can be tuned with the LTS method, finding the optimal values for each of them, which result in stable yet accurate simulations, can be challenging. This line can be very thin. How the parameters need to be set up is different for every problem that needs to be solved, which is a trial-and-error procedure and requires some experience, for the sake of accelerating transient simulations.

3.2.5. DECOUPLING THE VELOCITY AND PRESSURE

Once the velocity field is determined, solving the transport of a particular scalar (such as enthalpy and chemical species) is relatively easy, as the flow field of that scalar is ’frozen’, and the main challenge is usually to determine the source terms. However, to calculate the velocity field requires solving the discretised Navier-Stokes equations numerically, which is difficult. Not necessarily because it is non-linear, but due to the coupling of the velocity and pressure in the momentum equation. Consider again the incompressible steady-state Navier-Stokes equations with neglected gravity

$$\nabla \cdot \boldsymbol{U} = 0, \tag{3.21}$$

$$\boldsymbol{U} \cdot \nabla \boldsymbol{U} = -\nabla \frac{p}{\rho} + \nabla \cdot \nu \nabla \boldsymbol{U}, \tag{3.22}$$

consisting of 4 equations and 4 unknowns: U_x, U_y, U_z and p (note that the density here is not a variable but a reference value). This would intuitively mean that it is directly solvable, even if there is no explicit equation for p. However, the continuity equation is not really the ’closing’ equation, but acts as a restriction to the momentum equations in order to satisfy mass conservation. And since we are considering incompressible and isothermal flow, the equation of state cannot be used to calculate the pressure.

The following algorithms are adopted in this thesis to tackle this issue related to the velocity-pressure coupling: the **Semi-Implicit Method for Pressure-Linked Equations** (SIMPLE) and the **Pressure Implicit with Split Operator** (PISO). Both methods are based on decoupling the pressure and velocity fields, allowing for calculating them separately, then using one field to correct the other field iteratively until convergence is achieved.

THE SIMPLE ALGORITHM

The SIMPLE algorithm is developed by Patankar [82] to solve the steady-state incompressible Navier-Stokes equations. The main idea is to use the continuity equation to derive an equation for the pressure after splitting it from the momentum equations. The pressure equation will then act as a corrector for the modified momentum equation. It starts by by writing the momentum equations (Eq. 3.22) in general matrix form:

$$\boldsymbol{A}\boldsymbol{U} = -\nabla p, \tag{3.23}$$

where $\boldsymbol{A}$ is the coefficient matrix resulting from discretising Eq. 3.22 using the Finite Volume Method. All these coefficients are known. Eq. 3.23 can be solved when the pressure is known from the previous iteration (starting from an initial guess at the first iteration), of which its gradient acts as the source term here. Also, the non-linearity appearing in the LHS of Eq. 3.22 is resolved by computing one the velocities, which acts as the flux term, from the velocity of the previous iteration. The LHS of Eq. 3.23 is known in OpenFOAM as the velocity equation. The calculated velocity field does not yet satisfy continuity and is still a guess. This stage is called the momentum-predictor.

The derivation of the pressure equation starts from splitting the coefficient matrix $\boldsymbol{A}$ by extracting its diagonal into matrix $\boldsymbol{D}$:

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{D}\boldsymbol{U} - \boldsymbol{H}. \tag{3.24}$$

The reason is because a diagonal matrix can easily be inverted ($\boldsymbol{D}^{-1}$) which is very useful. $\boldsymbol{H}$ is the residual matrix containing the coefficients of all the neighbour cells, and will be used as a source term for the pressure equation later. Combining Equations 3.23 and 3.24, and multiplying on both sides with $\boldsymbol{D}^{-1}$ results in an equation for the velocity field which is an explicit function of the pressure field:

$$\boldsymbol{U} = \boldsymbol{D}^{-1}\boldsymbol{H} - \boldsymbol{D}^{-1}\nabla p. \tag{3.25}$$

In the OpenFOAM source code, the diagonal matrix $\boldsymbol{D}$ is defined as `rAU`, and the frequently recurring product $\boldsymbol{D}^{-1}\boldsymbol{H}$ is known as `HbyA`. Now we can use the continuity equation to derive an explicit equation for the pressure. Substituting Eq. 3.25 into Eq. 3.21 leads to following Poisson equation, which is known as the pressure corrector:

$$\nabla \cdot (\boldsymbol{D}^{-1}\nabla p) = \nabla \cdot (\boldsymbol{D}^{-1}\boldsymbol{H}). \tag{3.26}$$

After solving for p in the above pressure equation, the velocity field (Eq. 3.25 can be corrected to satisfy continuity. Notice that the velocity is calculated explicitly and this is why this method is called Semi-Implicit. The problem now is that, once the velocity is corrected, the pressure equation is no longer satisfied because the matrix $\boldsymbol{H}$ depends on the velocity which has been updated. Therefore this process is iterated until both the velocity and pressure fields no longer change after correction.

The SIMPLE algorithm can be extended to problems beyond steady incompressible flows. In such cases, most transport variables such as chemical species and enthalpy are solved after the momentum predictor step in OpenFOAM, and the turbulence transport equations are solved after the pressure corrector step. An example algorithm of a multi-physics solver is shown in Fig. 3.23.

SOLVING FOR CHECKERBOARD OSCILLATIONS IN OPENFOAM

As OpenFOAM uses collocated grids (all field variables stored on the cell centroids), the standard SIMPLE algorithm, in combination with the linear interpolation of the velocity flux and pressure gradient on the cell faces, will lead to the velocity term at the current cell centroid not being coupled with the pressure term at the current cell centroid in the discretised momentum equation. Instead, the momentum equation in the local cell centroid will depend on the pressure terms of the neighbouring cells. This is known as the checkerboard effect.

OpenFOAM solves this problem with a flux operator that follows the ’spirit’ of Rhie and Chow interpolation[88], and is described in detail in [89]. The flux operator corrects the velocity flux at the cell faces with the pressure gradient to ensure that the pressure at the current cell centroid is cast into the momentum equation at the cell centroid.

THE PISO ALGORITHM

This algorithm is developed by Issa [90] and is a modification of the SIMPLE algorithm, intended for unsteady incompressible flows. The loop of the SIMPLE algorithm is shown again in the equations below in consecutive order, and is also known as the ’outer’ loop.

- Momentum predictor

$$\boldsymbol{AU} = -\nabla p, \tag{3.27}$$

- Construct the $\boldsymbol{H}$-matrix

$$\boldsymbol{H} = \boldsymbol{DU} - \boldsymbol{AU}, \tag{3.28}$$

- Solve the pressure equation

$$\nabla \cdot (\boldsymbol{D}^{-1} \nabla p) = \nabla \cdot (\boldsymbol{D}^{-1} \boldsymbol{H}), \tag{3.29}$$

- Correct the velocity field with the updated pressure

$$\boldsymbol{U} = \boldsymbol{D}^{-1} \boldsymbol{H} - \boldsymbol{D}^{-1} \nabla p. \tag{3.30}$$

This algorithm can, in principle, be used to solve transient flows as well. The problem is however that the algorithm would proceed to the next time step once the outer loop converges, which similarly to steady-state problems can be thousands of iterations per time step. Especially when the time steps are small, this is computationally too expensive.

The PISO algorithm follows the same steps as the SIMPLE algorithm, except that it will calculate the momentum predictor once in every time step and use the flux-corrected $\boldsymbol{U}$ to update $\boldsymbol{H}$ directly. Therefore if performs 'inner' loops from Eq. 3.28 till Eq. 3.30. The inner loop ensures that the pressure equation converges and is a faster way to proceed to the next time step. However, as the momentum equation is not updated at the end of each inner loop, the algorithm is less stable and can lead to divergence of the momentum equation. This can be counteracted when the time step size is small enough, as the transient term $\Delta U / \Delta t$ becomes large and is located in the diagonal of matrix $\boldsymbol{AU}$ (cell centroid), which increases the diagonal dominance, similarly to what is achieved with under-relaxation factors in the SIMPLE algorithm.

If $Co < 1$, the diagonal dominance is strong such that a typical time step iteration can be performed with one momentum predictor step and two 'inner' pressure corrector steps that converge partially.

3.3. SOLUTION METHODS

3.3.1. INTRODUCTION

For each transport equation, the discretisation and linearisation process leads to a system of linear equations, which is written as the following general matrix equation:

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{3.31}$$

where **A** is the coefficient matrix resulting from the linearisation and mesh geometry, **u** is the vector that contains the unknown values of the dependent variable at the cell centroids which needs to be solved for, while **b** is the vector containing all the sources, constants and boundary conditions.

With the momentum-predictor in the previous section (Eq. 3.27), it is easy to see how it is translated into a general matrix equation:

- $\boldsymbol{A} \quad \equiv \mathbf{A}$,
- $\boldsymbol{U} \quad \equiv \mathbf{u}$,
- $-\nabla p \quad \equiv \mathbf{b}$,

while in the pressure-corrector (Eq. 3.29) the translation is a bit less apparent:

- $\boldsymbol{D}^{*} \quad \equiv \mathbf{A}$,
- $p \quad \equiv \mathbf{u}$,
- $\nabla \cdot (\boldsymbol{D}^{-1}\boldsymbol{H}) \quad \equiv \mathbf{b}$,

where $\boldsymbol{D}^{*}$ is the resulting coefficient matrix after combining $\boldsymbol{D}^{-1}$ with the discretised Laplace operator coefficients of p.

The linear systems are solved by linear solvers, for which different solution methods exist, which are generally grouped into *direct* and *iterative* methods, each consisting of many sub-groups. Due to the linearisation process with the finite volume method, the coefficient matrix is very large (many cells required for accuracy), very sparse, and diagonal dominant. Iterative methods therefore have been more popular because they are more suited for this type of applications as they typically require lower computational cost per iteration and lower memory. The (modified) direct methods can, however, be used as a preconditioner, in which they are used to replace the coefficient matrix by a matrix for which the corresponding linear system of equations is easier to solve, while keeping the same solution. The solution methods that are used in OpenFOAM are discussed here.

3.3.2. DIRECT METHODS

Even though direct methods, as stand-alone solvers, are not efficient for solving sparse systems of linear equations due to their high computational cost, their discussion will lead the way for introducing efficient iterative methods in the next sections.

Direct methods apply some form of Gaussian elimination techniques to solve the above mentioned system of linear equations. The modern version is the **LU factorisation**, which computes the lower and upper triangular matrix **L** and **U**, such that

$$\mathbf{A} = \mathbf{LU}. \tag{3.32}$$

For diagonal dominant matrices, the upper triangular matrix **U** is constructed by performing Gaussian elimination on matrix **A**, while the elements of **L** consist of the Gauss factors by which the rows of **A** are multiplied to get **U**. Furthermore, the diagonal elements of **L** are set equal to one, so for a 3x3 matrix, Eq. 3.32 becomes

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}. \tag{3.33}$$

When **L** and **U** are found, the problem $\mathbf{Au} = \mathbf{b}$ is split in two problems that are much easier to solve

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Uu} = \mathbf{y}. \tag{3.34}$$

To avoid rounding errors it might be necessary to apply some form of pivoting, i.e. reordering the rows of matrix **A**. However for diagonal dominant matrices this is not necessary [91]. If **A** is Symmetric Positive Definite (SPD) the LU-factorisation reduces to the **Cholesky factorisation** of **A**.

$$\mathbf{A} = \mathbf{CC}^{\mathsf{T}}, \tag{3.35}$$

Where **C** is a lower triangular matrix. This results to even more memory and compuational savings as only one triangular matrix needs to be computed.

The efficiency of the direct methods however is lost due to the occurrence of *fill-in* after factorisation, where many zero entries in the sparse matrix become nonzero during the factorisation process. This significantly increases the memory requirements and computational cost, thereby disrupting the advantages of sparse matrix storage, especially for large practical CFD applications. This leaves an opening for iterative methods.

It will be shown that direct methods are very suitable as preconditioners for the Krylov subspace methods where the sparsity pattern is deliberately kept after factorisation, known as **incomplete factorisation**, so that $\mathbf{A} \approx \mathbf{LU}$ or $\mathbf{A} \approx \mathbf{CC}^{\mathsf{T}}$.

3.3.3. BASIC ITERATIVE METHODS

The linear solvers in most CFD codes make use of iterative methods. Iterative methods are techniques created to obtain an approximate solution of linear systems. For the implementation of these methods, successive approximations are used. Starting from an initial guess $\mathbf{u_0}$, a new approximation $\mathbf{u_k}$ is obtained at each iteration k, until an approximated solution is found that is close enough to the exact solution $\mathbf{u}$.

The error vector is defined as

$$\mathbf{e}_k = \mathbf{u} - \mathbf{u}_k. \tag{3.36}$$

Solving the error vector is just as difficult as solving the exact solution of the discrete linear system $\mathbf{Au} = \mathbf{b}$. A computable measure of the quality of the approximation is therefore obtained from the residual,

$$\mathbf{r}_k = \mathbf{b} - \mathbf{Au}_k, \tag{3.37}$$

which represents the *imbalance* of the approximation of the conservation laws, with $\mathbf{r}_k = 0$ being the exact solution. A common stopping criterion or tolerance (ϵ) for iterative methods is the relative residual, defined as the 2-norm of the residual of the k-th iteration divided by the 2-norm of the right-hand side,

$$r_k = \frac{||\mathbf{r}_k||_2}{||\mathbf{b}||_2} \leq \epsilon. \tag{3.38}$$

The idea of an iterative method is that the matrix $\mathbf{A}$ is decomposed into two matrices, $\mathbf{M}$ and $\mathbf{N}$. Such that $\mathbf{A} = \mathbf{M} - \mathbf{N}$; and the original linear system $\mathbf{Au} = \mathbf{b}$ transforms into:

$$\mathbf{Au} = (\mathbf{M} - \mathbf{N})\mathbf{u} = \mathbf{b}. \tag{3.39}$$

rearranging terms we obtain:

$$\mathbf{Mu} = \mathbf{Nu} + \mathbf{b} = (\mathbf{M} - \mathbf{A})\mathbf{u} + \mathbf{b}. \tag{3.40}$$

The latter system is used to perform an iterative process, finding at each iteration (k) a more accurate solution. Most of the iterative methods are derived from the following recurrence relation:

$$\mathbf{u}_{k+1} = \mathbf{M}^{-1}(\mathbf{M} - \mathbf{A})\mathbf{u}_k + \mathbf{M}^{-1}\mathbf{b}, \tag{3.41}$$

Or after rearranging and using Eq. 3.37

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \mathbf{M}^{-1}\mathbf{r}_k, \tag{3.42}$$

where the matrix $\mathbf{M}$ is chosen such that $\mathbf{M}^{-1}$ can be determined easily, so for example a diagonal or triangular matrix.

Basic iterative methods (BIM) are obtained by decomposing the system matrix as $\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F}$ (see Fig. 3.6) . $\mathbf{D}$ being the diagonal of $\mathbf{A}$, and $\mathbf{E}$ and $\mathbf{F}$ are the strictly lower and upper parts. Based on the choice of $\mathbf{M}$ and $\mathbf{N}$, different iterative methods can be obtained. Some of the BIMs are presented in Table 3.1, of which Jacobi and Gauss-Seidel are the options for OpenFOAM's BIM solver `smoothSolver`.

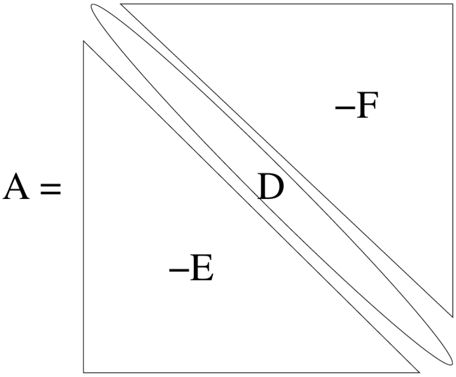

Figure 3.6: Splitting of the coefficient matrix A

Table 3.1: Basic iterative methods.

Method	M	N	Iteration
Richardson	$\mathbf{I}$	$\mathbf{I}-\mathbf{A}$	$u_{k+1} = (\mathbf{I}-\mathbf{A})\mathbf{u}_k + \mathbf{b}$
Jacobi	$\mathbf{D}$	$\mathbf{E}+\mathbf{F}$	$u_{k+1} = \mathbf{D}^{-1}(\mathbf{E}+\mathbf{F})\mathbf{u}_k + \mathbf{D}^{-1}\mathbf{b}$
Damped Jacobi	$(1/\omega)\mathbf{D}$	$\mathbf{E}+\mathbf{F}$	$u_{k+1} = \omega\mathbf{D}^{-1}(\mathbf{E}+\mathbf{F})\mathbf{u}_k + \omega\mathbf{D}^{-1}\mathbf{b}$
Gauss-Seidel	$\mathbf{D}-\mathbf{E}$	$\mathbf{F}$	$u_{k+1} = \mathbf{D}^{-1}(\mathbf{F}\mathbf{u}_k + \mathbf{E}\mathbf{u}_{k+1}) + \mathbf{D}^{-1}\mathbf{b}$
Successive Over-Relaxation	$\mathbf{D}-\omega\mathbf{E}$	$(1-\omega)\mathbf{D}+$ $\omega\mathbf{F}$	$u_{k+1} = \omega\mathbf{D}^{-1}(\mathbf{F}\mathbf{u}_k + \mathbf{E}\mathbf{u}_{k+1})+$ $(1-\omega)\mathbf{u}_k + \omega\mathbf{D}^{-1}\mathbf{b}$

Out of presented methods, the Successive Over-Relaxation method is significantly faster than the others for optimal damping parameters ω, which value is not straightforward to find. Nevertheless, BIMs in general are too slow for engineering problems as stand alone solvers, as they are only effective in eliminating the high frequency components of the error, but very slow with reducing the low frequency components (Fig, 3.7). This gets worse for finer meshes.

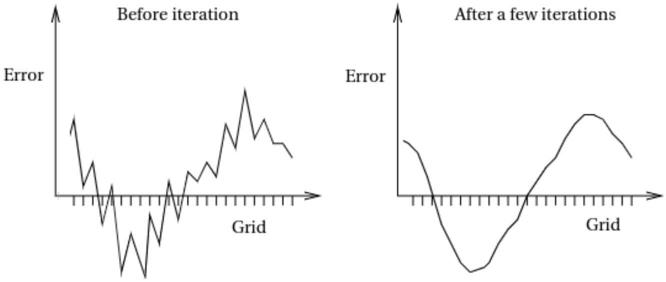

Figure 3.7: Smoothing process of a BIM

Nevertheless, it is the ability of eliminating the high frequency error components (smoothing) why the BIMs serve as the second set of building blocks for more efficient iterative methods, in addition to the direct methods. As smoothers for the multigrid method or as preconditioners for the Krylov subspace methods.

3.3.4. MULTIGRID METHODS

The convergence of a BIM can be accelerated with a multigrid method, which is known to be among the most efficient solvers for elliptic problems [92, 93] , such as the steady-state transport equation

$$\underbrace{\nabla\cdot(\rho\mathbf{U}\phi)}_{\text{advection}} = \underbrace{\nabla\cdot\Gamma\nabla\phi}_{\text{diffusion}} + \underbrace{Q_{\phi}}_{\text{source}} \, . \tag{3.43}$$

The basic idea of a multigrid method is to first smooth the error with a BIM, then transferring the remaining low frequency error components (recursively) to a coarser grid without losing information, until the problem is small enough to solve directly, after which a correction can be transferred back to the fine grid. As a result, the number of iterations required for convergence becomes nearly independent of the mesh element size, and the total computational cost scales approximately linearly with the number of unknowns. This high efficiency explains why multigrid methods are widely used in many CFD codes [72, 94, 95] . Depending on the size of the 2D or 3D mesh, the speed-up can be one or multiple orders of magnitude as compared to the BIM [84].

The detailed procedure is a follows:

1. **Presmoothing** - Starting from an initial guess $\mathbf{u}_0$, apply a few number ν_1 of iterations with a BIM to eliminate the high frequency error components (recall from Eq. 3.42).

$$\mathbf{u}_n^h = \mathbf{u}_{n-1}^h + \mathbf{M}^{-1}\mathbf{r}_{n-1}^h, \tag{3.44}$$

where $n = 1, .., \nu_1$ and the superscript h denotes the fine grid size.

2. **Restriction** - To transfer the residual to a coarser grid of grid size H, we use the *restriction* operator I_h^H

$$\mathbf{r}^H = I_h^H \mathbf{r}^h, \tag{3.45}$$

where superscript H denotes the coarse grid size.

3. **Solve coarse grid error** - The residual can now be used to determine the coarse grid error by combining Eqs. 3.36 and 3.37

$$\mathbf{A}^H \mathbf{e}^H = \mathbf{r}^H. \tag{3.46}$$

This is known as the *residual equation*. As mentioned before, solving this equation is just as hard as solving the original system of linear equations. However, at the current grid level the problem is small enough to solve it directly and the obtained error vector can be used as an approximation at the fine grid in the following step.

4. **Prolongation** - The error e^H is projected into the fine grid using the *prolongation* operator I_H^h (inverse of I_h^H), and the vector $u_{\nu_1}^h$ is corrected with the approximated error.

$$\mathbf{u}_{\nu_1}^h = \mathbf{u}_{\nu_1}^h + I_H^h e^H \tag{3.47}$$

5. **Postsmoothing** - Apply a few number of postsmoothing ν_2 iterations as described in step 1 to obtain $\mathbf{u}_1 =: \mathbf{u}_{\nu_1+\nu_2}^h$

In the above procedure, the sequence described by steps 2 to 4 is known as the *defect correction*, or *coarse grid correction* (CGC) and can be summarised by the following equation

$$\mathbf{u}_n^h = \left(\mathbf{I} - I_H^h (\mathbf{A}^H)^{-1} I_h^H \mathbf{A}\right) \mathbf{u}_{n-1}^h. \tag{3.48}$$

Although the BIM and the CGC are individually not efficient, the combination of the two as applied above is very efficient as they effectively eliminate all the frequency components of the error. The described procedure above is essentially a two-grid cycle when going through the steps once. When steps 1-2 (hence also steps 4-5) are done recursively within one loop we arrive at the multigrid method. Most of the times this is necessary as the long wavelengths of the error modes become shorter in the coarser mesh and therefore presmoothing is needed to eliminate them quickly. Applying a certain number of the restriction-step will lead to a negligible cost of solving the residual equation compared to a smoothing sweep at the finest grid.

Applying step 3 only once is called a V-cycle, and applying it multiple times can result in a W-cycle or an F-cycle, which are more accurate but require also more work. A schematic is shown in Fig. 3.8. There are many strategies and choices to tune the multigrid method, such as the amount of smoothing sweeps, grid-levels, and cycles, with varying trade-offs between speed of solving a single iteration and the overall rate of convergence.

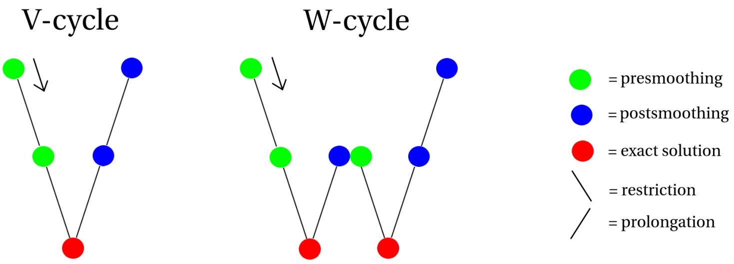

Figure 3.8: Different multigrid cycles starting from the finest grid at the top, towards the coarsest grid at the bottom and back.

One of the approaches to construct the coarser grid is by applying the restriction operator directly to the system matrix, which is known as the **algebraic multigrid** (AMG). This method is relatively easy to implement, but works as a black-box solver as it does not have any geometric interpretation of the mesh. The **geometric multigrid** (GMG) on the other hand obtains the coarser grid by clustering (or agglomerating) the cells of the mesh.

OpenFOAM works with the **generalised method of geometric-algebraic multigrid** (GAMG) which by default operates as a GMG solver, but also gives the option to enable the older AMG implementation (of its predecessor) instead [96]. The GMG solver agglomerates the cells by pairing each cell with an unpaired neighbour cell with the largest shared face area. This is repeated for every coarse grid level. An examples is shown in Fig. 3.9. If a cells does not have a match, it will be added to the neighbouring group. The system matrix is adjusted accordingly.

The GAMG is a V-cycle method where the user can select a smoother (Gauss Seidel being most favoured) and adjust certain parameters, such as the mesh size at the coarsest level, the amount of presmoothing and postsmoothing sweeps. The solver calculates automatically the amount of intermediate coarse grid levels. A practical example of how the GAMG solver is set up is shown in Sec. 3.4.

Next to multigrid methods, another class of solvers called Krylov subspace methods can be adopted.

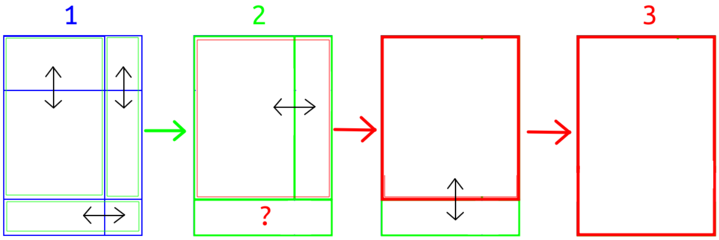

Figure 3.9: Geometric agglomeration using face pairs in three coarse grid levels 1, 2 and 3.

3.3.5. KRYLOV SUBSPACE METHODS

In this section a few other solvers are presented that are currently used in OpenFOAM, which are also one of the most efficient methods for solving large linear systems [83]. These methods are based on projection processes onto Krylov subspaces $\mathscr{K}_i$, that is

$$\mathscr{K}_i(\mathbf{A},\mathbf{r}_0) = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, ..., \mathbf{A}^{i-1}\mathbf{r}_0\}, \tag{3.49}$$

where $\mathscr{K}$ is called the Krylov space of dimension i corresponding to matrix $\mathbf{A}$ and initial residual $\mathbf{r}_0$ (Eq. 3.37). The basic idea of Krylov subspace methods is that, for large and sparse matrices, it is more efficient to use $\mathbf{A}$ only in matrix–vector products. In this way, the approximate solution is constructed as a polynomial in $\mathbf{A}$ applied to the initial residual, without explicitly forming or inverting the matrix [97].

The **Conjugate Gradient** (CG) method is a Krylov subspace method that is developed to minimize the following quadratic equation

$$F(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathsf{T}}\mathbf{A}\mathbf{u} - \mathbf{b}^{\mathsf{T}}\mathbf{u}, \tag{3.50}$$

where $\mathbf{A}$ is an SPD matrix. Solving this minimization problem (which is finding $\mathbf{u}^*$ such that $\nabla F(\mathbf{u}) = 0$) is equivalent to solving $\mathbf{A}\mathbf{u} = \mathbf{b}$, and falls under the type of gradient methods (see Fig. 3.10).

The solution $\mathbf{u}^*$ is approached recursively via the relation

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k, \tag{3.51}$$

where $\mathbf{d}_k$ is the search direction for the next iteration. One way is to choose the residual vector as the search direction, which is known as the *steepest descent* method. However, this can lead to the same search directions being computed more than once and oscillations occurring around the local minima, leading to very slow

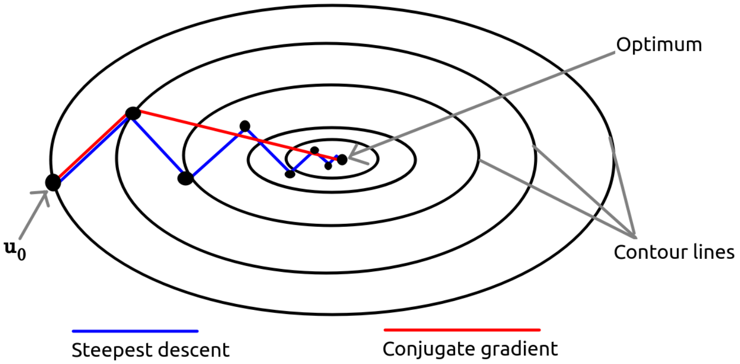

Figure 3.10: Comparison of the convergence between the steepest descent and CG method for an n x n matrix of size 2. In theory, the CG method should converge after n steps.

convergence. Therefore in the CG method, every search direction should be in a unique direction. This is accomplished by selecting a set of search directions that are **A**-orthogonal (or **A**-*conjugate*) to the previous directions. This means that they satisfy the condition $\mathbf{d}_k^\mathsf{T}\mathbf{A}\mathbf{d}_j = 0$, for $k \neq j$.

The CG method starts with the residual vector being chosen as the first search direction,

$$\mathbf{d}_0 = \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{u}_0. \tag{3.52}$$

The factor α_k in Eq. 3.51 ensures that the minimum point is found along the current search direction, which can be derived to the following expression

$$\alpha_k = \frac{\mathbf{d}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{d}_k^\mathsf{T}\mathbf{A}\mathbf{d}_k}. \tag{3.53}$$

After obtaining the new value for **u** (Eq. 3.51), the new residual is calculated,

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{d}_k. \tag{3.54}$$

To make sure that the next search direction is **A**-conjugate to the previous, the following coefficient is used which is simplified to,

$$\beta_k = \frac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}, \tag{3.55}$$

so that finally the next search direction can be calculated, which completes the algorithm,

$$\mathbf{d}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k. \tag{3.56}$$

If $\mathbf{A}$ is SPD and the denominators of Eq. 3.53 and Eq. 3.55 are equal to zero, it means that the CG method breaks down when the problem is already solved ($\mathbf{r}_k$= 0), which makes the method robust. The rate of convergence of the CG methods depends on the spectral properties (i.e. the eigenvalue distribution) of matrix $\mathbf{A}$ and is related to the condition number of matrix $\mathbf{A}$,

$$\kappa_2(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}, \tag{3.57}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of $\mathbf{A}$, respectively. The more clustered the eigenvalues are, the smaller the value of $\kappa_2(\mathbf{A})$, hence faster convergence. For CFD applications, the condition number is about the square of the maximum number of grid points, hence the standard CG method is slow [84].

PRECONDITIONERS

Preconditioners are used to accelerate the convergence by transforming the problem into a similar one, but with a more clustered eigenvalue distribution. This is done by multiplying the original system of equations by the inverse of the preconditioned matrix $\mathbf{P}$ as follows

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{u} = \mathbf{P}^{-1}\mathbf{b}. \tag{3.58}$$

In order for the preconditioned matrix $\mathbf{P}$ to be effective it must fulfill the following requirements. $\mathbf{P}$ should approximate $\mathbf{A}$, the computation of $\mathbf{P}^{-1}$ should be cheap, and the condition number of the transformed system should be smaller than that of the original system of equations,

$$\kappa_2(\mathbf{P}^{-1}\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{P}^{-1}\mathbf{A})}{\lambda_{\min}(\mathbf{P}^{-1}\mathbf{A})} << \kappa_2(\mathbf{A}). \tag{3.59}$$

For the CG method, $\mathbf{P}$ must also be an SPD matrix. Therefore, a good and common choice is the Cholesky factorisation (Eq. 3.35), which yields $\mathbf{P} = \mathbf{C}\mathbf{C}^{\mathsf{T}}$. But due to fill-in, it takes large amounts of work and memory to construct $\mathbf{C}$. Hence, the non-zero fill-in elements are disregarded, leading to the **incomplete Cholesky factorisation** of $\mathbf{P}$.

Algorithm 1 Preconditioned Conjugate Gradient (PCG) method

```
$\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{u}_0$ and $\mathbf{d}_0 = \mathbf{P}^{-1}\mathbf{r}_0$        ▷ Choose starting direction
for $k = 0, 1, ...,$ until convergence do
    $\mathbf{z}_k = \mathbf{P}^{-1}\mathbf{r}_k$
    $\alpha_k = \frac{\mathbf{d}_k^{\mathsf{T}}\mathbf{z}_k}{\mathbf{d}_k^{\mathsf{T}}\mathbf{A}\mathbf{d}_k}$
    $\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k\mathbf{d}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{d}_k$
    $\beta_k = \frac{\mathbf{r}_{k+1}^{\mathsf{T}}\mathbf{z}_{k+1}}{\mathbf{r}_k^{\mathsf{T}}\mathbf{z}_k}$
    $\mathbf{d}_{k+1} = \mathbf{z}_{k+1} + \beta_k\mathbf{d}_k$
end for
```

The algorithm of the CG method (Eqs. 3.51 - 3.56) can be transformed into that of the preconditioned CG method by replacing $\mathbf{r}_k$ with $\mathbf{z}_k = \mathbf{P}^{-1}\mathbf{r}_k$. The modification of the CG method is summarized in Algorithm 1.

Due to its superlinear convergence and for being robust, the preconditioned CG (PCG) method is one of the best methods for solving the heat equation (Eq. 2.77) , the P1 approximation (Eq. 2.69) and the pressure corrector equation (Eq. 3.26), as the coefficient matrices $\mathbf{A}$ that result from discretising these diffusion equations on the mesh are SPD.

For the general transport equations which contain the advection term, the coefficient matrix $\mathbf{A}$ is not symmetric. Therefore the PCG method is not applicable. A significant amount of Krylov subspace methods are developed for general non-SPD matrices $\mathbf{A}$, but none have all the following three nice properties which the CG method has for SPD matrices:

1. It is a Krylov subspace method: $\mathbf{u}_k \in \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0)$
2. Optimality: the minimizer of the quadratic function (Eq. 3.50) exists, hence convergence is guaranteed.
3. Short recurrences, hence requiring low computing work and storage.

GENERAL MATRICES

The **Bi-Conjugate Gradient** (BiCG) method is a Krylov subspace method for general matrices that transforms the unsymmetrical system into a symmetrical one so that it can by solved using a modified CG method. This is done by rewriting $\mathbf{A}\mathbf{u} = \mathbf{b}$ as

$$\begin{bmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}, \tag{3.60}$$

where $\hat{\mathbf{u}}$ is a dummy variable which is not solved for, but only used to convert the system. The update relations in the CG method for the residuals and search directions are augmented by their shadow equivalents which are based on $\mathbf{A}^\mathsf{T}$,

$$\hat{\mathbf{r}}_{k+1} = \hat{\mathbf{r}}_k - \alpha_k \mathbf{A}^\mathsf{T} \hat{\mathbf{d}}_k \tag{3.61}$$

and

$$\hat{\mathbf{d}}_{k+1} = \hat{\mathbf{r}}_{k+1} + \beta_k \hat{\mathbf{d}}_k. \tag{3.62}$$

The orthogonality of the residual and search direction with their shadow counterparts is ensured via the relation

$$\hat{\mathbf{r}}_i^\mathsf{T} \mathbf{r}_j = \hat{\mathbf{d}}_i^\mathsf{T} \mathbf{A} \mathbf{d}_j = 0, \quad i \neq j, \tag{3.63}$$

hence the name bi-orthogonal or bi-conjugate. The BiCG method is very close to the CG method and generates the same solution as the CG method if A is SPD. It does however require two matrix-vector multiplications (with $\mathbf{A}$ and with $\mathbf{A}^\mathsf{T}$) instead of one, such that every iteration is nearly twice as expensive as compared to the CG method.

Although the BiCG method shares two of the nice properties of the CG method, there is no optimality for general matrices $\mathbf{A}$, hence convergence is not guaranteed. Moreover, the irregular convergence can lead to serious break-down due to large rounding errors. The **Biconjugate Gradient STABilized** (Bi-CGSTAB) method [98] is a more robust variant where the recurrence relations are modified to a form

$$\tilde{\mathbf{r}}_k = Q_k(\mathbf{A}) P_k(\mathbf{A}) \mathbf{r}_0 \tag{3.64}$$

such that the multiplication with $\mathbf{A}^\mathsf{T}$ is avoided. For this recurrence, a k-th degree polynomial is taken of the form

$$Q_k(\mathbf{A}) = (\mathbf{I} - \omega_1 \mathbf{A})(\mathbf{I} - \omega_2 \mathbf{A}) \ldots (\mathbf{I} - \omega_k \mathbf{A}) \tag{3.65}$$

with suitable constants ω_k in the k-th iteration that minimizes residual r_k with respect to ω_k , giving the method a *semi-optimality* property and smoother convergence behaviour as compared to the BiCG method. The Bi-CGSTAB method is currently the most recent and advanced Krylov subspace method that is implemented in the standard library of OpenFOAM, and can be preconditioned using e.g. the **incomplete LU factorisation** which is appropriate for general matrices. It is much faster than its predecessor, but sudden breakdown due to rounding errors still occur for the more difficult problems in this thesis.

OTHER KRYLOV SUBSPACE METHODS

Although not used in this thesis, it is worth to mention the **Generalized Minimal RESidual** (GMRES) method [83], which falls under another class of Krylov subspace methods for solving general systems of linear equations. It is a stable method with respect to rounding errors, and has an optimality property which guarantees convergence. While the CG method aims at minimizing the A-norm of the error $\mathbf{u}-\mathbf{u}_k$ (only possible for SPD matrices), the GMRES method finds orthogonal vectors (search directions) that minimize the Euclidean norm of the residual $\mathbf{r}_k$, leading to a comparable superlinear convergence behavior as the CG method. The main drawback however is that the whole sequence of search directions from the previous iterates have to be stored and multiplied with, such that the computational and memory requirements quickly become prohibitive if the method does not converge after a few iterations. Restarting the method after a certain number of iterations overcomes this limitation of long recurrences, but comes at the cost of losing the nice convergence properties due to throwing away all search information [99]. The generalized conjugate residual (GCR) method is similar to the GMRES method but with somewhat more floating point operations per iteration. However, the GCR method gives the ability to truncate to the last few search directions that are saved for the next iteration, which in return performs better than restarting the GMRES method.

Hybrid methods provide a balance between the short recurrences of the BiCG-type methods on one side and the optimality property of the GMRES-type methods on the other. First on the list is the **GMRES Recursive** (GMRESR) method [99] that consists of an inner and outer loop. The solution is first approximated in an inner-loop with the restarted GMRES method, after which the found search directions are condensed to the outer-loop, where they are used to approximate the solution with the truncated GCR method. The GMRESR method generally speeds up the convergence rate of the restarted GMRES method. Recent improvements of the GMRESR-type methods are the **GCRO**-type methods, where the inner iteration loop takes place in a Krylov subspace orthogonal to the subspace of the outer loop (so-called *subspace recycling*), yielding further acceleration of the convergence rate[100] [101].

The final hybrid method worth mentioning, which leans more towards the BiCG-type methods, is the **Induced Dimension Reduction** (IDR) method. This method was already proposed before the Bi-CGSTAB method but has been revived to a more generalized variant IDR(s) [102]. The IDR(s) method is closely related to the Bi-CGSTAB method in the sense that the matrix-vector multiplication with $\mathbf{A}^\mathsf{T}$ is avoided via a minimizing polynomial. The main difference is that the generated residuals are forced to be in subspaces of decreasing order until an s-dimensional space is reached. For the right value of s (typically $s \leq 10.$), this robust and efficient method is at least as fast as the Bi-CGSTAB method when $s = 1$ (original IDR method) and significantly faster for increased values of s, especially for more difficult problems. When s is large enough, the rate of convergence is nearly as good as full GMRES [103]. Hybrid methods such as the **IDR(s)** method and the **GCRO**-type methods are good candidates for the next generation of linear solvers for general matrices to implement in OpenFOAM.

3.4. IMPACT LINEAR SOLVERS ON CONVERGENCE OF THE SIMPLE ALGORITHM

The speed of convergence towards a solution depends not only on the type of linear solvers used, but also on the type of problem that is being solved. This section evaluates the performance of the advanced linear solvers in OpenFOAM by conducting several numerical experiments on different cases.

3.4.1. RELATIVE TOLERANCE

The solver algorithms of OpenFOAM are constructed in 'inner' and 'outer' loops. In the inner loops, the linear systems of equations are solved sequentially, and this is where the linear solvers do the work, which are referred to as 'inner' iterations. The algorithm proceeds to the next linear system of equations in the inner loop once the current *relative tolerance* target is reached. In the outer loop, the flow develops to the next (pseudo) time step, which may or may not require a time integration, depending on whether the SIMPLE algorithm is used for steady-state problems, or the PISO algorithm for transient problems, or combinations thereof (PIMPLE). As the algorithms are iterative themselves, the linear solvers do not need to iterate to the smallest possible error in each loop before the algorithm progresses.

The 2D steady-state heat equation is a problem which does not require time integration or a flow to be developed. As such, this problem is solved within 1 outer iteration, and even up to 1 inner iteration with the most efficient solver, as is shown in Figs. 3.11 and 3.12.

In a fluid-related problem, however, the flow is physically developing on each outer iteration. Aiming for a low relative tolerance per inner loop can be more stable,

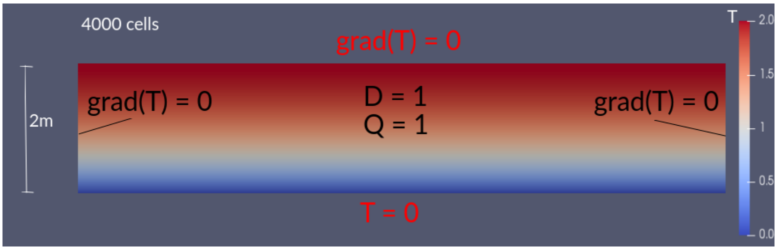

Figure 3.11: Example 2D steady-state heat equation problem in a rectangular surface with 4000 equally divided cells, along with its boundary conditions.

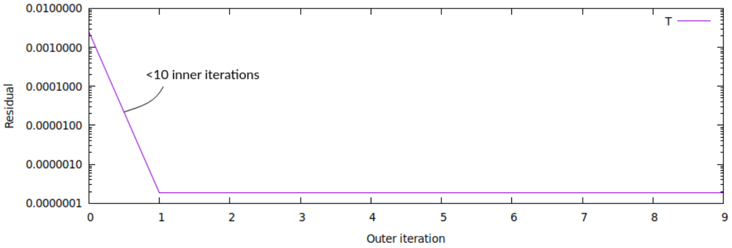

Figure 3.12: Residual plot of the 2D steady-state heat equation problem.

but it also makes it much more time-consuming. The famous Pitz and Daily wind tunnel tutorial case [86] demonstrates that using a relative tolerance of 10^{-1} instead of 10^{-3} for all equations will lead to about twice as fast convergence, while a relative tolerance of 10^{-3} leads to smoother convergence with less oscillations, as is shown in Fig. 3.13.

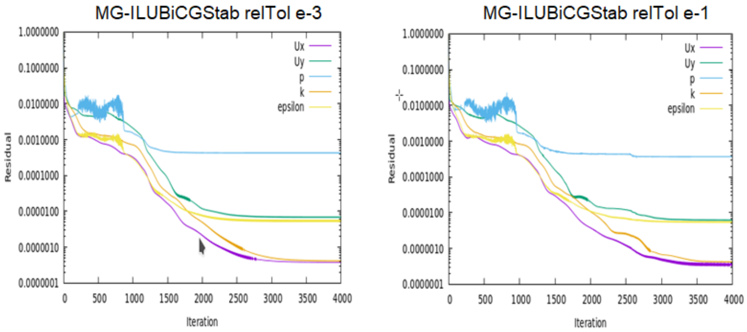

Figure 3.13: Comparison between two simulations of the Pitz and Daily wind tunnel tutorial case using 10^{-3} and 10^{-1} as relative tolerance, respectively. The pressure equation is solved with the GAMG method , while the other equations are solved with the Bi-CGSTAB method and ILU preconditioner.

3.4.2. NUMERICAL EXPERIMENT: NON-REACTING FLOW

This section evaluates the performance of the linear solvers on a three-dimensional full scale, non-reacting and non-rotating kiln. The used solver is `rhoSimpleFOAM`, which is the standard solver for turbulent non-reacting steady-state compressible flows.

The geometry is shown in Figs. 3.14 and 3.15, where the air flow enters the domain through the primary and secondary inlets with 0.15 kg/s and 1.6 kg/s, respectively, while the fuel inlet flow is excluded. The mesh contains 2.3 million cells and all simulations are run for 5000 outer iterations on 20 cores in a single node.

Fig. 3.16 shows the basic stream pattern in the longitudinal cross-section, where the annular shaped recirculation zone surrounding the burner is spotted. This case study is studied extensively in Chapter 6, but this section only focuses on the numerical performance of the linear solvers.

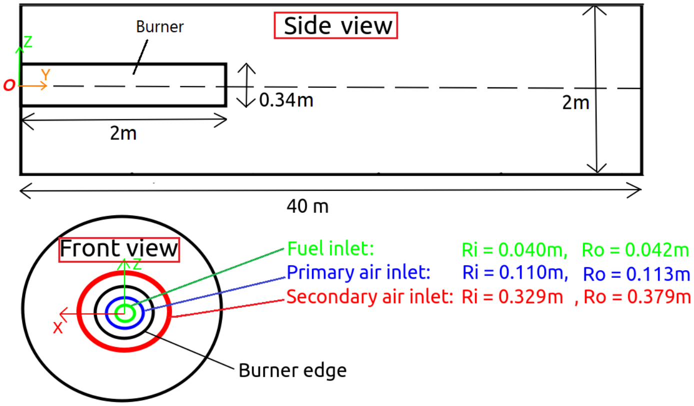

Figure 3.14: global dimensions of the kiln, with the inner and outer radii of the inlets

SOLVER COMBINATIONS

The most advanced standard linear solvers of OpenFOAM have been tested on this applications, in different combinations which are shown in Tab. 3.2. For each combination, all transport equations are solved with the same solver, except for the pressure equation, which is solved differently because the coefficient matrix is symmetric. Instead of the IC and ILU factorisations, OpenFOAM implements the simplified diagonal-based variants as preconditioner, respectively the DIC and DILU, where off-diagonal terms are dropped. The reciprocal of the preconditioned diagonal is calculated and stored.

The settings for OpenFOAM's MG solver (GAMG) is shown here:

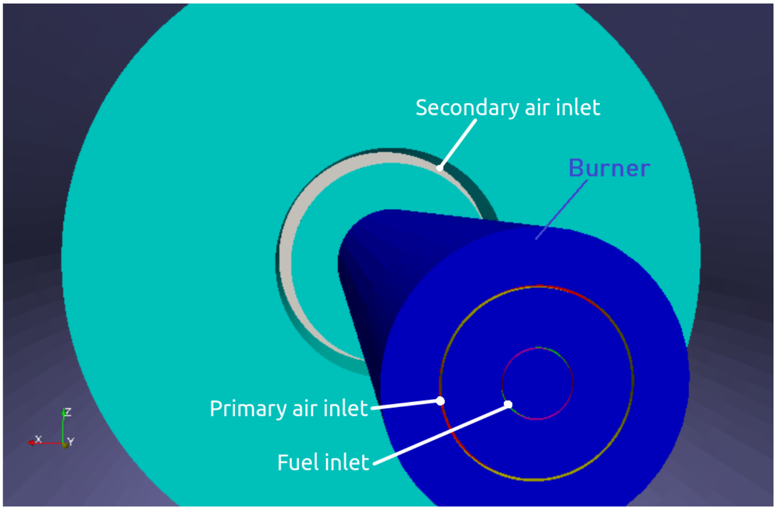

Figure 3.15: Inside view of the burner side of the kiln with the three inlets.

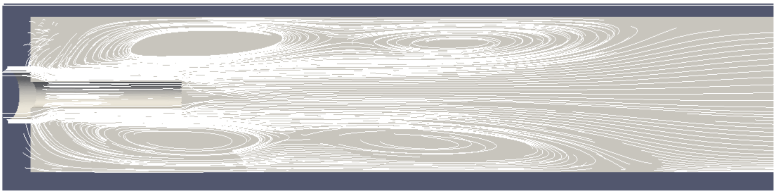

Figure 3.16: Stream pattern inside the kiln for the studied case in this section.

Table 3.2: Tested solver combinations. The first solver of each method is used for the pressure equation, while the second solver is used for all other independent variables .

Method	p Eq.	Other variables
MG-DILUBiCGSTAB	Solver: MG	Solver: Bi-CGSTAB, Prec: DILU
MG-MG	Solver: MG	Solver: MG
DICCG-MG	Solver: CG, Prec: DIC	Solver: MG
DICCG-DILUBiCGSTAB	Solver: CG, Prec: DIC	Solver: Bi-CGSTAB, Prec: DILU
MGCG-DILUBiCGSTAB	Solver: CG, Prec: MG	Solver: Bi-CGSTAB, Prec: DILU
MGCG-MGBiCGSTAB	Solver: CG, Prec: MG	Solver: Bi-CGSTAB, Prec: MG

```
solver                    GAMG;
tolerance                 1e-9;
relTol                    e-3;  //e-1;
smoother                  GaussSeidel;
cacheAgglomeration        true;
nCellsInCoarsestLevel     1500;
agglomerator              faceAreaPair;  //algebraicPair;
```

```
mergeLevels              1;
nPreSweeps               1; //1 for p, 0 for all other
nPostSweeps              2;
nFinestSweeps            2;
maxIter                  800;
directSolveCoarsest      false;
```

where the number of cells at the coarsest level is set to 1000, which is roughly the square root of the total amount of cells. For these type of problems with complex shape and flows, the geometric variant of the GAMG method (agglomerator: faceAreaPair) is preferred, which is about twice as fast as the algebraic variant.

Results

Before discussing the linear solver performance, it is observed that by using the advanced solvers of OpenFOAM usually leads to 1 or 2 inner iterations per outer iteration for all variables, except with the pressure equation, which usually requires 1 or more orders of magnitude of inner iterations, before moving to the next outer iteration. Therefore the pressure-corrector equation is the major bottleneck that delays the solution to converge quickly. This occurs for most non-reacting flow cases. Fig. 3.17 shows that the CG method with the DIC preconditioner is the least efficient method for the pressure equation, requiring 2 orders of magnitude of inner iterations on average to reach the relative target of 10^{-3}. The MG method performs significantly better, and the most efficient method is the combination of the former two: the CG method with the MG as preconditioner.

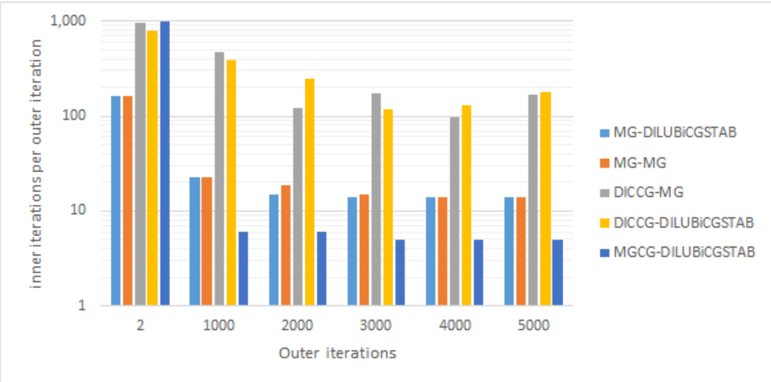

Figure 3.17: amount of inner iterations of the pressure equation per outer iteration, using a **relative tolerance of** 10^{-3}**.**

Fig. 3.18 shows the overall performance of the different linear solver combinations. Three critera are used to determine the speed of convergence:

- The time (in hours) to reach 5000 outer iterations,
- The amount of outer iterations to reach convergence,
- The time (in hours) to reach convergence.

The time needed to reach 5000 iterations with the different combinations is in line with the amount of required inner iterations for the pressure equation as is discussed above, since the pressure equation is the bottleneck.

When looking at the amount of outer iterations required to converge, the CG method with MG preconditioner also requires the least amount of iterations. This further accelerates the convergence time, which is a factor 2 to 3 times faster than the other solver combinations. However, if the solver is forced to a strictly compressible flow environment by setting the option *transonic* to *on*, the pressure equation becomes non-symmetric and the MG method will not work.

With regard to the other variables, the fastest solver is the Bi-CGSTAB method with the DILU preconditioner. The MG method, as solver or as preconditioner, is a bit slower because the other variables only require 1 or 2 inner iterations to reach the relative target of 10^{-3}, and one MG operation is computationally more expensive than one Bi-CGSTAB operation, and much more expensive than one DILU operation.

Table 3.3: Simulation time of the different methods w.r.t time and outer iterations, using a **relative tolerance of** 10^{-3}

Method	Time to 5000 iter	Convergence	Time to Convergence
MG-DILUBiCGSTAB	3.4 h	3000 iter	2.2 h
MG-MG	4.3 h	3500 iter	3.1 h
DICCG-MG	5.3 h	3000 iter	3.6 h
DICCG-DILUBiCGSTAB	6.4 h	5000 iter	6.4 h
MGCG-DILUBiCGSTAB	2.3 h	2500 iter	1.2 h
MGCG-MGBiCGSTAB	2.7 h	2500 iter	1.4 h

The simulation can be accelerated further by relaxing the relative target from 10^{-3} to 10^{-1} , as is shown in Fig. 3.19. This however may lead to less stability. In fact when using the DICCG-DILUBiCGSTAB method, the solution will even quickly diverge, as can be seen in Fig. 3.20, while the MGCG-DILUBiCGSTAB method remains stable and the convergence is accelerated with 25 % as compared with the relative tolerance of 10^{-3} . Comparing these two extremes, with on one side the slowest of the selected methods that is restricted to a relative target of 10^{-3} , and on the other side the fastest method that is stable at a relative target of 10^{-1}, a speed-

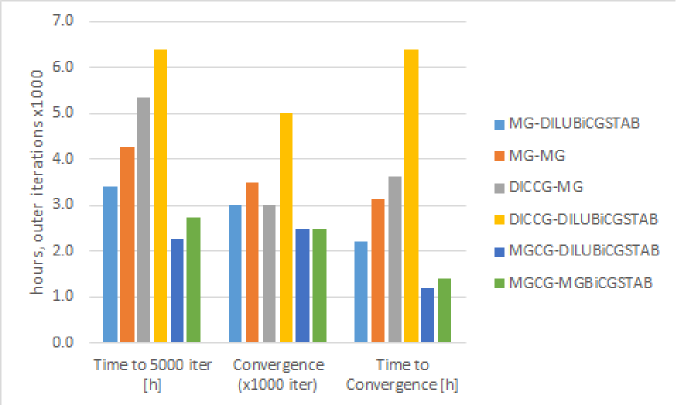

Figure 3.18: Simulations durations of the different methods w.r.t time and outer iterations, using a **relative tolerance of** 10^{-3}

up of nearly a factor 7 is reached. This speed up does not even consider the basic iterative solvers of OpenFOAM (`smoothSolver`).

Table 3.4: Simulation time of the different methods w.r.t time and outer iterations, using a **relative tolerance of** 10^{-1}

Method	Time to 5000 iter	Convergence	Time to Convergence
MG-DILUBiCGSTAB	2.0 h	3000 iter	1.2 h
MG-MG	2.5 h	4000 iter	2.0 h
DICCG-MG	2.7 h	3000 iter	1.8 h
DICCG-DILUBiCGSTAB	-	**Diverged**	-
MGCG-DILUBiCGSTAB	1.5 h	3000 iter	0.9 h

FURTHER ADVANCEMENTS

As discussed in the end of Sec. 3.3, faster linear solvers do exist than the ones provided by OpenFOAM. The state-of-the-art linear solvers and preconditioners can be found in the open source library **Portable, Extensible Toolkit for Scientific computation** (PETSc), and a plug-in into OpenFOAM is developed in [104]. Their work also discusses the very poor scalability of the MG preconditioner of OpenFOAM for massively parallel clusters (10^3 cores), whereas the DIC preconditioner is shown to have outstanding superlinear scalability. It is reported that the DIC preconditioner overtakes the MG preconditioner in convergence time when using more than 2000 cores for a 3D laminar lid driven cavity flow problem consisting of 64 millions cells. The PETSc counterpart of OpenFOAM's MG preconditioner solves the scalability issue.

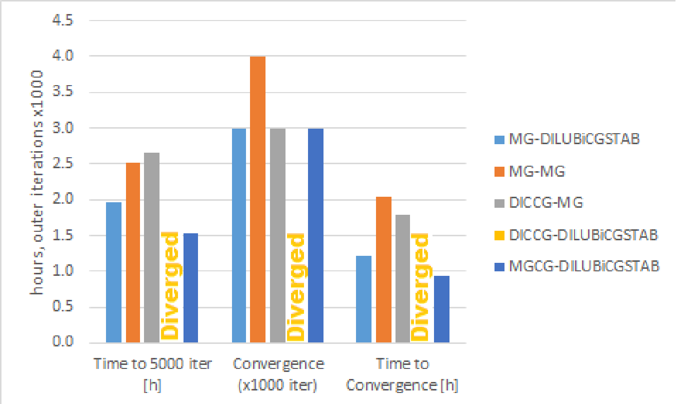

Figure 3.19: Simulations durations of the different methods w.r.t time and outer iterations, using a **relative tolerance of** 10^{-1}

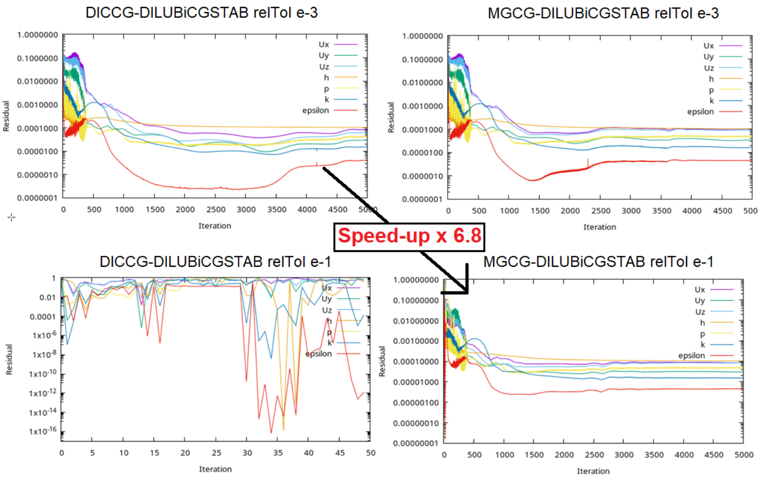

Figure 3.20: Residuals of the studied case for the least efficient and most efficient solver combination.

3.4.3. NUMERICAL EXPERIMENT: REACTIVE FLOW

While for non-reacting flows the performance of the linear solver really matters, it is a different story for reactive flows. When enabling the simple combustion model EDM for the studied kiln in the previous subsection. The speed-up of the CG method with MG preconditioner for the pressure equation is only 4% as compared with the DIC preconditioner. Deploying a detailed reaction mechanism completely diminishes this speed-up. For reacting cases, solving the thermochemistry turns out to be the bottleneck.

There are different approaches to accelerate the thermochemistry with OpenFOAM, and in this section it will be applied on the Sandia flame D tutorial (based on the experiment), where the burner dimensions can be found here [105]. This tutorial case uses the GRI-3.0 mechanism, which is an expensive mechanism. Two of the different ways to accelerate thermochemistry which were presented and discussed in Sec. 2.3 are applied here: using the TDAC method and by simplifying the reaction mechanism with e.g. the DRM19 mechanism. Since the studied case is a transient problem, also the LTS-method (Sec. 3.2.4) will be tested. The convergence times are shown in Fig. 3.21.

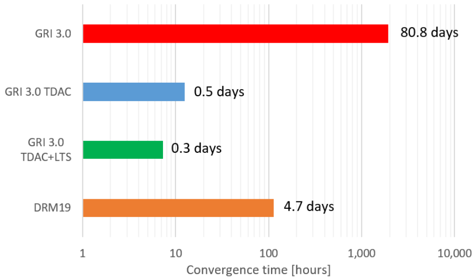

Figure 3.21: Convergence time of the different chemistry accelerations methods on a single core w.r.t. the GRI-3.0 mechanism.

Although a simplified mechanism significantly reduces the simulation time, the TDAC method is far more superior, with a speed-up factor of about 160. Local time stepping in this case is not significantly faster due to the small residence time of the domain where the transient simulation reaches a steady-state solution relatively quickly.

While the linear solvers do not affect the quality of the solution, the different ways of accelerating thermochemistry do have effect. Therefore also the quality of the solution will be compared with measurements [106] by looking at the temperature and mass fractions of certain species along the axis (data: DCL.Yfav). These are presented in Fig. 3.22. It is clear that the different methods effect the results. As compared with the reference simulation, the TDAC method and local time stepping show relatively small differences. The largest differences are with DRM19 mechanism, where the NO formation is solved with the post-processor. Therefore TDAC has preference over a simplified mechanism in a transient simulation, together with LTS if a steady-state solution exists.

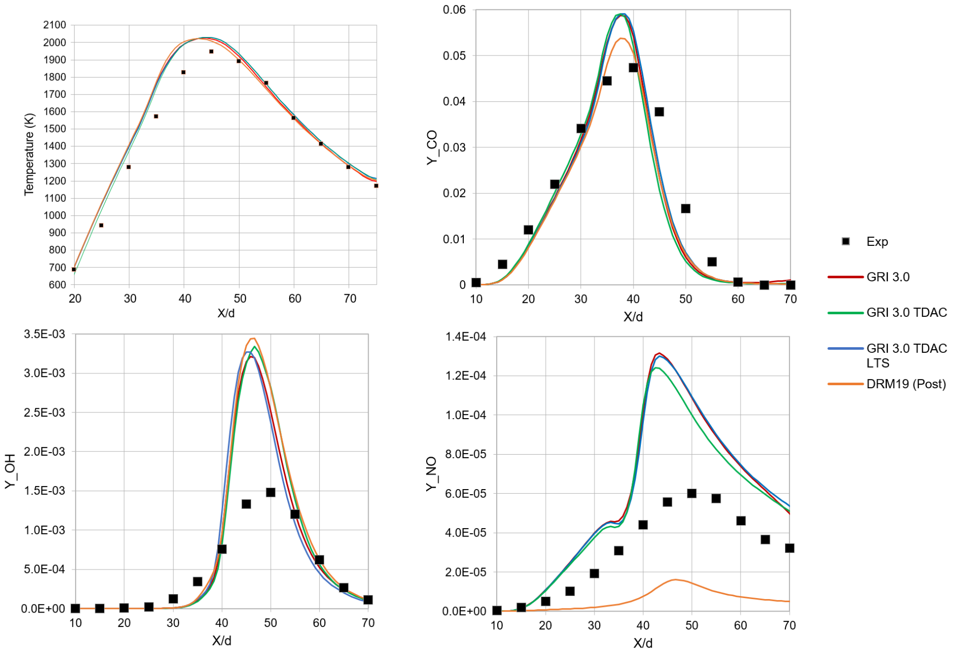

Figure 3.22: Plots of the gas temperature and mass fractions of CO, OH and NO along the longitudinal axis of the Sandia flame D.

3.5. PROPOSED METHODOLOGY: COUPLE ENERGY TRANSFER BETWEEN REACTIVE GAS AND SOLID

To conclude the theoretical part of this dissertation, a mathematical solver is proposed that implements all the necessary physical models to simulate the combustion and heat transfer process of the rotary kiln (or industrial furnace in general), as is discussed in Chapter 2. In the beginning of this research, the needed solver did not exist in OpenFOAM. The first solver to successfully couple the combustion solver and CHT-solver of OpenFOAM is developed by Tonkomo LLC [107]. The source code of this solver, `multiRegionReactingFoam`, can be found in [108], which we supported, tested and verified.

In recent years, OpenFOAM has added the combustion utility in the standard CHT-solver `chtMultiRegionFoam`. The main differences of the two solvers are how the compressibility ψ (Sec. 2.2.5) and the hydrostatic pressure ($\rho \mathbf{gh}$) are involved in the pressure corrector equation. In `chtMultiRegionFoam`, ψ is accounted for explicitly (`fvc::`) as a correction to $d\rho/dt$ and the hydrostatic pressure is treated implicitly (`fvm::`):

```
fvScalarMatrix p_rghEqn
(
fvc::ddt(rho) + psi*correction(fvm::ddt(p_rgh))
+ fvc::div(phiHbyA)
- fvm::laplacian(rhorAUf, p_rgh)
);
```

This makes `chtMultiRegionFoam` applicable for general fluids, including multiphase flows and buoyant flows.

In `multiRegionReactingFoam`, ψ is treated implicitly in the pressure corrector, while the hydrostatic pressure is excluded:

```
fvScalarMatrix pEqn
(
fvm::ddt(psi, p)
+ fvm::div(phid, p)
- fvm::laplacian(rhorAUf, p)
);
```

In here, the first two terms discretise the compressibility-weighted material derivative of pressure, i.e. the pressure-work term introduced in the energy balance equation (Eq. 2.14):

$$\psi \frac{Dp}{Dt}. \tag{3.66}$$

`multiRegionReactingFoam`, is therefore more robust for high speed gas flows and has our preference, as the fuel inlet channel flows of the kiln's burner exceed Mach 0.6.

The general flow diagram of the solver's algorithm is shown in Fig. 3.23. The solver strategy for each iteration is to first solve all fluid regions (yellow) in assigned order, and then to proceed solving for all solid regions (green) in assigned order. If transient simulations are conducted, the pressure corrector step is repeated multiple times in an internal loop, and the time step size is determined by the most critical fluid region. For best practice, also the recommended linear solvers are given for each balance equation.

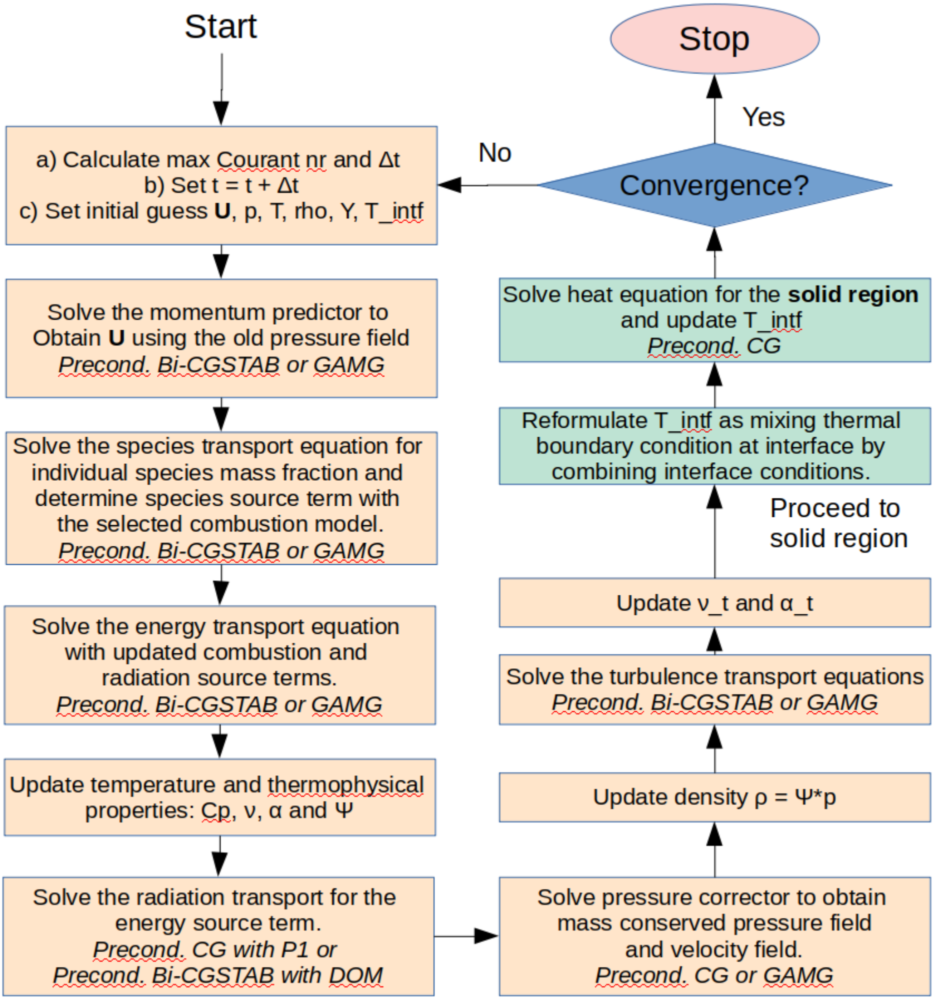

Figure 3.23: Flow diagram of the solver multiRegionReactingFoam with recommended linear solvers

II

APPLICATIONS

4

IMPLEMENTATION AND VALIDATION OF THE COMBUSTION/CHT SOLVER

In this chapter, the capabilities of the new solver are investigated by testing it on the first and most simple case, which is an open turbulent flame from the Sandia laboratory. This 2D axisymmetric case will be used to validate the solver's implementation of turbulent combustion by comparison with experimental results. Afterwards, CHT is also activated by adding a cylindrical solid region that represents the furnace wall. This way, the effect of the walls on the flame characteristics can be analysed and the heat flux profile on the wall can be determined. The results are then compared with the ones generated by ANSYS-Fluent. This chapter continues investigating the conjugate heat transfer module of the solver applied on a more complex application; the burner flow reactor adds recirculation to the problem which occurs in almost every combustion application in the industry due to its flame stabilizing effects.

This chapter is based on the article:

M. el Abbassi, D.J.P. Lahaye, and C. Vuik. Modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM. In *Numerical Mathematics and Advanced Applications ENUMATH 2019: European Conference, Egmond aan Zee*, The Netherlands, pages 1137–1145. Springer, 2020,

and additional work.

4.1. INTRODUCTION

Industrial furnaces such as kilns are pyroprocessing devices in which a heat source is generated via fuel combustion. In order to make a numerical prediction of the temperature distribution along a solid (e.g. the material bed, furnace walls, or heat exchanger), one must model the coupled effects of the occurring physical phenomena. The heat released by the fluctuating turbulent flame may be transferred to the solid through all heat transfer modes: thermal radiation, conduction, and convection. Thermal radiation is transmitted to the solid directly from the flame, or indirectly from the hot exhaust and other solids. Conduction occurs within solids and through contact with other solid particles, while convective heat may be exchanged via any contact between gas and solids. In return, the fluctuating heat transfer affects the turbulent flow and flame characteristics. Controlling the flame enables achieving the desired heat distribution with minimum emissions. Coupling combustion and heat transfer is essential to find optimal solutions to these conflicting interests, particularly in view of increasing environmental concerns (which view reducing the furnace emissions and fuel consumption as urgent), along with the growing demand for an increase in furnace production rate.

Incorporating the heat transfer between fluids and solids into one mathematical problem may be referred as conjugate heat transfer (CHT). CHT is implemented in many popular CFD codes. There are several publications available on furnace models where combustion and CHT are coupled. For example, in the work [27], the prediction of the furnace wall heat distribution was made with CD-Adapco's STAR-CCM+. ANSYS-CFX was used in [28] to model the heat distribution, while ANSYS-Fluent was the CFD-tool for the works of other researchers [29, 30, 31].

At the time of this research there were no publications on coupling turbulent combustion and CHT with the open source CFD-toolbox OpenFOAM. OpenFOAM sets a structured object-oriented framework and includes numerous applications to solve different kinds of CFD-related problems. The source code is fully accessible and allows building new or modified applications while making use of existing libraries, models and utilities to link them. OpenFOAM also allows high performance computing using e.g. MPI and GPU's that do not include any license costs, and hence may lead to significant savings for large and complex problems. There are numerous studies in which combustion solvers of OpenFOAM were benchmarked against different experiments and other solvers (e.g. [32, 33, 34, 35]) and some include thermal radiation for the heat transfer (e.g. [36]). The capabilities of OpenFOAM's CHT solver have also been studied extensively and some recent investigations into this matter, (with and without radiative heat transfer) can be found in [37, 38, 39, 40].

Although all the necessary libraries were available in OpenFOAM to model the required physical phenomena, there was no standard implementation available in the existing releases that coupled combustion and CHT. However, by the time of attempting to couple the solvers, an implementation was proposed and developed for OpenFOAM by Tonkomo LLC [108], that combines the turbulent-non-premixed-combustion solver `reactingFoam` with the CHT-solver `chtMultiRegionFoam`. This provided new opportunities for modelling furnaces or any other combustion and heat transfer related problem. A similar code was developed very recently in [8]. In our work [109][110], the capabilities of the new solver are investigated by testing it on different 2D axisymmetric cases, in increasing order of complexity, by means of RANS simulation.

The first case is an open turbulent flame from the Sandia laboratory which is used to validate the solver's implementation of turbulent combustion. Afterwards, CHT is also activated by adding a cylindrical solid region that represents the furnace wall. This way, the effect of the walls on the flame characteristics can be analysed and the heat distribution on the wall can be determined. The results are then compared with the ones generated by ANSYS Fluent.

4.2. NUMERICAL SET-UP

4.2.1. TEST CASES

The solver is tested on three methane-air combustion cases. In the first case, the implementation of combustion in the new solver is validated with experimental data from a turbulent piloted diffusion flame from the Sandia National Laboratories (Sandia Flame D). The burner dimensions can be found here [105] and the experimental data are found here [106], where the data file DCL.Yfav is used.

For the second case, CHT is activated and the Sandia Flame D is confined by a cylindrical wall made of refractory material, with inner and outer diameters of respectively 300 and 360 mm. The axial length of the calculation domain (excluding fuel and pilot channels) is 600 mm. The boundary conditions of the two cases can be found in Table 4.1, and the wall dimensions and properties are shown in Table 4.2.

For the third and final case, a geometry is designed to introduce circulation in the flow. For this reactor, the dimensions are adopted from the Burner Flow Reactor (BFR) in [111], with some major differences. Rather than using swirling air for flame stabilisation, we chose to use hot co-flow that is injected axially. We left out the narrowed exhaust pipe to increase the adverse pressure gradient along the central axis, causing the jet to decelerate more, and thus further improve flame stability. The notion behind these modifications is to have a simpler flow, similar to that of a backward facing step, which we can better understand in 2D. The final geometry is shown schematically in Fig. 4.1, where the fuel channel diameter is 10 mm and the hot co-flow inner and outer diameters are respectively 11 mm and 200 mm. The fuel tube has a wall thickness of 0.5 mm. The boundary conditions are shown in Table 4.3 and the wall material is the same as for case 2. The operating power is 200 kW with a fuel-air equivalence ratio of 0.8.

Table 4.1: Boundary and initial conditions for Sandia Flame D. zG stands for the Neumann boundary condition zeroGradient. The axial velocities are expressed in m/s, and the temperatures in K. Species are denoted in mass fractions.

Variable	Fuel jet	Pilot jet	Co-flow	Gas-wall interface	Outer wall surface	Side wall surfaces
U_{axial} (m/s)	49.6	11.4	0.9	0	-	-
T (K)	294	1880	291	Coupled	291	zG
Y_{CH_4}	0.1561	0	0	zG	-	-
Y_{O_2}	0.1966	0.054	0.23	zG	-	-
Y_{N_2}	0.6473	0.742	0.77	zG	-	-
Y_{H_2O}	0	0.0942	0	zG	-	-
Y_{CO_2}	0	0.1098	0	zG	-	-

Table 4.2: Thermal properties of the refractory material.

Density ρ	Thermal conductivity λ_s	specific heat capacity c_p	emissivity ϵ_s
2800 $\mathrm{kgm^{-3}}$	2.1 $\mathrm{Wm^{-1}K^{-1}}$	860 $\mathrm{Jkg^{-1}K^{-1}}$	0.6

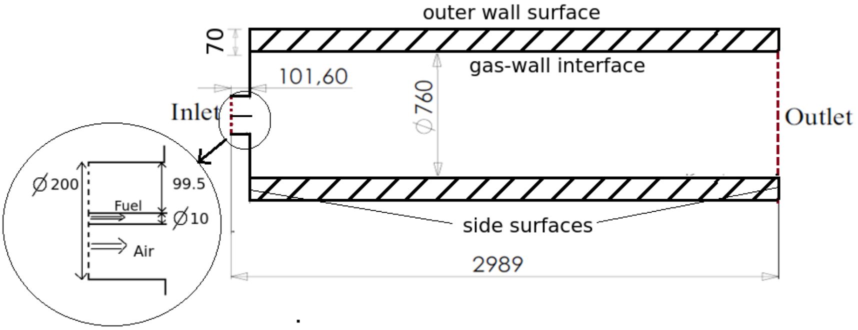

Figure 4.1: Schematic geometry of the modified BFR. Dimensions are denoted in mm.

Table 4.3: Boundary and initial conditions for the modified BFR. zG stands for the Neumann boundary condition zeroGradient. The axial velocities are expressed in m/s, and the temperatures in K. Species are denoted in mass fractions.

Variable	fuel jet	co-flow	gas-wall interface	outer wall surface	side wall surfaces
U_{axial} (m/s)	82.22	6.619	0	-	-
T (K)	300	800	Coupled	800	zG
Y_{CH_4}	1	0	zG	-	-
Y_{O_2}	0	0.234	zG	-	-
Y_{N_2}	0	0.766	zG	-	-

4.2.2. PHYSICAL MODELS

The chosen physical models for the solver are listed in Table 4.4 for each test case. For cases 1 and 2, the Reynolds stresses are closed with the Realizable k-ε model, and the PaSR model was used for the mean species source term. The 2-step Westbrook and Dryer (WD) reaction mechanism [46] is used for test cases 1 and 2, while the GRI 3.0 mechanism [47] is only used for validation of test case 1. To have a comparison with Fluent as close as possible, test case 3 applies the Standard k-ε model for turbulence, and the EDM for combustion, using the single-step mechanism of Westbrook and Dryer. The mean radiative heat source is modelled using the DOM and the species emissivities are determined with OpenFOAM's sub-model `greyMeanAbsorptionEmission` for all cases.

Table 4.4: Selected physical models for the test cases

Case	1 & 2	3
Simulation type	Transient	Steady-state
Turbulence	Realizable k-ε	Standard k-ε
Wall treatment	Log-law	Log-law
Combustion	PaSR	EDM
Mechanism	GRI-3.0 & 2-step WD	1-step WD
Radiation	DOM	DOM
Absorption model	greyMeanAbsorptionEmission	greyMeanAbsorptionEmission

4.2.3. NUMERICAL METHODS

The computational domains of cases 1 to 3 consist of respectively 55000, 69000 and 67000 quadrilateral cells. The computational domains are twice as long in axial direction as compared with the geometries to avoid the strong influence of the outlet boundary in radiation. Transient simulations are conducted for cases 1 and 2, and the time step is determined from the max Courant number which is set to 0.4. The first-order implicit Euler discretisation scheme is used for the unsteady terms. Steady-state simulations are carried out for Case 3.

Regarding spatial discretisation, the central differencing scheme is selected for the gradient and Laplacian terms. The convection terms for velocity, enthalpy, chemical species and radiation intensity are discretised using the linear upwind scheme while for the kinetic energy and energy dissipation this is done with the first-order upwind scheme, as to prevent OpenFOAM from crashing. The discrete ordinates for the DOM are divided in 5 azimuth angles and 1 polar angle per quadrant of the 2π plane angle.

The generated linear systems of equations are solved as follows. The mass fluxes are solved using the Preconditioned Conjugate Gradient (PCG) method using the diagonal incomplete Cholesky factorisation as a preconditioner, while for the pressure-corrector the multigrid method is preferred as preconditioner combined with the PCG. The remaining variables are solved with the Preconditioned Bi-CGSTAB method and using the diagonal incomplete LU factorisation as preconditioner, which is suitable for non-symmetric sparse matrices caused by the convection terms.

4.2.4. VERIFICATION WITH ANSYS-FLUENT®

In order to verify the new solver, the results of case 3 are compared with the results of ANSYS Fluent. The test case in Fluent is therefore set up as close as possible to that of `multiRegionReactingFoam`, using the models as shown in Table 4.4. The gas properties are also made consistent, such as applying ideal gas mixing law, Sutherland model and unity Lewis number. However, there are two major differences. The first difference is Fluent's second-order upwind scheme which is slightly different than OpenFOAM's linear upwind scheme. The second difference is that the gas mixture's absorption coefficients are computed with a more advanced grey gas assumption model, the Weighted Sum of Gray Gases (WSGG), where the absorption coefficients also depend on beam length and partial pressures of the gas species, which are not taken into account with the `greyMeanAbsorptionEmission`. Regarding wall treatment, Fluent's standard wall function comes closest to OpenFOAM's combination of `nutkWallFunction`, `kqRWallFunction` and `epsilonWallFunction` which are based on the log-law.

4.2.5. SIMULATION PROCEDURE

The simulations run until convergence is observed based on 2 criteria. Firstly, the residuals have to converge at their own lowest tolerance levels (minimum 10^{-9}) where they oscillate (Fig. 4.2), with only exception for U_Z as this is a 2D simulation. Secondly, convergence needs to be observed with the maximum gas temperature, maximum solid temperature, average outlet temperature and average outlet CO_2 fraction (Figs. 4.3 and 4.4). The computation starts with a solved non-reacting turbulent flow field as initial state.

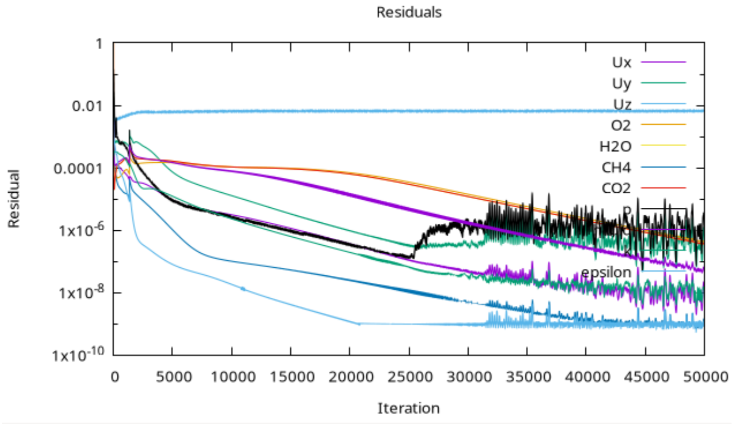

Figure 4.2: Example residual plot of a converged solution.

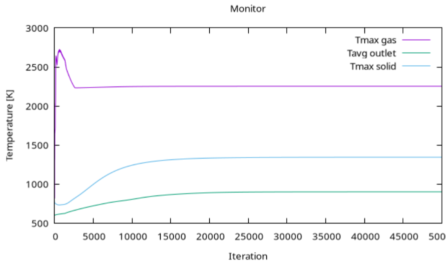

Figure 4.3: Example temperature monitor plot of a converged solution.

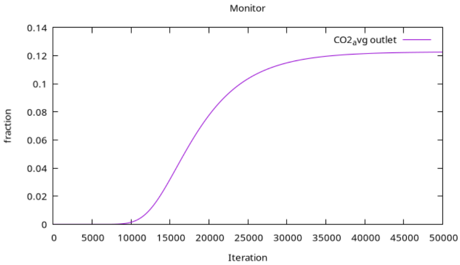

Figure 4.4: Example species monitor plot (outlet) of a converged solution.

4.3. RESULTS AND DISCUSSION

4.3.1. CASE 1

In Fig. 4.7 the temperature along the axis of symmetry is plotted. It shows that the `multiRegionReactingFoam`'s prediction is identical to that of `reactingFoam`, as would be expected when CHT is switched off. Both solvers over-predict the ignition delay, temperature rise and peak temperature with the 2-step reaction mechanism. This poor prediction is the consequence of not taking enough intermediate species into account and therefore resulting in a smaller reaction zone with sharp peaks. As radiation scales with T^4, the wall heat fluxes will be over-predicted as well with the 2-step reaction mechanism. When using the full GRI mechanism, these features are much better captured, but still not within the experimental uncertainty of 3% - 5% [112], especially before the peak. This is mainly due to the limitation of the k-ε model which doesn't resolve the turbulent fluctuations responsible for mixing.

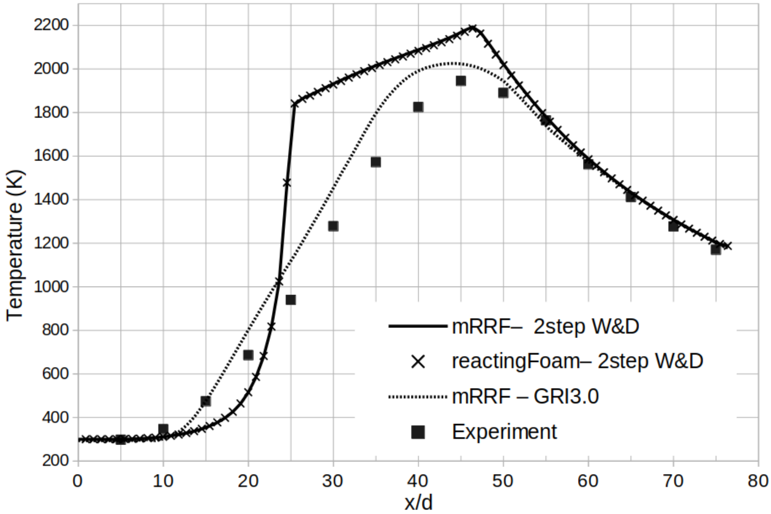

Figure 4.5: Temperature progression along the center line (Case 1), where 'd' is the main jet inner diameter of 7.2 mm.

Figure 4.6: Contour plot of the temperature (Case 2).

4.3.2. CASE 2

Now that a wall is introduced around the Sandia Flame D, it absorbs some of the energy, as can be seen in Fig. 4.6. Fig. 4.8 shows a decomposition of the heat transfer to the wall in which the wall is being heated only due to thermal radiation.

The wall is not heated via convection due to the fact that the hot gas heated by the flame leaves the domain before coming into contact with the wall. In fact, the convective heat transfer part plays a cooling role by transferring some of the wall's heat to the cold adjacent air, hence the negative contribution. The radiative energy absorption by the wall has an additional cooling effect on the flame, as can be seen in Fig. 4.7.

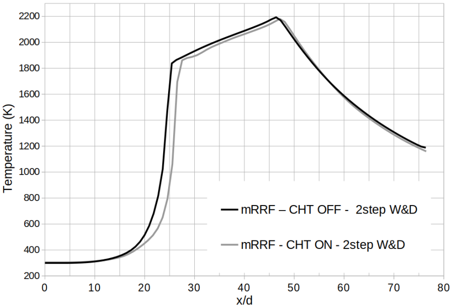

Figure 4.7: Temperature progression along the center line (Case 2), with and without Conjugate Heat Transfer. 'd' is the main jet inner diameter of 7.2 mm.

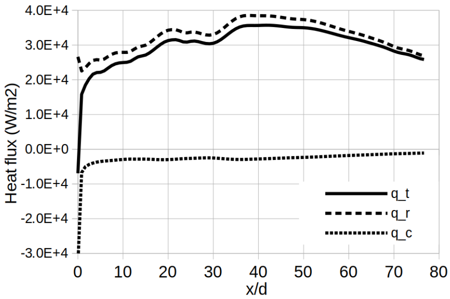

Figure 4.8: Heat flux along the inner wall surface (Case 2). q_t, q_r and q_c are respectively the total, radiative and convective heat fluxes. 'd' is the main jet inner diameter of 7.2 mm

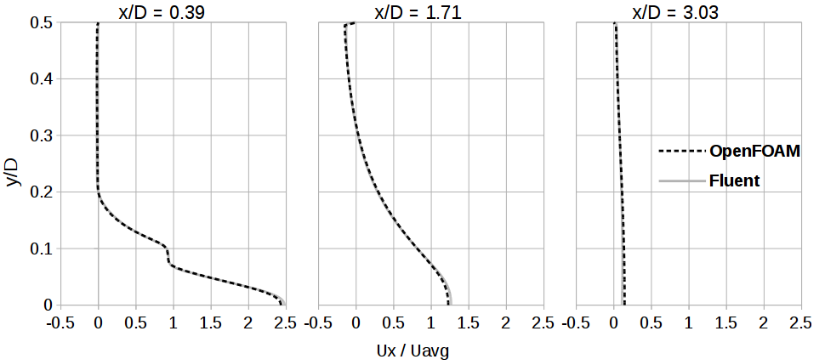

Figure 4.9: Comparison of the axial velocity profiles plotted in the radial direction y at several positions x/D. D is the gas chamber diameter and U_{avg} is the average inlet velocity of 6.8077 ms^{-1}.

4.3.3. CASE 3

For this case, first a comparison is made with a non-reacting flow. Fig. 4.9 shows the agreement of the cold flow velocity profiles predicted by both solvers. A small difference can be noticed in maximum velocity at the center line. For the second comparison, combustion is included without thermal radiation. In the contour plot

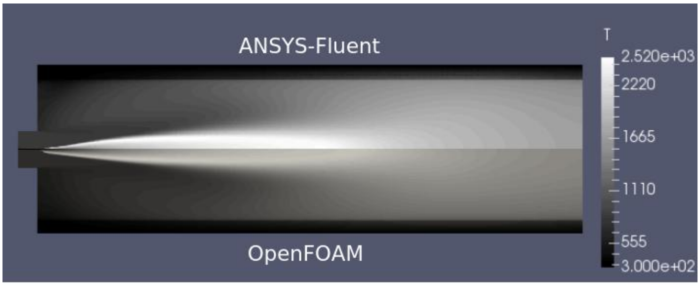

Figure 4.10: Comparison of the contour plots of the temperature (K) in Case 3 (radiation switched off).

of the temperature (Fig. 4.10), similar flame characteristics can be noticed, such as the flame's length and position. Although the maximum temperature differs only by 52 K, the hot spots are located at different regions. This is influenced by the different prediction of the temperature distribution (and probably also by the species and flow speed) in the recirculating gas. However, along the center line, the temperatures are in very good agreement (Fig. 4.12). Because of the flow recirculation, the hot gases are now entrained towards the wall and have a large effect on the heat exchange with the solid. Fig. 4.13 shows the heat transfer along the inner wall surface, which is only due to the convective heat transfer. The heat transfer peaks near the reattachment point, where the flow practically impinges on the surface (see Fig. 4.11). Although the overall behaviour of the heat transfer is predicted similarly by both solvers, the differences increase upstream and downstream of the reattachment point. The probable cause might be due to the subtle differences in the second-order discretization schemes, and in the wall treatments where the turbulent thermal diffusivity α_t is computed implicitly in Fluent rather than explicitly as in OpenFOAM.

When radiation is activated, both solvers predict higher wall heat transfer (Fig. 4.14) and lower flame temperature (Fig. 4.12), where Fluent still predicts higher peak temperature (about 80 K difference). The radiative heat fluxes are aligned very well except close to the inlet and outlet, and this may be caused by Fluent's second-order upwind scheme which is more diffusive than OpenFOAM's linear upwind scheme. Also the WSGG model could play a role, as the `greyMeanAbsorptionEmission` model doesn't take into account the dependence of the beam length and partial pressures of the gas species. The convective heat transfer plays a minor role in both solvers due to the much smaller temperature differences between the gas and solid. Nevertheless, the convective contribution of roughly

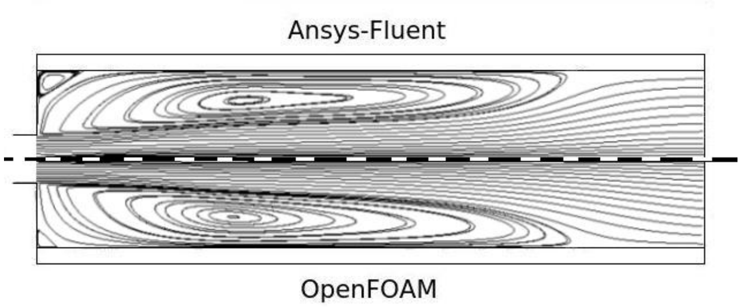

Figure 4.11: Comparison of the stream patterns

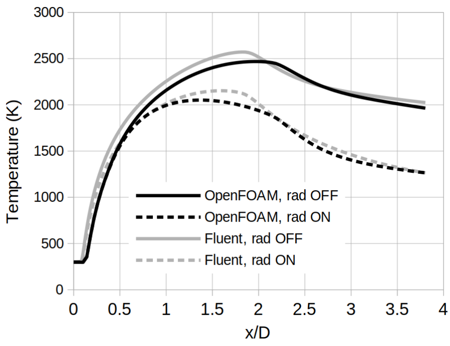

Figure 4.12: Temperature progression along the central axis of the gas chamber. D is the gas chamber diameter 0.76 m.

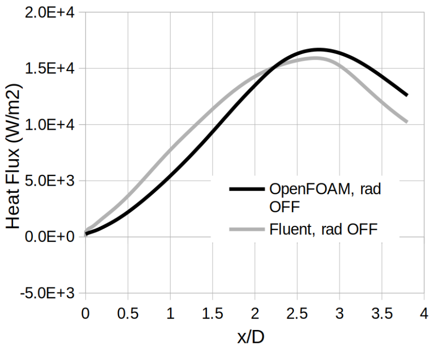

Figure 4.13: Heat flux (radiation OFF) along the inner wall surface of Case 3. q_c is the convective heat flux, which for this case is equal to the total heat flux q_t. OF = OpenFOAM.

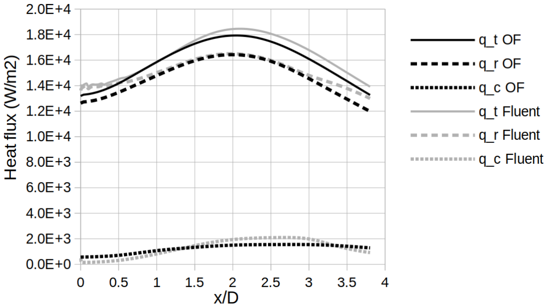

Figure 4.14: Heat flux (radiation ON) along the inner wall surface of Case 3. q_t, q_r and q_c are respectively the total, radiative and convective heat fluxes. OF = OpenFOAM.

10% of the total heat transfer (in accordance with [1]) can certainly not be neglected.

4.4. CONCLUSIONS

This work has shown that OpenFOAM's standard solvers, `reactingFoam` and `chtMultiRegionFoam` are succesfully coupled in the new solver `multiRegionReactingFoam`. This enables the modelling of combustion with conjugate heat transfer. The results of the new solver, with conjugate heat transfer turned off, are identical to `reactingFoam`, which means that the quality of the combustion simulation does not depend on the solver itself.

The quality of the conjugate heat transfer was benchmarked against Fluent, which resulted in good qualitative agreements with convection alone, and quantitative agreements when also radiation is involved.

Since the temperature of the outer wall surface is fixed to a certain value, it is more realistic and recommended to change it to a Robin boundary condition and include radiative and convective heat losses at the outer wall boundary. This allows a more accurate prediction of the heat distribution in the solid and the hot gas mixture.

5

THE EFFECT OF CHT AND EXTERNAL RADIATIVE HEAT LOSS ON THE REACTING FLOW AND WALL

This chapter elaborates on the wall thermal boundary conditions, applied on a two-dimensional axisymmetric rotary kiln, which follows the recommendation of [13] and extends their kiln model. In addition to their work, the target is to include the effect of the refractory wall and the ambient heat losses via thermal radiation. The latter is done by implementing a Robin boundary condition at the outer wall surface. The simulations are conducted in three steps. First, the model without lining and an imposed temperature on the wall is simulated. Next, the lining is added using the CHT formulation. Finally the radiative heat loss to the environment is included. The calculated temperatures and radiative heat fluxes in these three cases are compared and shown in the results.

This chapter is based on the article:

M. el Abbassi, D.R.A. Fikri, D.J.P. Lahaye, and C. Vuik. Non-premixed combustion in rotary kilns using OpenFOAM: the effect of conjugate heat transfer and external radiative heat loss on the reacting flow and the wall. In *ICHMT Digital Library Online. Begel House Inc.*, 2018.

5.1. INTRODUCTION

Rotary kilns are long, slightly tilted, cylindrical furnaces that are used in a wide range of material processing industries, such as cement production, waste treatment, and metallurgy. The primary role of this device is to gradually heat the material that is moving slowly from the cold end, towards the hot end where the burner is positioned [1].

The phenomena of turbulent flow, combustion and heat transfer occurring in the kiln are strongly coupled. The heat released by the turbulent flame is transferred towards the material and the refractory lining. The absorption, emission and reflection of radiative heat influence the flow of gases in the freeboard and the flame characteristics. Moreover, as rotary kilns usually stand outside, a significant amount of heat is lost to the environment. With the increased environmental awareness and growing demand in production, numerical simulation is ideally suited to find optimal solutions in reducing the pollutant formation and specific fuel consumption, while also achieving a desired heat distribution to improve the production rate.

Different aspects of the rotary kiln have been studied using commercial CFD packages. In [30] we see the benefits of flue gas recirculation during oxy-fuel combustion. [27] shows how to prevent ring formation by changing the air-to-fuel ratio. [13] is a good example that shows how CFD can aid in the operation and design of a kiln to meet the specified requirements. However, [13] differs from the first two investigations in that it disregards the influence of the refractory wall and heat loss to the environment. In [14] this influence is shown to some extent, but lacks detail and quantification.

In this work [113], we follow the recommendation in [13] and extend their kiln model. Our target is to include the effect of the refractory wall and the ambient heat losses. We integrate the required physics in a solver by using the library of the open source CFD toolbox OpenFOAM-v4.1. We make use of the developed code `multiRegionReactingFoam` [108] which couples the combustion utility of OpenFOAM with the conjugate heat transfer (CHT) utility. In addition we implement the Eddy Dissipation Model for the turbulence-chemistry interaction and a Robin boundary condition for the external heat loss.

The simulations are conducted in three steps. First, the model without lining and an imposed temperature on the wall is simulated. Next, the lining is added using the CHT formulation. Finally the radiative heat loss to the environment is included. The calculated temperatures and radiative heat fluxes in these three cases are compared and shown in the results.

5.2. SET-UP

5.2.1. GEOMETRY AND BOUNDARY CONDITIONS

Two rotary kiln geometries with their dimensions can be found in [13], of which the so-called 'case A' geometry is selected. In this theoretical full-scale kiln, the fuel methane enters from a circular pipe at the burner head, while the primary air is supplied around the fuel in an annular-shaped channel. The secondary air enters around the burner throughout the cross-section of the kiln. Since the kiln is modelled with steady-state RANS simulations and the flow is (statistically) symmetric, the kiln geometry is reduced to a 2D axisymmetric model. The mesh at the first 5-meter section of the kiln is shown in Fig. 5.1, where the refinement at the burner and at the lining can be seen. The mesh consists of 46,263 control volumes, of which the solution is grid independent. The wall dimensions and properties are shown in Table 5.1. The boundary conditions for the case of 10 MW are shown in table 5.2. The mass flow rates are calculated from the given power, which is based on the fuel input, and the pressure at the outlet is fixed at 1 atm. The air excess ratio of 1.12 is considered, of which 10% of the air enters as primary air. For the 40 MW case, the mass flow rates are quadrupled.

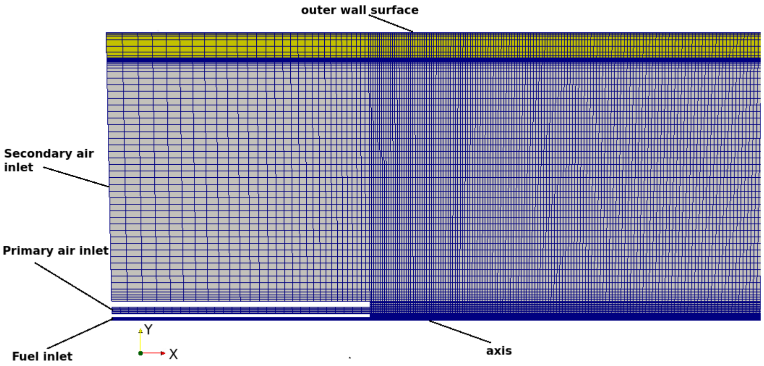

Figure 5.1: First 5 m section of the axisymmetric mesh of the kiln's freeboard region (grey) and 0.2 m thick refractory wall (yellow).

Table 5.1: Thermal properties of the refractory material.

Density ρ	Thermal conductivity λ_s	specific heat capacity c_p	emissivity ϵ_s (Opaque)
$2800\,\mathrm{kgm^{-3}}$	$2.1\,\mathrm{Wm^{-1}K^{-1}}$	$860\,\mathrm{Jkg^{-1}K^{-1}}$	0.6

Table 5.2: Boundary conditions for the kiln model (10 MW). zG stands for the Neumann boundary condition zeroGradient. The patch normal mass flow rates are expressed in kg/s, and the temperatures in K. Species are denoted in mass fractions.

Variable	Fuel jet	Primary air	Secondary air	Gas-wall interface	Outer wall surface
$\dot{m}$ [kg/s]	0.2	0.3896	3.5061	0	-
T [K]	293	293	523	523/Coupled*	523*/$q_{r,ext}$**
Y_{CH_4}	1	0	0	zG	-
Y_{O_2}	0	0.23	0.23	zG	-
Y_{N_2}	0	0.77	0.77	zG	-

*Lining is included. **Lining and external heat loss is included.

5.2.2. PHYSICAL MODELS

The chosen physical models for the solver are listed in Table 5.3. As OpenFOAM doesn't have the non-premixed mixture-fraction-based combustion model which was used in the reference paper we benchmark `multiRegionReactingFoam` with our simulation in ANSYS Fluent where we keep the settings as close as possible to each other (and with the reference paper). The Reynolds stresses are closed with the Realizable k-ε model. The finite-rate EDM was used for the mean species source term, using the single-step Westbrook and Dryer (WD) reaction mechanism. The mean radiative heat source is modelled using the P1 model. The same differences hold as in Sec. 4.2.4 regarding the absorption models and the numerical schemes (OpenFOAM's linear upwind vs Fluent's second-order upwind).

Table 5.3: Selected physical models for the test cases

	Reference paper	OpenFOAM-v4.1.	ANSYS Fluent
Turbulence	Realizable k-ε	Realizable k-ε	Realizable k-ε
Wall treatment	Log-law	Log-law	Log-law
Combustion	Non-premixed β-PDF	EDM	EDM
Chemistry	Mixture fraction	single-step WD	single-step WD
Radiation	P1	P1	P1
Absorption model	WSGG	greyMeanAbsorptionEmission	WSGG

5.2.3. EXTERNAL HEAT LOSS

In order to have a more realistic thermal boundary condition at the outer wall surface, we incorporate heat loss to the environment by introducing a Robin boundary condition on the outer wall surface, using OpenFOAM's `externalWallHeatFluxTemperature` function. It is assumed that there is no wind so that heat loss due to forced advection can be neglected (note that there is freedom to incorporate this as well, using the convective heat transfer coefficient). Referring to Fig. 5.2, the radiative heat loss is calculated as follows.

We define the inner wall temperature at y=0 as $T_{w,\mathrm{i}}$:

$$T|_{y=0} = T_{w,\mathrm{i}}. \tag{5.1}$$

At the outer wall surface (where y=1) we use the Stefan-Boltzmann law to determine the flux, such that

$$\lambda_s \frac{\partial T}{\partial y}\bigg|_{y=1} = q_{r,\mathrm{ext}} = \epsilon\sigma(T_{w,\mathrm{o}}^4 - T_\infty^4), \tag{5.2}$$

where T_∞ is the ambient temperature which is set to 288.15 K (ISA). Using Eqs. 5.1 and 5.2 yields the following relation for the outer wall surface temperature:

$$T_{w,\mathrm{o}} = \frac{\epsilon\sigma}{\lambda_s}(T_{w,\mathrm{o}}^4 - T_\infty^4)y_1 + T_{w,\mathrm{i}}, \tag{5.3}$$

where y_1 is equal to the thickness of the refractory wall. Eq. 5.3 is valid when the interior surface area of the wall is equal to its outer surface area. In case of a kiln, this is a valid assumption in this case.

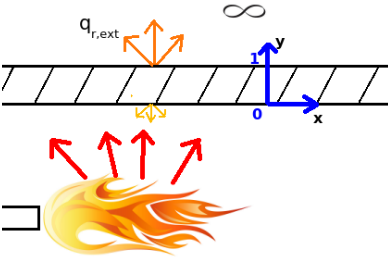

Figure 5.2: Schematic of a confined turbulent flame with radiative heat loss to the environment via the wall.

5.2.4. CASE STUDIES

Three cases are studied in our work. In the first case, both the 10 MW and the 40 MW operating conditions are modelled, and the simulations are repeated with ANSYS-Fluent for verification. In the second case, the refractory lining is added and the constant wall temperature of 523 K is now shifted to the outer surface of the lining ($T_{w,os}$ = 523 K). In the last case, the external radiative heat flux $q_{r,ext}$ is incorporated and $T_{w,os}$ is defined by Eq. 5.3.

5.3. RESULTS AND DISCUSSION

5.3.1. CASE 1: VERIFICATION

In Fig. 5.3 the stream patterns and temperature contour plots are shown both with OpenFOAM-v4.1 and Fluent. The Fig. shows a typical Craya-Curtet flow under the condition where the momentum ratio of the central jet to the co-axial flow exceeds the critical value where the entrainment demands of the jet cannot be satisfied by the coflow, and an annular recirculation zone forms at the wall [114]. Both solvers agree on the maximum temperature and overall temperature distribution. We do see a small difference in both the ignition delay and the total length of the reaction zone (see also Fig. 5.6). The difference is caused by the absorption models, as the `greyMeanAbsorptionEmission` model doesn't take into account the dependence of the beam length and partial pressures of the gas species like Fluent's WSGG model does, and the agreement would be even better if a constant absorption coefficient is chosen. From the aerodynamics perspective, there is also good agreement in the flow field, with small discrepancy in the location of the recirculation zone. Overall, the results are very similar.

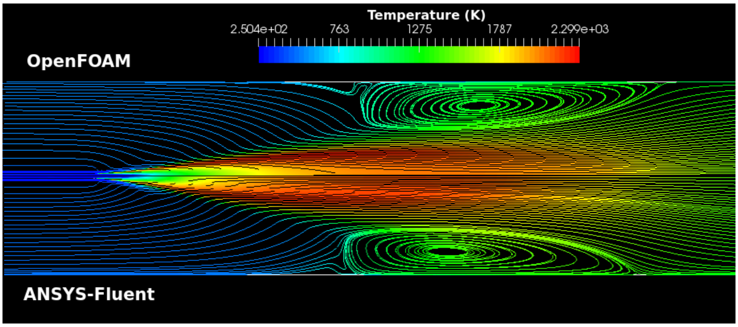

Figure 5.3: Comparison of the stream patterns and temperature at the first 16m section of the kiln. 40 MW.

HIGH-SPEED FLOW EFFECTS IN THE INLET CHANNEL

When modelling the 40 MW case, prescribing a constant velocity at the fuel inlet leads to inaccurate results. A simple estimate assuming atmospheric pressure yields an inlet velocity of approximately Mach 1.6. Under such conditions, the flow becomes choked, as viscous effects and boundary-layer growth cause the fuel inlet channel to behave similarly to a nozzle, with the effective throat located near the channel exit.

The resulting shock-like behavior is indicated by a sharp spike in the axial velocity profile (Fig. 5.4). However, it is not accurately captured due to the ill-posed supersonic inlet condition combined with a `zeroGradient` pressure boundary. Once the flow is choked, the required mass flow rate can only be achieved by increasing the upstream pressure [115], which is reflected in the pressure rise shown in Fig. 5.5. Prescribing a constant inlet velocity prevents this natural upstream pressure build-up and leads to an overprediction of the fuel mass flow rate, resulting in a fuel-rich mixture.

As shown in Fig. 5.6, a velocity inlet produces a narrow peak temperature similar to that reported in [13], whereas prescribing a mass-flow inlet yields a higher and broader temperature profile. Using a mass-flow inlet ensures mass conservation and provides a more physically consistent inlet representation.

As for the 10 MW case, the fuel inlet speed is nearly at Mach 0.4 and therefore the flow is almost incompressible, leading to similar results between a velocity inlet and mass flow inlet (see Fig. 5.7)

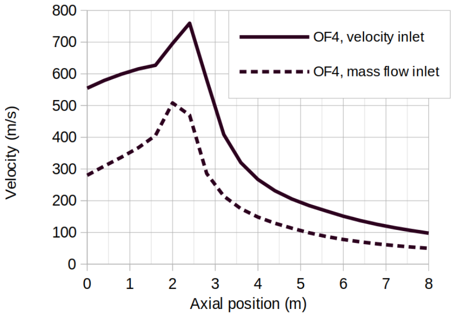

Figure 5.4: Progression of velocity along the central axis. 40 MW.

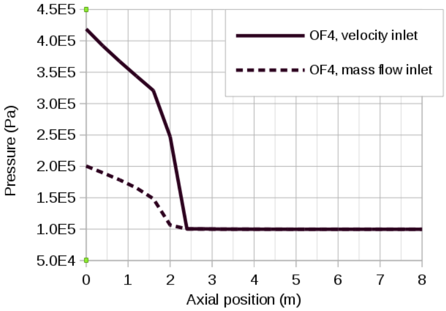

Figure 5.5: Progression of pressure along the central axis. 40 MW.

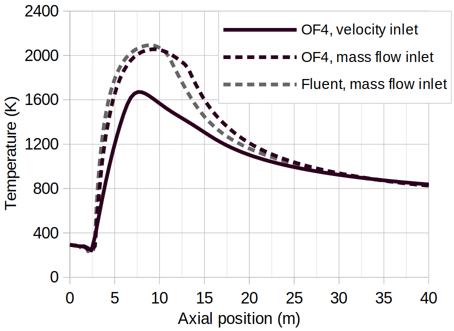

Figure 5.6: Temperature along the central axis in OpenFOAM-v4.1 and Fluent. 40 MW

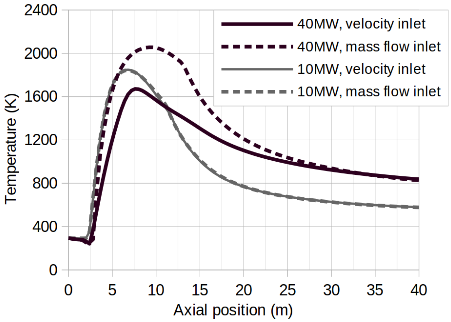

Figure 5.7: Temperature along the central axis in OpenFOAM-v4.1. 10 vs 40 MW.

5.3.2. CASE 2: INFLUENCE OF THE REFRACTORY WALL

All of the following simulations are done with the 10 MW power configuration. Fig. 5.8 - 5.10 shows the significant thermal effects on the gas domain when incorporating the refractory material. The lining provides insulation and leads to a much higher gas temperature (Fig. 5.8), with an increase of 10-90 % at the kiln's central axis. Also, the maximum flame temperature increases by about 12 %, from 2267 K to 2548 K.

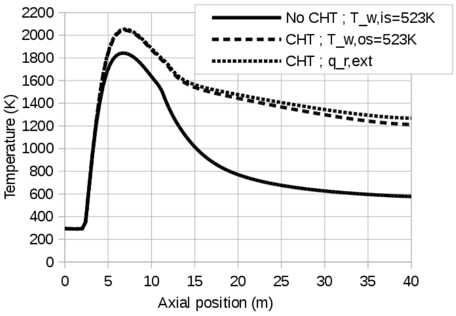

Figure 5.8: Temperature progression along the central axis. 10 MW.

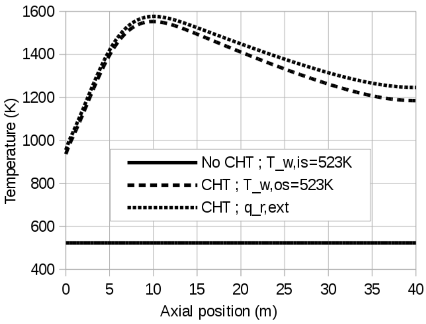

Figure 5.9: Temperature progression along the lining's inner surface. 10 MW.

The inner surface of the lining has a bell-shaped temperature distribution of which the maximum temperature is increased almost threefold (Fig. 5.9), with a 60 % difference between its minimum and maximum. This is in sharp contrast with assuming a fixed temperature at the wall, even if it is scaled up. Due to the lower temperature difference between the flame and the wall, the peak radiative heat flux to the wall decreases with nearly 80% as compared to a constant wall temperature of 523 K (Fig. 5.10). Although the numbers seem drastic so far, they reduce with a better estimation of a constant wall temperature. Nevertheless, the inaccuracies of assigning a constant temperature cannot be neglected.

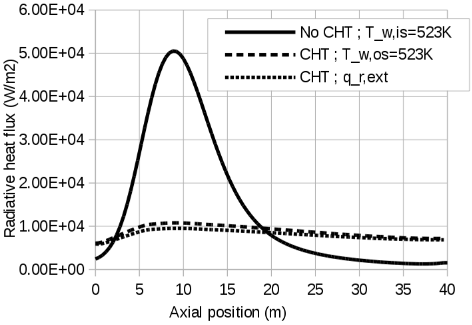

Figure 5.10: Radiative heat flux progression along the lining's inner surface. 10 MW.

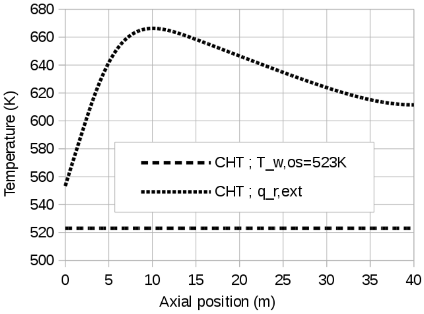

Figure 5.11: Temperature progression along the lining's outer surface. 10 MW.

5.3.3. CASE 3: INFLUENCE OF EXTERNAL HEAT LOSS

When incorporating radiative heat loss to the environment, we also see a bell-shaped temperature distribution at the outer wall surface, with a maximum temperature increase of nearly 30% (Fig. 5.11). In the gas domain, however, the effects are small (Fig. 5.8 - 5.10). This is due to the high thermal resistance of the refractory, in which the thermal effects of one side of the wall do not translate immediately to the other side. However, the influence on the gas domain is not insignificant; both the temperatures of the gas and inner wall surface increase by nearly 60 K at the outlet, and the maximum flame temperature increases by 18 K.

5.4. CONCLUSIONS

The aim of our work is to show the importance of including the energy transport in the refractory lining, and the external heat loss. Compared with the reference model where a constant temperature is assigned to the wall, our model predicts a temperature increase of 10 - 90 % throughout the central axis of the kiln. Our results show a bell-shaped temperature distribution at the inner wall surface, with more than a 200 % increase in the peak temperature and a 60 % difference between the minimum and maximum. Due to the increased thermal insulation, we see that the maximum flame temperature increases by 12 %, and the maximum radiative heat flux to the wall decreases by 80 %. The inclusion of external radiative heat loss causes the outer wall surface temperature to increase by 30%, whereas the effects on the inside of the kiln are much smaller. This work confirms the important role of thermal boundary conditions and has introduced the CHT approach to represent it accurately.

As a final remark, specifically for channel flows, we stress the importance of applying a mass flow rate at the inlet boundaries when the flow cannot be considered incompressible, in order to ensure mass conservation at the inlets.

6

THERMAL NO SOLUTIONS FOR A FULL SCALE 3D ROTARY KILN

Chapters 4 and 5 show different applications modelled with the developed solver `multiRegionReactingFoam`*, in increasing order of complexity. By understanding the solution of the previous geometries, better confidence can be gained in what to expect in a more complex geometry. The aim of this chapter is to apply the solver to a real-life example of an industrial rotary kiln, which is coping with the problem of extremely high* NO_x *emissions and ring formation of the melting material due to the high temperature of the flame. The purpose is to propose three solutions to tackle this problem. However, not only are the burner and inlet geometries more complicated, but it will also be the first 3D geometry. A 3D geometry brings additional challenges; compared to a 2D geometry with the same global dimensions, it requires nearly an order of magnitude more cells, and the coefficient matrix of the linear system of equations is less sparse, which makes it more difficult to converge to a solution. Another challenge is that vortex stretching is introduced, a three-dimensional process and an essential aspect of turbulence, which has an unstable effect on the flow that increases with the Reynolds number.*

This chapter is based on the article:

M. El Abbassi, D.J.P. Lahaye, and C. Vuik. The effect of variable air-fuel ratio on thermal NOx emissions and numerical flow stability in rotary kilns using non-premixed combustion. *Processes*, 9(10):1723, 2021,

and additional work.

6.1. INTRODUCTION

Rotary kilns are long cylindrical furnaces that are horizontally positioned and slightly inclined. A wide range of material processing industries make use of these devices, such as cement production, metallurgy, and waste treatment [1]. Due to the inclination and axially rotating motion of the kiln, the raw material gradually moves from the cold (upper) end towards the hot (lower) end, where the burner is located. A schematic is shown in Fig. 6.1. The primary role of these devices is to gradually heat the material. We study its application in the cement industry, where the material bed is mixed, heated and subjected to sintering reactions to form clinker. The material leaves the furnace at the lower end to be further processed. Sufficiently high temperatures are required to heat the material to reach the sintering reactions, with flame temperatures exceeding 2000 K. This makes the process prone to the formation of thermal nitric oxides (thermal NO).

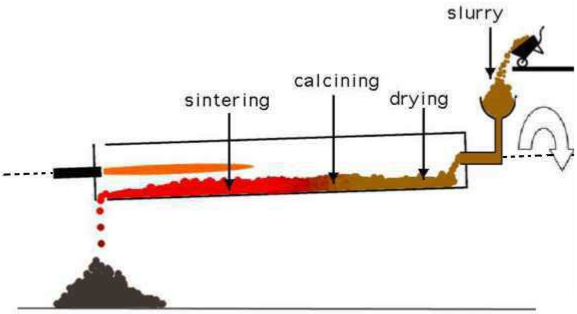

Figure 6.1: Cross section of a rotary kiln. [2]

The aim of this chapter is to propose three solutions to reduce the thermal NO emissions of an actual operating kiln with a natural gas burner. The *first* solution proposal (based on our paper [116]) is to change the air inlet geometry in order to create a better flow structure and more optimal mixing conditions for the non-premixed flame. However, it does require rebuilding the inlet side of the kiln and reinstalling the combustion air infrastructure.

An easier solution would be to modify the operating condition, which is a matter of turning the valves. This brings us to the *second* solution proposal, which is based on our paper [117]. The proposal is inspired by [27], where it was shown that increasing the volumetric air-fuel ratio (AFR), defined as the ratio of the *volumetric* flow rates of air to fuel, leads to both lower flame temperature and wall temperature, thus preventing ring formation. The lower flame temperature indicates that thermal NO formation may also drop, though it is not that simple because more oxygen and nitrogen are provided as well. This is investigated both numerically and experimentally in this chapter.

Finally, our *third* solution proposal is to apply Exhaust Gas Recirculation (EGR), which is an NO_x reduction combustion technique. Like most other NO_x reduction techniques, the goal is to reduce thermal NO emissions by lowering the flame's temperature. When recycling a portion of the exhaust gases to mix with the secondary air inlet, the oxygen concentration is lowered by the inert combustion products, such as CO_2 and H_2O. This leads to a less sharp temperature rise at the flame front and therefore a lower peak temperature. The combination of low temperature and low oxygen concentration prevents thermal NO formation.

For all simulations in this chapter, we use the developed code `multiRegionReactingFoam` [108] in OpenFOAM version 5.0. This solver couples the conjugate heat transfer (CHT) utility of OpenFOAM with the combustion utility. Additionally, the Eddy Dissipation Model is implemented for the turbulence-chemistry interaction, and the P1 approximation is employed for modelling thermal radiation. Thermal NO will be calculated with the post-processor (Sec. 2.4), and for the external radiative heat loss to the environment, we implement the Robin boundary condition (Sec. 2.6.5).

Sec. 6.2 will start off with explaining the numerical setup and solver settings for the actual operating kiln, which is used as our reference. This is followed by the results and discussion. Afterwards, each of the proposed thermal NO mitigating solutions is presented in a subsequent section. Sec. 6.3 will discuss the improvement of the secondary air inlet design, and Sec. 6.4 will study the effect of varying AFR. Finally, the effect of EGR is presented in Sec. 6.5.

6.2. SIMULATING THE ORIGINAL KILN GEOMETRY

6.2.1. PHYSICAL MODELS

The chosen physical models for the solver are summarized in Table 6.1, and are listed next to models in the reference paper for comparison. The goal is not to achieve identical results but the difference in model selection will give insight in the result outcomes.

Table 6.1: Selected physical models for the test cases

	Reference paper [27]	OpenFOAM v5.0
Turbulence	Realizable k–ε	Realizable k–ε
Wall treatment	Log-law	Log-law
Combustion	Standard Eddy Break-Up (EBU)	Eddy Dissipation Model (EDM)
Mechanism	6 species, 4 reactions	Single-step Westbrook-Dryer
Radiation	DOM	P1
Absorption model	Participating gas (non-grey)	greyMeanAbsorptionEmission
Wall heat loss	CHT + (unknown) forced convection	CHT + external radiation

6.2.2. GEOMETRY

The geometric model of the kiln is shown in Figs. 6.2 to 6.4, which give an exterior view of the complete kiln and a more detailed interior view of the burner region, respectively. This is the actual operating kiln, and its geometry is referred to as geometry A. (see Fig. 6.24). The burner consists of 16 fuel inlets, which are surrounded by the annular-shaped primary air inlet to cool the burner. The vast majority of the air (90%) enters from the rectangular-shaped secondary air inlet, which is preheated. The secondary air inlet breaks the axisymmetry of the kiln and flow; hence, it has to be modelled in three spatial dimensions.

The material bed only occupies a small fraction of the volume of the kiln and has a negligible limited impact on temperature distribution. Therefore the material bed is not taken into account in the model, and an empty kiln is simulated. Another simplification is that the refractory lining does not rotate for the time being.

The challenging aspect of this geometry is that the inlets of the burner are three orders of magnitude smaller than the axial length of the kiln, imposing challenges in the mesh generation process.

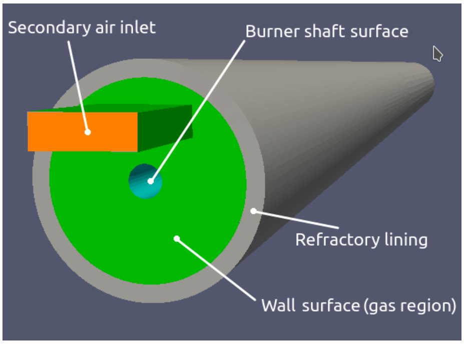

Figure 6.2: Outside geometry of the kiln

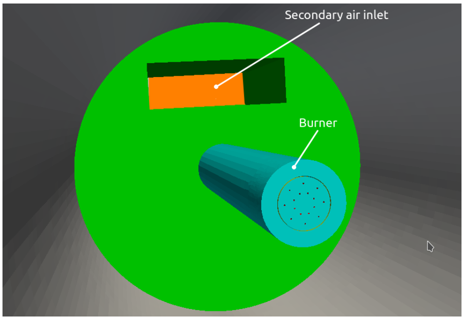

Figure 6.3: Inside geometry of the kiln

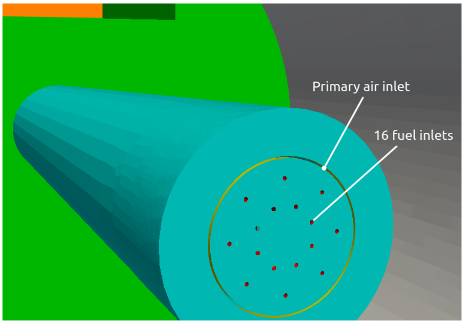

Figure 6.4: Geometry of the burner

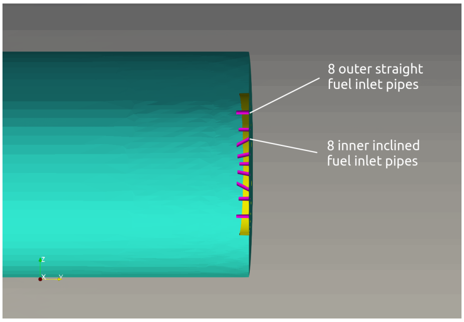

Figure 6.5: Burner head cross section

6.2.3. MESH

The mesh model is shown in Figures 6.6 to 6.8, which give an exterior view of the mesh in the air inlet region, an interior view of the mesh and a detailed view of the mesh on the burner, respectively. In order to visualise the local refinement better, the coarse mesh is shown.

Due to the very large difference in size between the kiln and fuel inlets, a structured grid is not a good option. Not only will the cells become extremely stretched along the rotational axis (where the flow is definitely not one-dimensional), but the local refinement at the burner is inherited in the entire domain. This leads to an excessive amount of volume elements, or cells. In order to reduce the amount and preserve the quality of the cells, an unstructured grid is required, or multi-block structured meshes.

Tetrahedra are the simplest type of volume elements. As their faces are plane segments, both face and volume centroid locations are well defined. They can be applied to any complex geometry without losing the quality of the individual cells, with little effort. A disadvantage is that tetrahedra cannot be stretched too much. To achieve a reasonable accuracy, a much larger number of control volumes is needed than if hexa- and polyhedral meshes are used. Furthermore, computing gradients at cell centers using standard approximations can be problematic, as tetrahedral control volumes have the least amount of neighbours, and given that they are not orthogonal.

For this kiln, the cartesian mesh is employed with about 4 million cells. This is an orthogonal body-fitted mesh. It consists of predominantly hexahedral cells with polyhedra in the transition regions between the cells of different sizes (the border of a certain refinement level), and prisms to connect with the geometry (or its wall refinement) which is not alligned well with the grid. The cartesian mesh combines the good properties of different mesh types. The mesh is unstructured, which disconnects the burner inlets from the rest of the domain, and allows local refinement without deforming the cells too much.

The mesh is generated with cfMesh, which is a cross-platform library for automatic volume mesh generation based on OpenFOAM. It requires a geometry file and a script with at least the maximum cell size defined. In addition, the user can add where and how many times the local refinement should occur. By making use of this script, generating a mesh becomes trivial for the user.

6.2.4. NUMERICAL AND SOLUTION METHODS

The simulations are run in steady-state, and therefore the unsteady terms disappear and only the spatial terms are discretised. The central differencing scheme is used

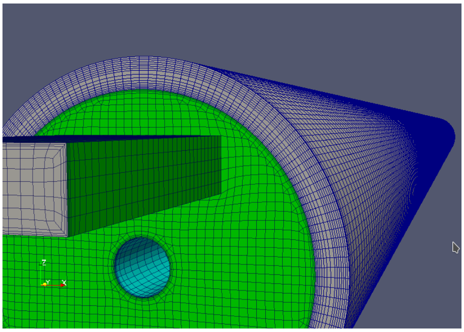

Figure 6.6: Outside surface mesh of the kiln

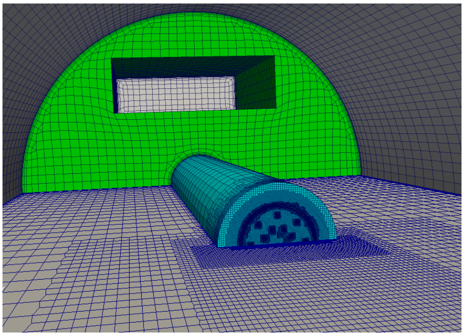

Figure 6.7: Inside mesh of the kiln

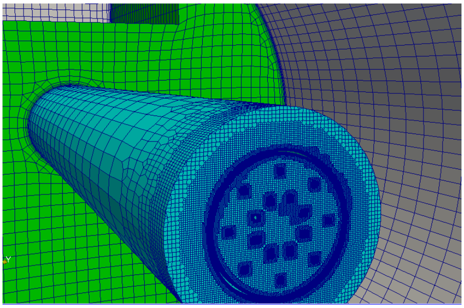

Figure 6.8: Surface mesh of the burner

for the gradient and Laplacian terms. The advection terms for velocity, enthalpy, chemical species, and radiation intensity are discretised using the second-order upwind scheme, while for the kinetic energy and energy dissipation, this is done with the first-order upwind scheme to prevent OpenFOAM from crashing.

The generated linear system of equations is solved as follows. The mass fluxes are solved using the Preconditioned Conjugate Gradient (PCG) method with the diagonal incomplete Cholesky factorisation as a preconditioner, while the equations for the pressure correction and radiation intensity are solved with the same solver but with the geometric-algebraic multi-grid (GAMG) preconditioner. The remaining independent variables are solved with the Preconditioned Biconjugate Gradient Stabilised (PBiCGSTAB) method, using the diagonal incomplete LU factorisation as a preconditioner, which is suitable for non-symmetric sparse matrices caused by the advection terms.

Some important simplifications are applied to the simulations. As the inclination angle and kiln's rotational speed are very low, they are excluded. Due to the complexity of implementing a model for the sintering reactions of the material bed, this is excluded as well. As the material bed only accounts for 5% of the total kiln's volume, with a residence time of 30 minutes or more, it will have limited impact on the reactive flow.

Hexahedral dominant cartesian meshes are generated for the kiln geometries using cfMesh [118], consisting of 4.2 million to 4.4 million cells depending on the secondary air inlet geometry. As the meshes aren't completely orthogonal, an additional non-orthogonal corrector step is required in the SIMPLE algorithm.

6.2.5. SIMULATION STRATEGY

The simulations run until convergence is observed based on 2 criteria. Firstly, the residuals have to converge at their own lowest tolerance levels (minimum 10^{-9}) where they oscillate. Secondly, convergence needs to be observed with the maximum gas temperature, maximum solid temperature, average outlet temperature and average outlet CO_2 fraction (Figs. 4.3 and 4.4).

Simulations have shown that it is very difficult to achieve a steady-state solution for this case. When looking at the previous results in [27] and the position of the rectangular secondary air inlet, experience tells us that a steady-state solution does not exist for this configuration. Unlike the cases in the previous chapters, starting from a guess as the initial condition, or even a solution of a non-reacting flow, does not lead to a converged solution. Although it is possible for scalar quantities such as temperature and chemical species to stabilise at the outlet, the solution of the flow does not converge. Fig. 6.9 shows an example plot of the residuals of the field variables. Notice that they do not drop anymore and strongly fluctuate around a certain magnitude. It becomes more clear that the flow does not converge when looking at the drastically changing stream pattern at different iterations (Fig. 6.13). The instability might be due to the nature of the flow. However, it is clear from the nonphysical result of the wall temperature (Fig. 6.10) that the simulation did not go well when compared to the result of [27].

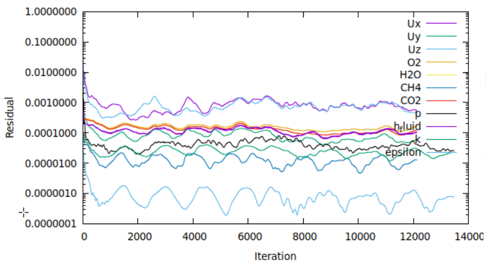

Figure 6.9: Example residual plot of an unconverged solution

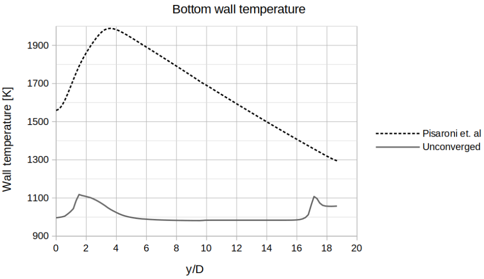

Figure 6.10: Wall temperature of an unconverged solution vs [27]

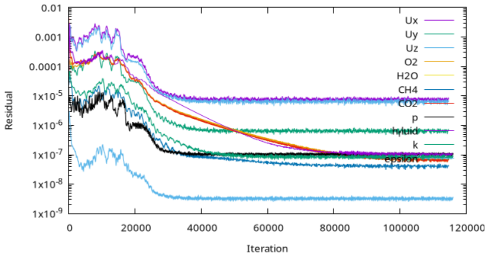

Figure 6.11: Example residual plot of a converged solution

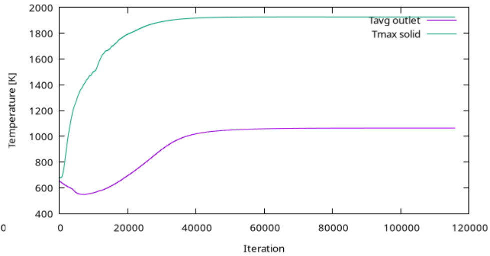

Figure 6.12: Example monitor plot of a converged solution

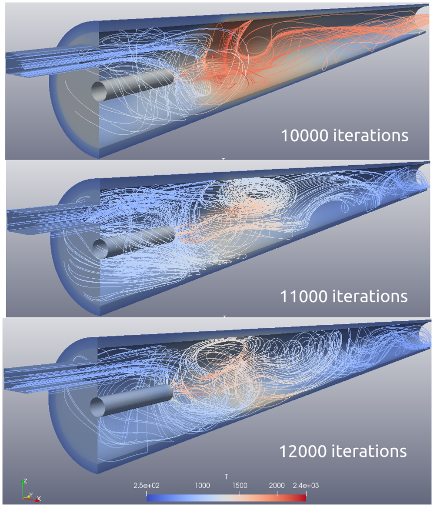

Figure 6.13: Different stream pattern snapshots of an unconverged solution

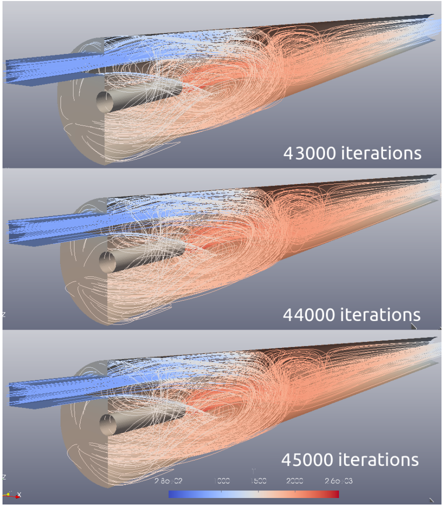

Figure 6.14: Different stream pattern snapshots of a converged solution

The strategy to cope with the multiphysics and complex flow is to conduct the simulation in a stepwise approach, which is done as follows. Firstly, only the gas region is solved, and the effect of the solid region is accounted for using the Robin boundary condition (section 2.6.5). Secondly, the non-reacting flow is simulated to allow the reactants to flow in first. This is used as a starting solution when combustion and thermal radiation are activated.

As a result, notice in Fig. 6.12 that the residuals transition from unstable fluctuations to stable oscillations. Fig. 6.14 shows that the streamlines hardly change every 1000 iterations. These are signs that tell us that the solution has indeed converged.

The figure shows a typical Craya-Curtet flow under the condition where the momentum ratio of the central jet to the co-axial flow exceeds the critical value where the entrainment demands of the jet cannot be satisfied by the co-flow, and an annular recirculation zone forms at the wall.

6.2.6. RESULTS AND DISCUSSION 3D

As a first analysis, the reactive flow simulation is shown here and compared with the results in [27], where the mixture has a volumetric air-fuel ratio of 9.

AERODYNAMICS

To understand the results of temperature and emissions better, we investigate the stream pattern first in Fig6.15, which qualitatively agrees well with [27]. The pattern can be decomposed into four streams.

- Stream 1: the high-momentum primary jet stream leaving the burner.
- Stream 2: the secondary co-flow stream entering from the rectangular inlet at a much lower velocity.
- Stream 3: upper recirculation zone due to Craya-Curtet flow.
- Stream 4: lower recirculation zone due to backward-facing step.

We observe that all four streams interact with each other. The interactions of streams 1, 2, and 3 are similar to those in a Craya-Curtet flow. Stream 2 slows down due to the adverse pressure gradient downstream and is being sucked towards the high momentum of stream 1, where the static pressure is lower, causing stream 2 to detach from the wall. When stream 2 bends downwards, it separates into two directions. Part of the stream is entrained by stream 1, flowing towards the kiln outlet, while the other part turns upstream towards stream 4. Stream 4 is a recirculation zone that exists due to the geometry of the kiln, which is similar to a backward-facing step. The recirculation zone in stream 3 results from the detachment of stream 2 and causes stream 1 to bend downwards. The flow of stream 3 that escapes the recirculation zone is either entrained by stream 1 or by stream 2 towards stream 4. In order to reach stream 4, however, it has to go around stream 1. The recirculation zone of stream 4 is very unstable. The reason for this is that the flow coming from streams 2 and 3, and from the vortex itself, is severely disturbed when flowing around the burner jet of stream 1.

EFFECT ON THE TEMPERATURE

Fig. 6.16 shows the vertical axial cross-section of the computed temperature near the flame region. A zone of high temperature is shown, which approximates the location and shape of the flame. This zone corresponds to a thin reaction zone where

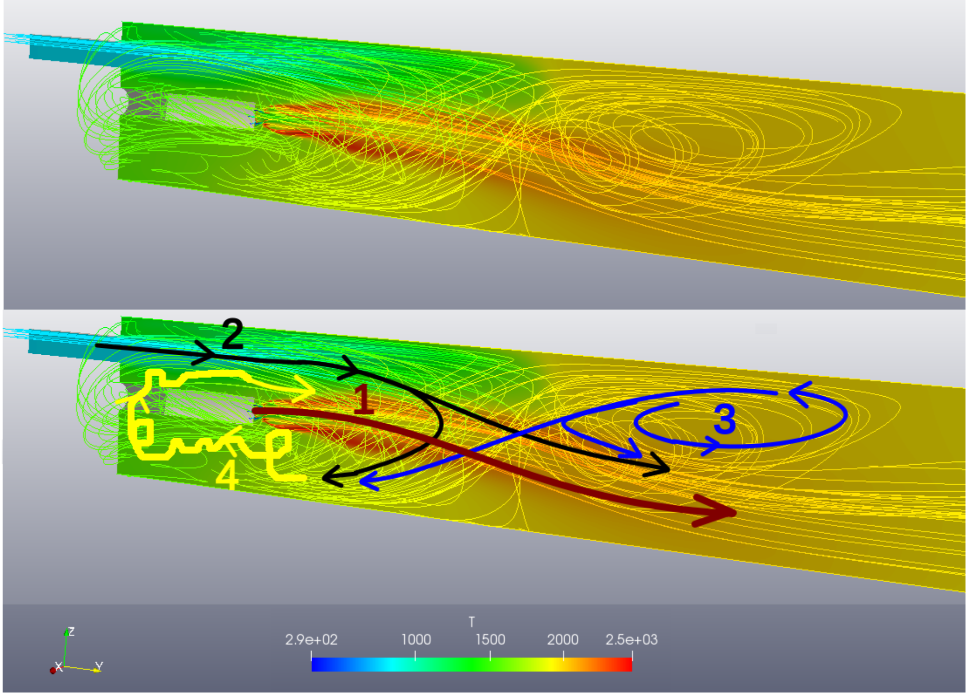

Figure 6.15: Stream pattern and temperature contour plot of the kiln

the fuel and oxidiser meet. The reaction zone can be seen to bend downwards along stream 1. Due to the fact that stream 2 detaches from the wall and bends downward, there is a sudden drop in temperature of the secondary air flow near the end of the flame. This corresponds to the border of the cold stream 2 and the much hotter vortex of stream 3. The results of the temperature contour plot agree very well [27], reaching even the same maximum temperature. One difference is that in OpenFOAM, the hottest region is located at the lower side of the flame near the flame foot. Also, when comparing the heat transfer to the wall, there is a discrepancy in the temperature distribution (see Fig. 6.17). The maximum temperature is located at the same axial distance, and from that point the discrepancy starts to increase linearly towards the end. This is due to the inconsistent physical model selection and wall boundary conditions (Table 6.1), where an unknown forced convection has been applied on the outer wall in the reference paper instead of radiation heat loss.

EFFECT ON THERMAL NO EMISSIONS

Fig. 6.18 shows the computed thermal NO on an axial plane close to the burner. The figure shows that the thermal NO concentration has a peak at the lower part of

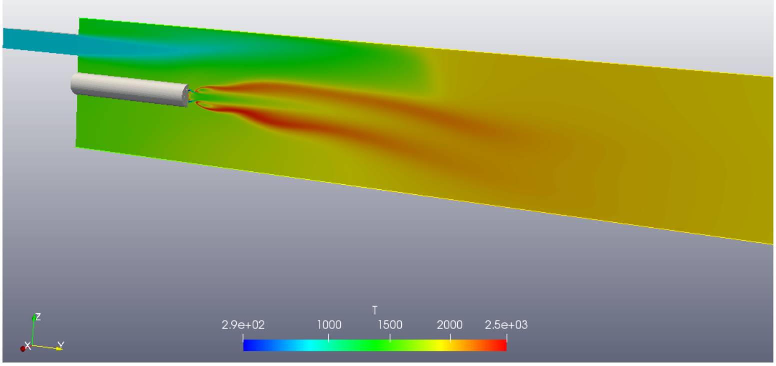

Figure 6.16: Temperature contour plot on the vertical longitudinal cross-section

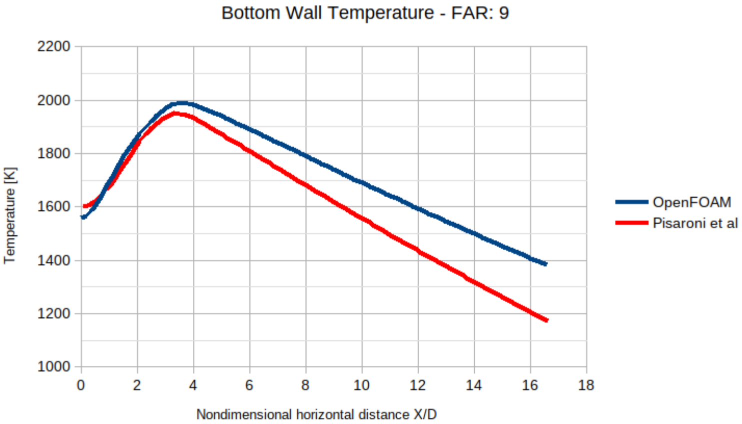

Figure 6.17: Temperature plot along the bottom wall surface in axial direction

the flame close to the burner. This peak corresponds to the peak in temperature at the same location shown at the top of Fig. 6.16. This temperature peak is caused by the recirculation of hot stream prior to reaching the lower part of the flame. The recirculation entrains flow from streams 2 and 3, which pass through the hot reaction zone and are additionally heated by the hot products before reaching the lower part of the flame. This ensures the stabilisation of the flame but leads to very high NO production as well.

6.3. SOLUTION PROPOSAL 1: ALTERNATIVE AIR INLET

By studying the aerodynamics in Sec. 6.2.6, the influence of the secondary air inlet geometry on the NO formation is better understood. The rectangular inlet causes the flame temperature to be unevenly distributed and hence also the thermal

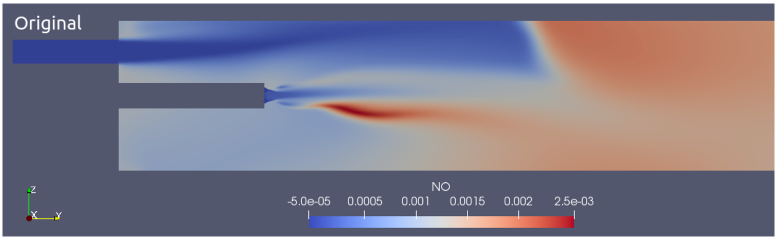

Figure 6.18: Thermal NO contour plot of the kiln

NO concentration, which grows exponentially with temperature. In this section the air inlet geometry is modified in such a way that it will result in a more evenly distributed temperature and hopefully avoid the local hot spot, which is caused by streams 1 and 3 (Fig. 6.15). Our choice is to replace the rectangular air inlet in the geometric model by an annular inlet with a diameter of one-third of the kiln's diameter. The mesh of this new geometry B is shown in Fig. 6.19. The figure shows the multi-nozzle burner and cooling slot in the foreground and the annular secondary air inlet channel in the background. The lower half of the computational domain is filled with the volume mesh.

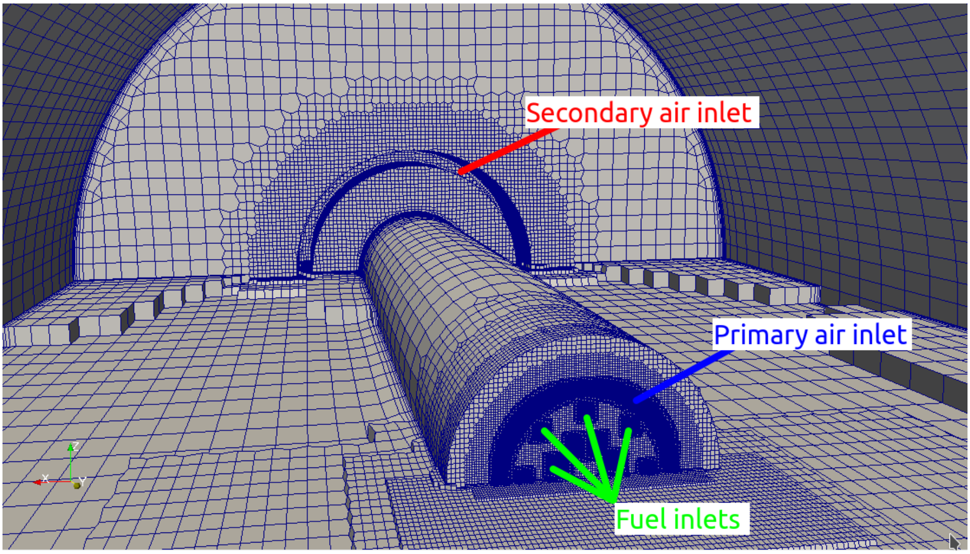

Figure 6.19: Mesh inside of the kiln of geometry B with modified (annular) secondary air inlet. The figures show the multi-nozzle burner and cooling slot in the foreground, the burner pipe and the secondary air inlet channel in the background. Only the lower half of the kiln's internal volume grid is visible (some unclipped cells appear out of the plane) to show the refinement regions. For increased visibility, a coarse mesh is shown here.

6.3.1. RESULTS

Let's compare the results of the alternative secondary air inlet with the original by looking at the stream pattern first. Fig. 6.20 shows the computed streams for the original rectangular (top) and modified annular (bottom) inlets. Different recirculation patterns can clearly be seen. The complex flow of the original geometry is explained in detail in section 6.2.6. In the case of the annular air inlet, we see the more familiar annular-shaped recirculation zone that we expected, which encircles the burner pipe and flame. This stable recirculation zone consists of two vortices. The most upstream vortex arises like in any confined jet flow, while the downstream vortex is generated by the Craya-Curtet flow condition. In the lower part of the recirculation zone, we see that these two vortices are separated, but in the upper part, they are merged into one.

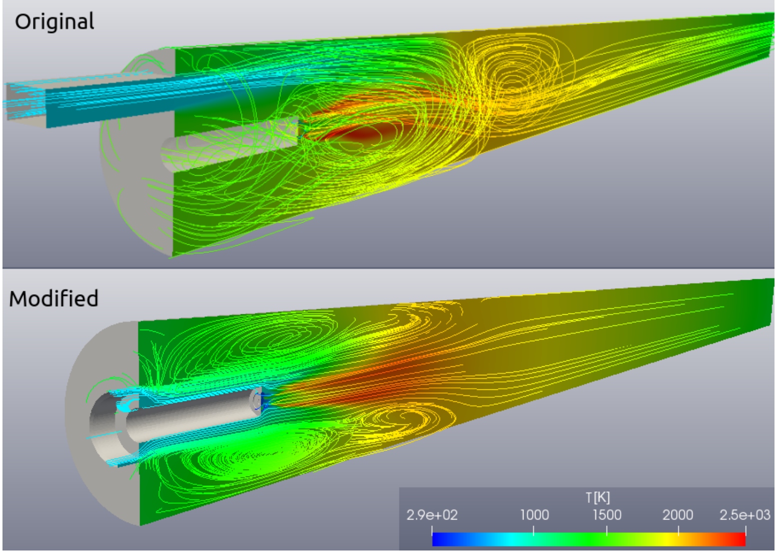

Figure 6.20: Computed streamlines coloured with temperature values for the original rectangular (top) and modified annular (bottom) secondary air inlets. For both cases, a hot zone due to the flame and the recirculation due to the secondary air can be seen.

Due to the axisymmetric nature of the flow in the modified kiln, the computed temperature is more homogeneously distributed throughout its rotational axis, as can be seen in the vertical axial cross-section in Fig. 6.21. The combustion air is supplied more homogeneously around the burner by the annular inlet, and the gas temperature is averaged out above and below the burner, as compared to the original kiln. With the created annular recirculation zone, we avoid that the combustion air will pass through the hot reaction zone and be additionally heated before reaching the lower part of the flame. This leads to the disappearance of the local hot spot below at the foot of the flame and a lower overall peak temperature. The reaction zone also remains in the center of the kiln and is therefore farther away from the bottom of the kiln, where the material bed lies, which can be beneficial as well.

The thermal effect on the refractory lining can be seen in Fig. 6.22, which shows the temperature on the interface between the freeboard gases and the refractory lining. The solid and dashed dark lines correspond to the bottom and top walls in the rectangular configuration, respectively. The grey lines correspond to the annular inlet. All four graphs show a peak near the end of the flame and a linear decrease beyond the peak value. In the case of the rectangular inlet, the peak values of the bottom and top walls differ more than in the case of the annular inlet. The peak values in the case of the annular inlet are lower than in the case of the rectangular inlet. At the top, the annular inlet yields the same peak temperature as the rectangular air inlet. Further downstream, the temperature of the annular inlet is slightly lower. All these observations can be explained by the fact that in the case of the annular inlet, the flame temperature is lower and more uniformly distributed, and the heat is mainly transported via radiation. The results for the rectangular air inlet are in good agreement with Fig. 7 in [27]. It shows that the annular air inlet can be beneficial in counteracting ring formation that is being caused by the unwanted melting of the bed material.

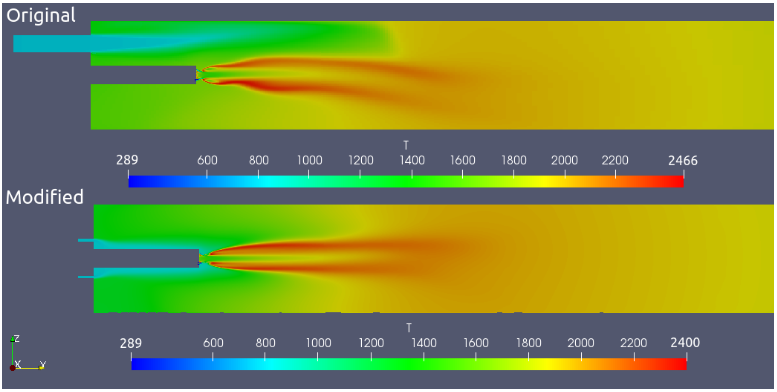

Figure 6.21: temperature on the central axial slice for the original rectangular (top) and modified annular (bottom) secondary air inlet. An air-fuel ratio equal to nine is used. Different scales are chosen to highlight the maximum temperature.

The reduction of the flame temperature also has a direct effect on the thermal NO formation. Fig. 6.23 shows the computed thermal NO mass fraction, again on the vertical axial cross-section close to the burner for the rectangular (top) and annular (bottom) secondary air inlets. The figure shows that the peak of the thermal NO concentration where the hot spot is located (top of Fig. 6.21), disappears as well in

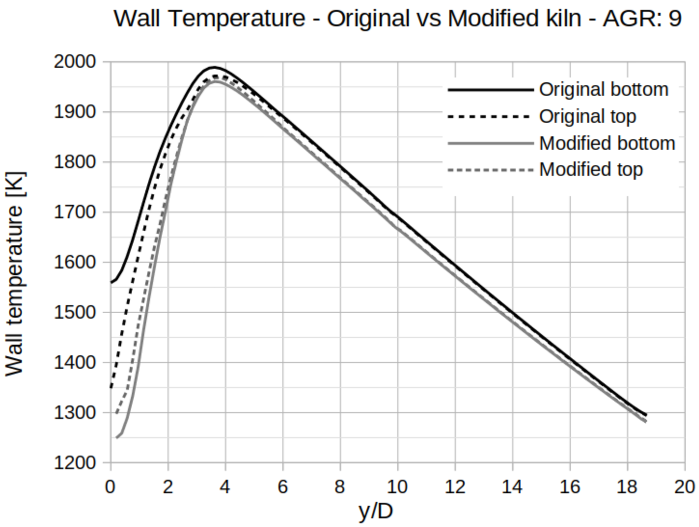

Figure 6.22: Computed top and bottom wall temperature along the axis of the kiln with the original rectangular and modified annular secondary air inlets. Air-fuel ratio: 9.

the co-axial configuration. The annular air inlet results in a significant reduction of thermal NO formation.

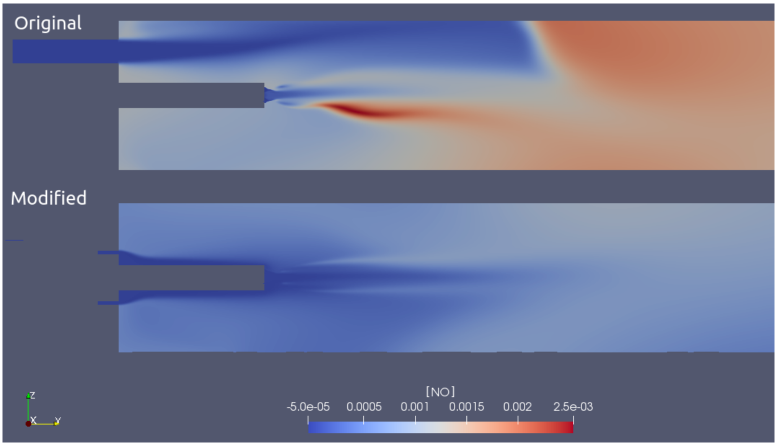

Figure 6.23: Contour plot of the computed NO mass fraction near the burner on the central axial slice for the original rectangular (top) and modified (bottom) secondary air inlet.

6.4. SOLUTION PROPOSAL 2: VARIABLE AIR-FUEL RATIO

Changing the air inlet geometry shows promising results in reducing both the peak temperatures and NO emissions. However, it does require rebuilding the inlet side of the kiln and reinstalling the combustion air infrastructure. A less radical solution would be to modify the operating condition, which is a matter of turning the valves. In [27] it was shown that increasing the AFR from 9 to 12 leads to both lower flame temperature and wall temperature, thus preventing ring formation. The lower flame temperature may also reduce thermal NO formation. Therefore, we investigate the effect of variable air-fuel ratio in this section.

The normalised version of the AFR, the air-fuel *equivalence ratio*, is generally associated with premixed flames and has the widely known parabolic relation with the adiabatic flame temperature, where the maximum temperature occurs slightly on the fuel-lean side of the stoichiometric equivalence ratio (>1). As for diffusion flames, such as in rotary kilns, the reactions take place in the flame front where the fuel and oxidiser occur in stoichiometric ratios, leading to maximum adiabatic temperature, regardless of the equivalence ratio [42]. Which is why the mixture fraction is a better-suited parameter for diffusion flames in general. The AFR however, is something that the operator can control, unlike the mixture fraction. However, unlike premixed flames, it is less obvious for diffusion flames whether NO emissions will increase or decrease with the AFR, and it depends on the type of diffusion flame application, as is recently shown in [119]. Another study is done in [120] on confined diffusion flames, where it is shown that depending on the confinement ratio, the AFR either has a small or large impact on NO emissions. However, the range of the AFR is small and is on the fuel-rich side, while we aim for the fuel-lean side.

There are some studies applied to rotary kilns where, partially, the influence of the AFR on NO formation is investigated. In [121], both experimental and numerical works (using a chemical reactor model) are done on a small pilot scale and a full-scale kiln. In [122], experimental work is conducted on the precalciner, whereas the kiln is experimentally simulated. In both mentioned studies, the NO formation increases with the AFR on the lean side (excess air). This has to do with the fact that they are using coal as an energy source, which contains a large amount of fuel-bound nitrogen, leading to fuel NO_x formation to dominate with higher excess air.

No CFD related study has been done thus far on the effect of the AFR on thermal NO emissions in rotary kilns using natural gas as fuel. Therefore, the focus of our work is to find out how the temperatures and thermal NO emissions are affected by the AFR, as this is a relatively simple way to influence both outcomes.

6.4.1. FLOW INSTABILITY AT HIGH REYNOLDS NUMBER

The AFR is increased by increasing the secondary air inlet mass flow rate. Although we get a steady-state solution at AFR of 9, this only lasts until we increase the AFR to 10. Beyond the AFR of 10, the flow becomes unstable, and the solution procedure is not converging. On the other hand, if we reduce the AFR to values lower than 9, the flow remains stable and the solution procedure converges sooner.

Fig. 6.24 shows the stream pattern on the vertical cross-section for several geometries and for a range of AFRs. In the left column we see the stream pattern for the real kiln geometry (geometry A). We see that the upper recirculation zone is pushed further downstream by the increasing momentum of the secondary air inlet that increases with the AFR. Beyond an AFR of 10 this vortex breaks and the flow becomes unstable, for which no steady-state solution can be found. The instability beyond an AFR of 10 has shown to be a problem in studying the effect of AFR, as we are interested in higher ratios as well.

To solve this issue, we use a modified air inlet geometry in order to have a stable, axisymmetric, and predictable flow (geometry B). With an annular inlet we expect that the concentric recirculation zone will remain stable with higher AFRs as well, since there is no collision between the large recirculation zone and the jets coming from the burner and secondary air inlet. Surprisingly, the flow becomes unstable as well at AFRs beyond 9 and without convergence of the simulation. Even at an AFR of 9, the simulation took much longer to converge compared to the original geometry. Observing the stream pattern snapshots in Fig. 6.24 (2nd column), the dynamic and asymmetric behaviour of the recirculation zone can be seen at increasing AFR. We see a varying elongation and shortening of the recirculation zone. The number of vortices that the recirculation zone consists of alternates between one and two during the shortening and elongation, respectively.

UNDERSTANDING THE PURPOSE OF RANS AND THE MEANING OF STABILITY

For both the original and modified geometries (A and B), we see a trend that the higher the Reynolds number (which is proportional to AFR), the more instability occurs. This of course makes sense as the flow is getting more and more turbulent and the dampening of the viscous forces gets smaller. For pipe flows, the flow is completely turbulent at Reynolds numbers Re_D above 4000. Re_D in the kiln (evaluated at the outlet) is about 30,000, which is considered highly turbulent.

What is meant by unstable flow when turbulent flow is inherently unstable? If Re_D is below 4000, the flow is laminar, which is truly stable in the sense that the flow does not change in time, and therefore a steady-state solution exists. When the laminar flow is disturbed for any reason, it will 'stumble' and a chain reaction of

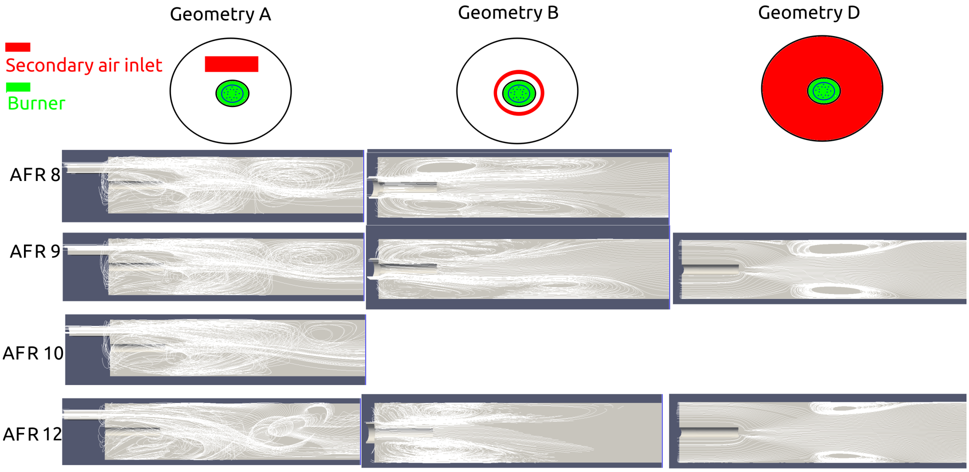

Figure 6.24: Stream patterns at different air-fuel ratios for the original kiln (geometry A), the modified kiln with annular air inlet (geometry B), and the modified kiln with co-flow air inlet over the entire lateral cross-section around the burner (geometry D).

flow eddies is caused, which becomes chaotic, leading to turbulent flow. The flow is now changing over time, and a steady-state solution does not exist. The Reynolds number tells us that the higher its value becomes, the sooner the laminar flow will transition to turbulent flow and the larger the spectrum of length scales of the eddies one has to solve in time [123][124].

The RANS method is a mathematical method that aims to compute ensemble-averaged flow fields in order to obtain the steady-state property of laminar flow back because it saves much time. What it physically does is it makes the flow more viscous in order to dampen the turbulent fluctuations so that the flow will look laminar again, or as we can call it, statistically stationary. This is done by adding the turbulent viscosity μ_t to the laminar viscosity in the momentum equation. This method is justified by the fact that the time average of turbulent flow also looks laminar. However, the dampening effect of making the flow more viscous has a limit. At a certain point, or to be more precise, at a certain high enough Reynolds number, large mean velocity gradients lead to high turbulence production and dissipation rates. The steady RANS solution requires a delicate balance between convection, turbulence production, turbulent diffusion and dissipation. As Reynolds number increases, these competing terms grow in magnitude, making the nonlinear system increasingly stiff and sensitive to numerical perturbations [125]. This is especially the case in flows with strong swirl and recirculation. The sensitivity to disturbances becomes so large that even the uniqueness of the mean field will not exist. Or, as we can call it, statistically unstable.

This trend that we see with flow instability and the Reynolds number helps us to understand the results better. We see that compared to non-reacting flow, stability and convergence improve with reacting flow because the average Reynolds number, e.g. at AFR of 9, drops from 29,750 to 21,006 due to density changes and increased molecular viscosity at elevated temperatures. The stability and convergence also improve by simplifying the geometry even more so that fewer disturbances occur while keeping the average Reynolds number the same. This can be seen in the stream pattern of geometry D in Fig. 6.24 where the secondary air flows in from the entire diameter around the burner, which removes the recirculation zone caused by the annular air inlet (geometry B) and results in a pure Craya-Curtet flow. Geometry D does not suffer from convergence problems for the full range of AFR, but due to lack of recirculation (hence convection), the wall temperatures are relatively low and not representative of geometry A (Fig. 6.25).

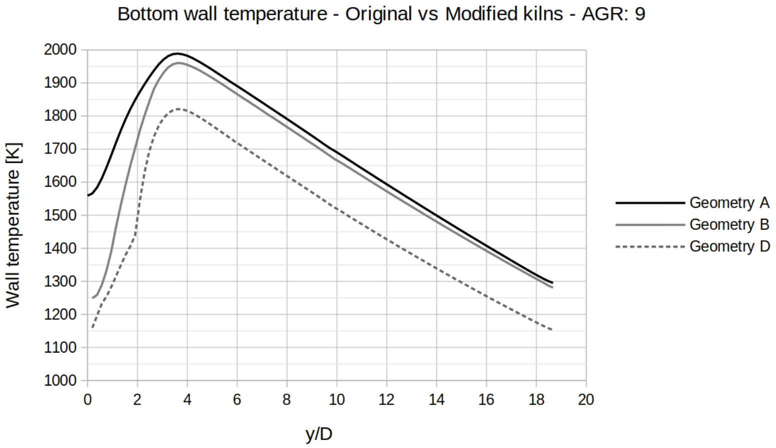

Figure 6.25: Computed bottom wall temperature along the axis of the kiln with geometries A, B, and C. Air-fuel ratio: 9.

VORTEX STRETCHING

The question now is what causes the flow to be unstable in the geometry with the annular air inlet? We believe the answer lies in when we plot the streamlines. In ParaView we plot the streamlines through a line, such that the stream pattern will look like a sheet with several three-dimensional rotations. However, we always get the complete three-dimensional concentric recirculation zone originating from that line. Apparently the streamlines propagate in the radial direction ('out of the plane') around the burner.

This gets even clearer when we plot the streamlines through a line at the center of the vortex. In Fig. 6.26 we see how the streamlines propagate in the radial direction

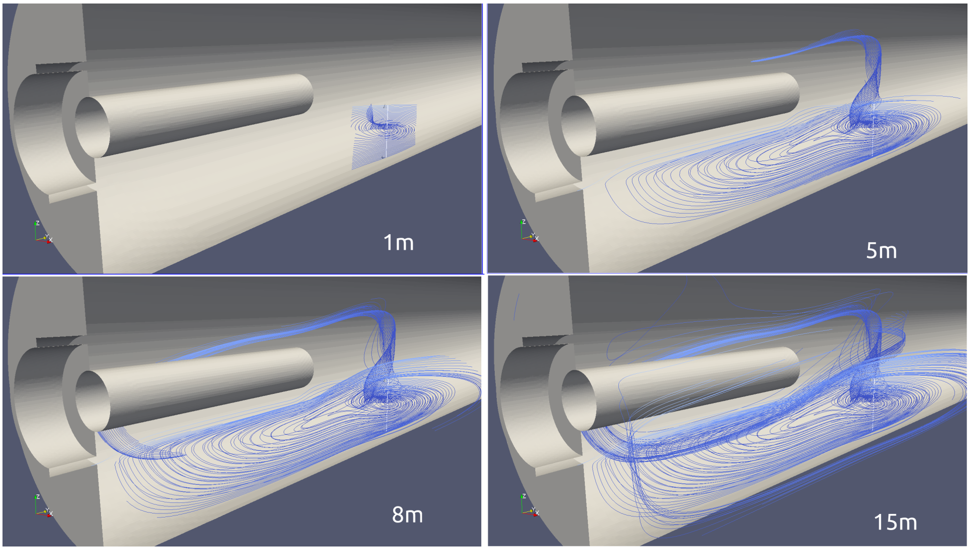

Figure 6.26: development of the vortex stretching with increasing streamline length.

in a swirling motion when we increase the maximum length of the streamlines. This phenomenon is known as vortex stretching, which is the key mechanism of the turbulent energy cascade and an essential aspect of turbulence [126]. We believe that the velocity gradients generated by the vortex stretching causes the instability in the axisymmetric geometry. That's why turbulence models that account for mean vorticity better predict recirculation zones [125].

To explain why this problem does not occur in two-dimensional simulations we look at the evolution of vorticity that is obtained by taking the curl of the Navier-Stokes equations [123], which for the case of incompressible flow leads to

$$\frac{D(\bar{\omega})}{Dt} = (\bar{\omega} \cdot \nabla)\bar{u} + \nu \nabla^2 \bar{\omega}, \tag{6.1}$$

where ω is the vorticity, which is the curl of the velocity, ∇ x $\bar{u} = \bar{\omega}$. Equation 6.1 is known as the vorticity equation, where angular momentum is conserved. The first term on the right-hand side of equation 6.1 represents vortex stretching and vortex tilting. Mathematically it can be derived that this term vanishes for two-dimensional flow (see e.g. [127], section 1.3) and explains why vortex stretching is purely a three-dimensional phenomenon.

What we learn from this is that even for symmetric geometries with symmetric boundary conditions may still lead to asymmetric flows with steady RANS simulations. Small asymmetries originating from mesh discretization, numerical schemes, or boundary condition implementation may be amplified. This is also observed in [128] and therein referenced literature.

6.4.2. TWO-DIMENSIONAL AXISYMMETRIC SIMULATIONS

We have seen how vortex stretching leads to statistically unstable flow in three dimensions. This instability does not occur in 2D axisymmetric simulations. Therefore, in order to continue investigating the effect of increasing the AFR to higher values, we choose to conduct the simulations of this analysis in 2D and are aware of neglecting the effect of vortex stretching. For this, we have to modify Geometry B (with the annular-shaped secondary air inlet) by making the fuel inlet axisymmetric as well. This results in Geometry C, where the fuel inlet is also annular-shaped (see Fig. 6.27). The fuel inlet thickness is determined such that the high fuel inlet Mach number remains roughly the same as in geometries A and B (Tab. 6.2). Geometry C is the exact 3D version of its 2D axisymmetric counterpart. The 2D mesh consists of 92,634 control volumes of which the solution is grid independent. The dimensions of the kiln and the wall are shown in Fig. 6.28.

Table 6.2: Fuel inlet mach numbers.

	Geometries A and B	Geometry C, 2D and 3D
Fuel inlet Mach number	0.52	0.58

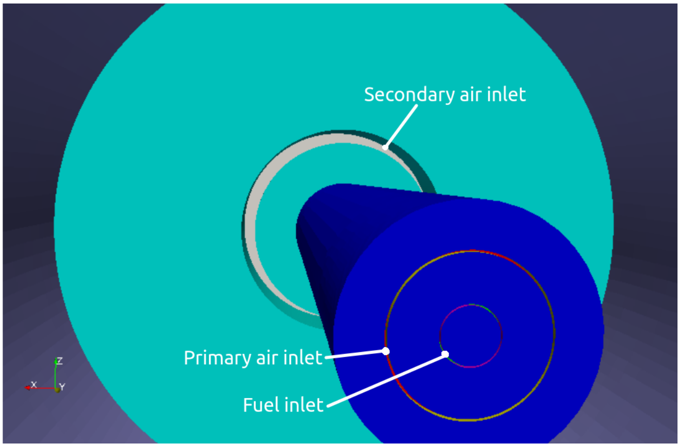

Figure 6.27: Modified kiln geometry C where all three inlets are annular shaped

6.4.3. BOUNDARY CONDITIONS

For the sensitivity analysis of varying AFR, we keep the mass flow rates of the fuel inlet and primary air inlet constant at 0.1 kg/s and 0.15 kg/s, respectively. The AFR

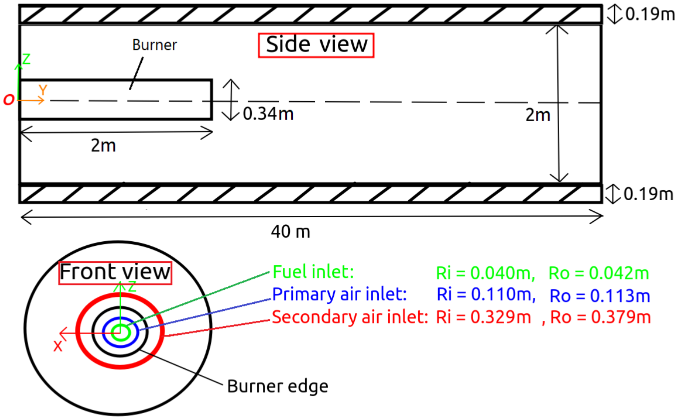

Figure 6.28: Dimensions (not to scale) of the modified kiln geometry C where all three inlets are annular shaped.

is increased from 9 to 14 by raising the secondary air inlet flow, and the pressure at the outlet is fixed at 1 atm. Further boundary conditions are shown in Table 6.3. The wall properties are given in Table 6.4. Combustion is solved with the Eddy Dissipation Model (EDM) and thermal radiation with the P1 approximation. Also, the external heat loss to the environment $q_{r,ext}$ is applied (Sec. 2.6.5), where the outer wall emissivity is equal to 0.9 (oxidised steel). Thermal NO formation is evaluated with the postprocessor as described in Sec. 2.4.

Table 6.3: Boundary conditions for the kiln model C with varying AFR.

Variable	Fuel inlet	Primary air	Secondary air	Gas-wall interface	Outer wall surface
T[K]	293	293	773	Coupled	$q_{r,ext}$
Y_{CH_4} [-]	1	0	0	zG*	-
Y_{O_2} [-]	0	0.23	0.23	zG	-
Y_{N_2} [-]	0	0.77	0.77	zG	-
$\dot{m}$ [kg/s]	0.1	0.15	1.6-2.5	-	-

*zG stands for the Neumann boundary condition zeroGradient.

Table 6.4: Thermal properties of the refractory wall.

Density ρ	Thermal conductivity λ_s	specific heat capacity c_p	emissivity ϵ_s
$2800\,\mathrm{kgm^{-3}}$	$2.1\,\mathrm{Wm^{-1}K^{-1}}$	$860\,\mathrm{Jkg^{-1}K^{-1}}$	0.6

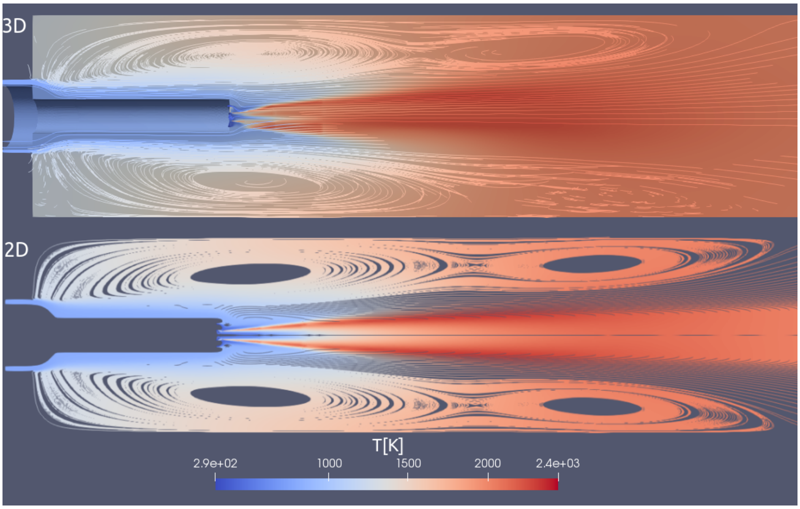

Figure 6.29: Streamline and temperature contour plot on the vertical longitudinal cross-section with the 3D and 2D simulations.

6.4.4. EFFECT ON WALL TEMPERATURE

To verify the results in 2D, Fig. 6.29 shows the streamline plots on the vertical longitudinal cross-section, coloured by the temperature contour, and is compared with the 3D simulation. The recirculation zone surrounding the flame consists of two annular vortices that are connected to each other, where the most upstream vortex arises like in any confined jet flow and the downstream vortex is generated by the Craya-Curtet flow condition. The 2D flow is statistically stable with a fully symmetric recirculation zone, whereas with the 3D simulation we see light effects of the vortex stretching causing non-symmetry in the recirculation zone and flame. In Fig. 6.30 we see that the fuel flow is more diffuse in the 3D simulation, leading to a slightly wider and shorter flame as compared with the 2D simulation.

Although the flow stability is a major difference between the simulations in 2D and 3D, when we look at the wall temperature (see Fig. 6.31) we see that the difference is very small. The temperature profile for the 2D is slightly shifted downstream by 2.5% of the kiln's length.

This difference in wall temperature is acceptable and allows us to continue investigating the effect of changing AFR on the wall temperature, which can be seen in Fig. 6.32. As the AFR continues to increase, the wall temperature throughout the kiln drops. This makes sense. Surprisingly, the wall temperature peaks at an AFR of 9 and not at the stoichiometric AFR of 9.5. We know that the wall is heated predominantly by the flame via thermal radiation, which is related to the flame temperature.

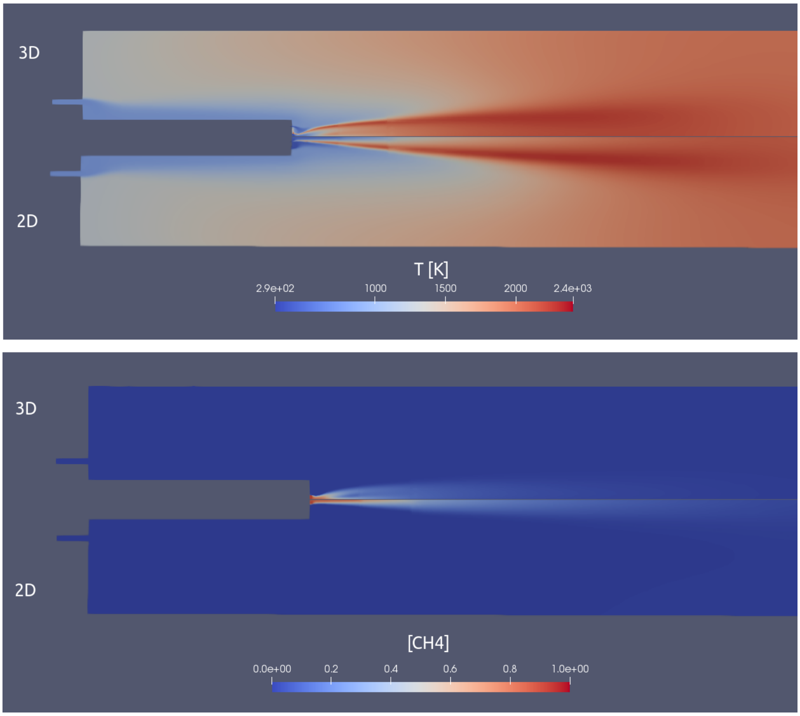

Figure 6.30: Contour plots of temperature and methane concentration on the vertical longitudinal cross-section with the 3D and 2D simulations.

Fig. 6.33 shows that the maximum flame temperature indeed peaks at the fuel-rich side of stoichiometric AFR. Theoretically the adiabatic flame temperature peaks at stoichiometry. For diffusion flames (like in our case), this even occurs at all AFR, as reactions always take place at stoichiometric conditions regardless of the global AFR [42]. However, in reality heat is lost due to dissociation of the products. Since the extent of dissociation is greater on the lean side, the peaking occurs on the rich side. Moreover, the reduced momentum of the co-flow at lower-than-stoichiometric AFR may also contribute to the increased temperature due to less heat dissipation by the secondary inlet stream.

6.4.5. EFFECT ON THERMAL NO FORMATION

Fig.6.34 shows the average concentration of thermal NO at the outlet as a function of AFR. Contrary to the maximum flame temperature, thermal NO formation relatively peaks on the lean side of stoichiometry due to competition of other species that react with oxygen as well. Fig. 6.35 helps to understand the complexity of thermal NO formation. Not only does it depend on the gas temperature and oxygen

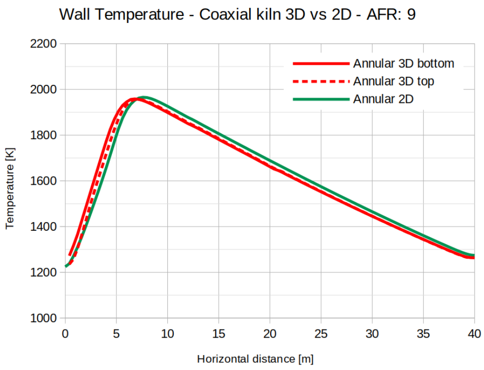

Figure 6.31: Computed top and bottom wall temperature along the axis of the kiln, 2D (green) vs 3D (red). AFR: 9

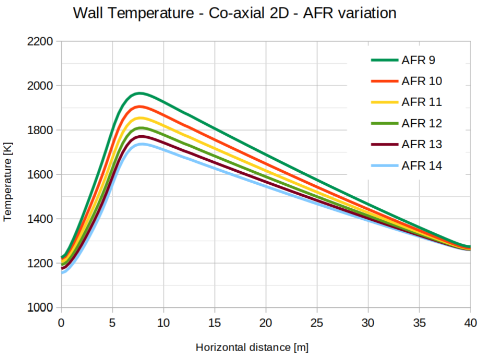

Figure 6.32: Computed wall temperature in the longitudinal direction of the kiln for different AFRs.

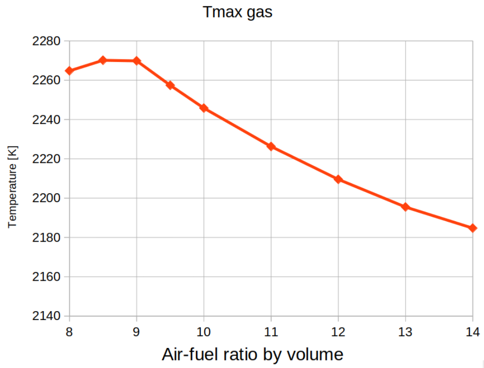

Figure 6.33: Computed maximum flame temperature for different AFRs.

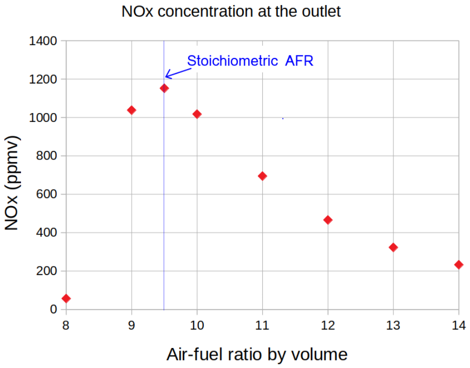

Figure 6.34: Average thermal NO concentration at the outlet of the kiln for different AFRs.

concentration, but on the flow as well, as was proven in [116]. An example in Fig. 6.35 is that the temperature profiles of AFRs 8 and 10 are similar, but the oxygen concentration is less in the hot region of the gas in the case of AFR of 8. Secondly, the thermal NO contour plot shows that the NO concentration peaks at an AFR of 9, but when we look at the outlet (Fig.6.34) more thermal NO is emitted at an AFR of 9.5. The streamlines show that more NO is trapped in the vortex for the AFR of 9, which explains why less NO is emitted at the outlet.

6.4.6. PRELIMINARY NO$_X$ VALIDATION

As a first attempt to validate the varying AFR simulations, the NO concentration at the outlet of geometry C is compared to measurements of the actually operating

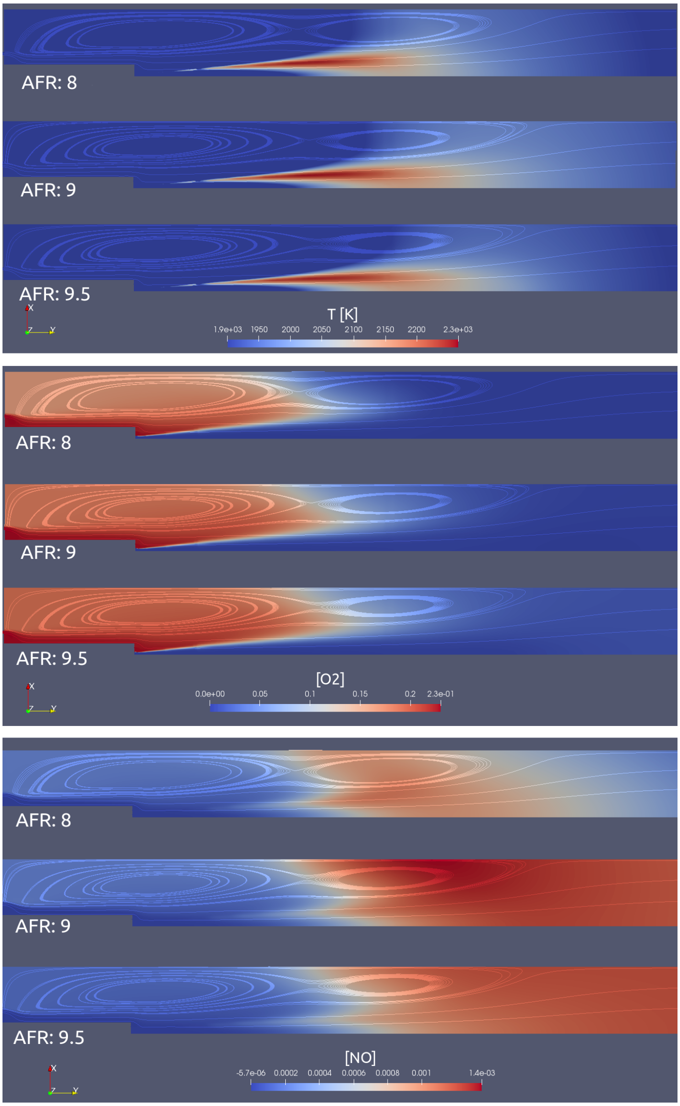

Figure 6.35: Contour plot of the temperature, and the mass fractions of oxygen and thermal NO, respectively for three different air-fuel ratios, from fuel-rich mixture (AFR 8) to stoichiometric mixture (AFR 9.5). The minimum temperature is deliberately set at 1900 K to highlight the critical areas where thermal NO can potentially be formed.

kiln with geometry A. The box plot of the measurements is shown in Fig. 6.36, where the simulation data are added. Recall that the simulation model is simplified, which makes it not a really fair comparison. Not only are geometries A and C different, but also the way the fuel and air mix. Furthermore, the model is without rotation, inclination, and material bed flow. On the other hand, the material bed has a cooling effect on the refractory wall, and therefore the measurements can only be conducted during operation, i.e. with flowing raw material. Also, the detector does not distinguish between thermal NO and other sources of NO_x, while the model predicts thermal NO exclusively. Lastly, since the actually operating kiln is burning natural gas which contains nitrogen, fuel NO_x may form with excess air, as what occurs with coal combustion shown in [121] and [122].

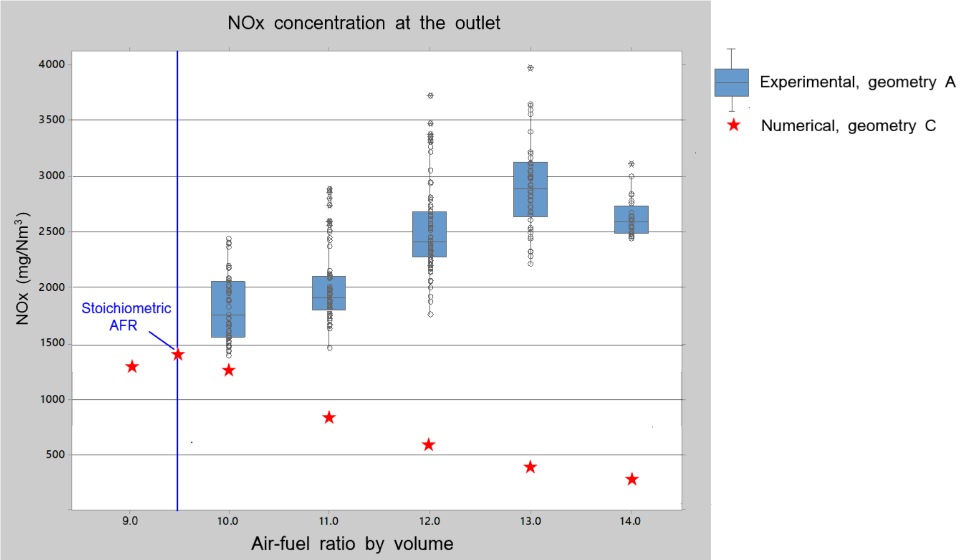

Figure 6.36: Average thermal NO concentration at the outlet of the kiln for different AFRs.

The main difference between the results is that the measured data show a peak well beyond stoichiometry, around AFR 13, and maybe less when considering the uncertainties. Although theory suggests that the maximum should lie near the stoichiometric AFR at the lean side, studies have shown that for diffusion flames it may increase with leaner mixtures [119] and it has been reported that the peak can reach air-fuel equivalence ratios of 1.3 to 1.4 (~ AFR 13) [129][130].

The reason why the predicted NO peak remains tied to stoichiometric mean conditions is because the NO post-processor does not include a turbulence-chemistry interaction model, and therefore the fluctuating temperature and species concentrations are not captured (Sec. 2.4). In turbulent flames operating under globally lean conditions, the absent presumed *Probability Density Function* of temperature and species concentrations would allow for instantaneous stoichiometric conditions to occur locally, even at lean conditions, which can significantly enhance NO formation.

However, we are aware that many factors play a role that leads to this behaviour that we see in the measurements. One thing is certain, which is that the simulation model needs to be made more complex in order to get closer to understand where the large shift of the peak comes from.

6.5. SOLUTION PROPOSAL 3: EXHAUST GAS RECIRCULATION

It was observed in Sec. 6.4 that increasing AFR may not necessarily lead to thermal NO reduction, as the 2D model has shown. One of the potential reasons is that local hot spots are still present in the complex 3D flow (Figs. 6.16 and 6.18), which are provided with more air at higher AFR, which is a source for thermal NO.

This section proposes to apply Exhaust Gas Recirculation (EGR) as the third solution strategy, which is an NO_x reduction combustion technique. It is also known as Flue Gas Recirculation, and it is a well-established technique applied in internal combustion engines as well as high-temperature industrial processes [30, 131, 132, 133, 134]. Like most other NO_x reduction techniques, the goal is to reduce thermal NO emissions by lowering the flame's temperature.

When recycling a portion of the exhaust gases to mix with the secondary air inlet, the oxygen concentration is lowered by the inert combustion products, such as CO_2 and H_2O. This leads to a less sharp temperature rise at the flamefront and therefore a lower peak temperature. The combination of low temperature and low oxygen concentration prevents thermal NO formation.

6.5.1. BOUNDARY CONDITIONS

The simulations are carried out in the same 2D geometry as in Sec. 6.4. As a parameter, the EGR rate is defined as:

$$\mathrm{EGR} = \frac{\dot{m}_{\mathrm{exh}}}{\dot{m}_{\mathrm{tot}}} \mathrm{x}100\%, \tag{6.2}$$

where $\dot{m}_{\mathrm{exh}}$ and $\dot{m}_{\mathrm{tot}}$ are respectively, the recirculated exhaust mass flow rate and the total mass flow rates of the secondary air inlet. For the sensitivity analysis of varying EGR rates, we keep the mass flow rates of the fuel inlet, primary inlet, and secondary inlet constant at 0.1 kg/s, 0.15 kg/s, and 2.49 kg/s, respectively. This combination has a volumetric AFR of 14. The reason for this basis is that we start with excess air and slowly dilute the oxygen with exhaust gases until the mixture of oxygen and fuel is nearly stoichiometric, which coincides with an EGR rate of 7.1%. Table 6.5 shows that the EGR rate is gradually increased by lowering the O_2 concentration and raising the N_2, CO_2 and H_2O concentrations. In reality, the exhaust mixture changes with the changing inlet mixture, which makes it a coupled process. To simplify this, it is assumed that the recirculated exhaust mixture is fixed. The chosen mixture for the recirculated exhaust is that of combustion products resulting from stoichiometric combustion.

Table 6.5: Boundary conditions for the kiln model C with varying EGR. Species are denoted in mass fractions.

Secondary air inlet variable	EGR 0%	EGR 2.3%	EGR 5.0%	EGR 7.1%
$\dot{m}$[kg/s]	2.49	2.49	2.49	2.49
T[K]	773	773	773	773
Y_{O_2}	0.23	0.2070	0.1800	0.1590
Y_{N_2}	0.77	0.7872	0.8075	0.8230
Y_{CO_2}	0	0.0032	0.0069	0.0100
Y_{H_2O}	0	0.0026	0.0056	0.0080

6.5.2. EFFECT ON WALL TEMPERATURE AND NO FORMATION

It can be observed in Figs. 6.37 and 6.38 that EGR successfully lowers the flame temperature, resulting in a reduction of 90% in thermal NO formation as compared with the base AFR of 14, and a reduction of 99.8% as compared with the original operating condition at AFR of 9.

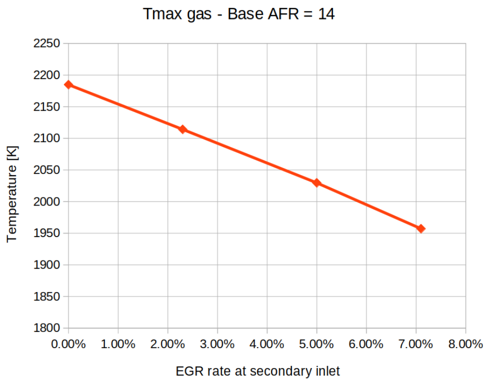

Figure 6.37: Computed maximum gas temperature for different EGR rates.

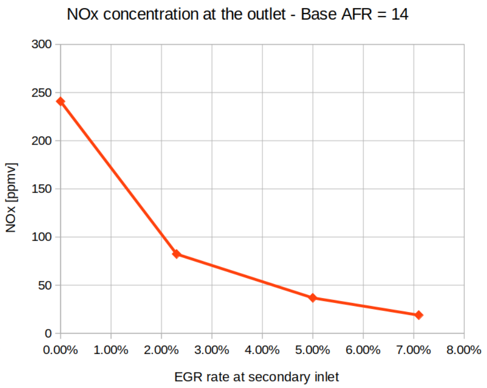

Figure 6.38: Average thermal NO concentration at the outlet of the kiln for different EGR rates.

In the previous section, it was observed that increasing the AFR from 9 to 14 results in a wall temperature drop of nearly 250 K (Fig. 6.32). This drop is too severe for the sintering process of the raw mixture. Although the gas temperature drops even further with increasing EGR, the wall temperature actually increases up to 100 K, which is a reduction of 150 K as compared with an AFR of 9. The reason why the wall temperature increases with increasing EGR is that the emissivity of the inert gases CO_2 and H_2O are much higher than that of air, and increasing their concentrations with EGR increases the radiative heat transfer, as is shown in Fig. 6.40.

There is now a playing field where the optimal wall temperature can be found for the sintering process, which is already too high at an AFR of 9. EGR can therefore help tackle both thermal NO emissions as well as prevent ring formation of the raw mixture. However, there are also challenges. Too much EGR can lower the oxygen available for combustion, potentially leading to incomplete combustion and higher CO emissions. Also, the system needs to be redesigned, which may require modifications like EGR coolers, heaters, control valves, and additional temperature monitoring systems to regulate gas flow and maintain optimal performance.

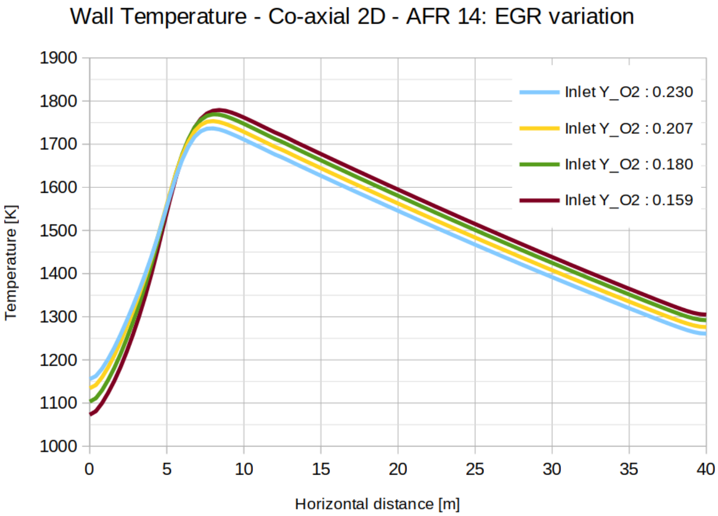

Figure 6.39: Computed wall temperature in the longitudinal direction of the kiln, starting from the AFR of 14 and reducing inlet oxygen mass fractions due to increasing EGR.

6.5.3. LIMITATIONS

As is shown in Sec. 6.4, the promising results in this oversimplified model may not be expected in reality. An attempt was made to include detailed chemistry in the model and conduct a transient simulation using the Local Time Stepping approach, starting from a steady-state solution. However, this fails as the flame blows off, a problem that does not occur with a mixed-is-burned combustion model. It is

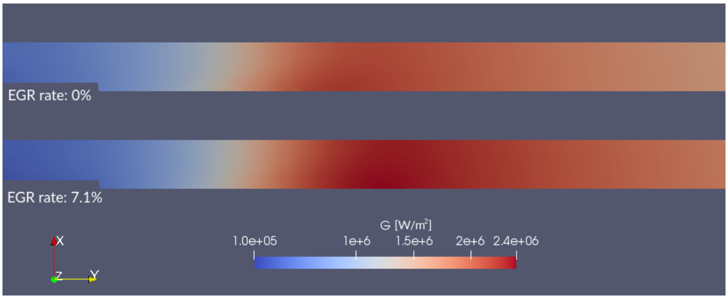

Figure 6.40: Comparison of the incident radiation field between the EGR rate of 0% and the maximum EGR rate of 7.1%.

found that the annular fuel jet does not mix well with its surroundings. In the 3D geometry with 16 circular fuel inlets, the jets mix well, and no lift-off is observed. However, the simulations still fail with LTS. Full transient URANS simulations were not conducted as the computational resources were limited during the time of research. Moreover, a transient simulation requires that thermal NO formation be resolved instead of calculated in the post-processing stage.

6.6. CONCLUSIONS

In this chapter, we applied the developed solver `multiRegionReactingFoam` to a full-scale 3D rotary kiln, where three potential strategies were investigated to mitigate the high thermal NO emissions: by altering the air inlet geometry, by modifying the AFR, and finally by applying EGR (in 2D).

The simulations of the original situation (Geometry A and AFR of 9) demonstrated the complexity of modelling the 3D flow structure. The 3D simulation gave insights into how to mitigate thermal NO formation, which is strongly influenced by the flow. Changing the secondary air inlet from a rectangle into an annulus around the burner root creates a stable and favourable flow condition with significantly lower thermal NO formation.

Reaching a steady-state became a challenge when studying the effect of increasing AFR. Three-dimensional simulations have shown that we are limited in carrying out steady-state simulations at low AFRs only. Even when the geometry and boundary conditions are symmetric, convergence to a steady-state solution becomes unfeasible at higher AFRs, due to the limitation of steady RANS modelling at high Reynolds numbers in the presence of strong swirl and recirculation. One of the contributing factors is the inherently three-dimensional nature of the mean flow, in which mechanisms such as vortex stretching play an important role in shaping the turbulence dynamics. While vortex stretching is a fundamental mechanism of three-dimensional turbulence, in RANS its effects are only indirectly reflected through the modeled turbulence quantities associated with strong mean velocity gradients. As a result, symmetric geometries with symmetric boundary conditions may still lead to asymmetric solutions in steady RANS simulations due to the nonlinear nature of the governing equations and their sensitivity to small numerical perturbations. Two-dimensional simulations have been shown to converge, as the phenomenon of vortex stretching vanishes, and high AFRs no longer lead to convergence problems. The impact of this simplification is shown to be acceptable for the thermal behaviour.

It is shown that both the wall temperature and thermal NO emissions peak at the fuel-rich and fuel-lean sides of the stoichiometric AFR, respectively. If the AFR continues to increase, the wall temperature decreases significantly and thermal NO emissions drop dramatically. The results demonstrated that increasing AFR reduces thermal NO formation but also leads to a significant drop in wall temperature, which may negatively impact the sintering process.

In an attempt to validate the AFR study, the results were compared with NO_x measurements in the fully operating kiln. The main difference between the results is that the measured data show a peak well beyond stoichiometry, around AFR 13, and it has been reported in literature that the peak can reach air-fuel equivalence ratios of 1.3 to 1.4 (~ AFR 13). The reason why the predicted NO peak remains tied to stoichiometric mean conditions is because the NO post-processor does not include a turbulence-chemistry interaction model, and therefore the fluctuating temperature and species concentrations are not captured. However, we are aware that many factors play a role that leads to this behaviour that we see in the measurements.

The measurements showed unexpected results as the measured NO_x formation peaks at significantly fuel-lean conditions. However, the conditions are too different to consider it a valid comparison.

EGR proved to be highly effective in reducing thermal NO emissions – achieving reductions of up to 99.8% - while simultaneously maintaining a more favourable temperature profile along the kiln wall as compared with the AFR study, despite the low flame temperature. This is due to the higher emissivity of the abundant combustion products.

Although the results with EGR are promising, it needs to be addressed that the simulations fall short in that the study did not account for detailed chemistry, 3D geometry, or the presence of the material bed. The inclusion of detailed chemistry was attempted in 3D, with the Local Time Stepping method to save time, but convergence issues prevented its full implementation, while in 2D, flame blow-off was observed due to lack of mixing.

III

FINAL REMARKS

7

CONCLUSIONS AND RECOMMENDATIONS

The main objective of this thesis was to contribute to the development and application of a general multiphysics CFD solver for rotary kiln applications. This objective has been achieved by implementing and validating a solver capable of modelling turbulent flow, combustion, (wall) heat transfer, and thermal nitric oxide (NO) formation within the OpenFOAM framework. The developed solver enables consistent investigation of the coupled physical processes governing rotary kiln operation and has been successfully applied to industrial-scale test cases. The strengths and limitations of the solver and the employed models are discussed in the remainder of this chapter.

7.1. MODEL FORMULATION

This dissertation established a comprehensive mathematical framework for simulating the complex interactions within rotary kilns, focusing on turbulent combustion, heat transfer, and thermal NO formation. For these applications to be successful, a good understanding of the numerical implementation is a prerequisite. Hence, to form a 'bridge' between physics and numerical analysis, various numerical and solution methods in OpenFOAM were highlighted and discussed.

The theoretical work offered insights into the advantages and limitations of different modelling approaches and ended by presenting the coupled methodology to model turbulent combustion and CHT, establishing a foundation for the solver's application in practical industrial cases.

7.2. NUMERICAL AND SOLUTION METHODS

Special attention was given to the impact of linear solvers and tolerances on the simulation accuracy and efficiency. These were the main conclusions:

- For non-reacting flow cases, the pressure-corrector equation was the major bottleneck that delayed the solution from converging quickly. Several of the more advanced standard linear solvers were tested with a 20-core high-performance computer on a 3D full-scale kiln, consisting of 2.3 million cells. The most efficient and stable solution method turned out to be the Conjugate Gradient (CG) solver, combined with the Generalised Geometric-Algebraic MultiGrid (GAMG) as a preconditioner. A speed-up factor of 7 was reached as compared with the slowest of the advanced methods. Even higher speed-ups were reached compared with the basic linear solvers of OpenFOAM in most of the standard tutorials. Also, the stability is enhanced when the GAMG is used as preconditioner for the Bi-CG Stabilized (Bi-CGSTAB) method instead of the basic preconditioners.

- In the case of reacting flows, solving the thermochemistry was the bottleneck. When enabling the simple Eddy Dissipation Model (EDM) for the studied kiln, the speed-up of the CG method with GAMG preconditioner for the pressure-corrector equation was only 4% as compared with the Diagonal Incomplete Cholesky preconditioner. When deploying a detailed reaction mechanism, which was tested on the Sandia flame D benchmark, the speed-up of the linear solvers completely diminished.

- To speed up simulations with a detailed reaction mechanism, the simulation time could have been significantly reduced with a simplified reaction mechanism, but the NO emission predictions were greatly affected. The Tabulation of Dynamic Adaptive Chemistry (TDAC) method was far superior, with a speed-up factor of about 160 for the GRI-3.0 mechanism and small differences in NO predictions. However, the speed-up of TDAC greatly reduces with the size of the reaction mechanism. Local time stepping (LTS) in this test was ’only’ 40% faster due to the small residence time of the domain where the transient simulation reached a steady-state solution relatively quickly. Therefore, using TDAC along with LTS is the fastest option in the case of activated detailed chemistry, with minimum impact on the emission predictions.

7.3. CODE DEVELOPMENT AND VALIDATION

The developed solver `chtMultiRegionFoam` was verified and validated using the well-documented Sandia flame D configuration (Chapter 4):

- This work has shown that OpenFOAM's standard solvers, `reactingFoam` and `chtMultiRegionFoam` are succesfully coupled in the new solver `multiRegionReactingFoam`. This enables the modelling of combustion with conjugate heat transfer. The results of the new solver, with conjugate heat transfer turned off, are identical to `reactingFoam`, which means that the quality of the combustion simulation does not depend on the solver itself.

- The quality of the CHT for reacting cases was benchmarked against ANSYS Fluent, which resulted in good qualitative agreements with convection alone, and quantitative agreements when also radiation is involved.

7.4. APPLICATION TO KILN SIMULATIONS

The solver was then tested on a rotary kiln benchmark to study the significance of including the energy transport in the refractory wall and the external heat loss to the environment (Chapter 5):

- Compared with the reference kiln model, where a constant temperature was assigned to the wall, a temperature increase of 10–90% was predicted throughout the central axis of the kiln. Our results showed a bell-shaped temperature distribution at the inner wall surface, with more than a 200% increase of the peak temperature and a 60% difference between the minimum and maximum. Due to the increased thermal insulation, the maximum flame temperature increased by 12%, and the maximum radiative heat flux to the wall decreased by 80%. The inclusion of external radiative heat loss caused the outer wall surface temperature to increase by 30%, whereas the effects on the inside of the kiln were much smaller. This work confirms the important role of thermal boundary conditions and has introduced the CHT approach to represent it accurately.

Finally, the fully integrated model was applied to a full-scale 3D rotary kiln with a focus on reducing thermal NO emissions with three potential strategies (Chapter 6):

- The simulations of the original situation (Geometry A and volumetric air-to-fuel ratio of 9) demonstrated the complexity of modelling the 3D flow structure. The 3D simulation provided insights into how to mitigate thermal NO formation, which was strongly influenced by the flow. Changing the secondary air inlet from a rectangle into an annulus around the burner root created a stable and favourable flow condition with reduced hot spots and significantly lower thermal NO formation.

- Reaching a steady-state became a challenge when studying the effect of increasing AFR. Three-dimensional simulations have shown that we are limited in carrying out steady-state simulations at low AFRs only. Even when the geometry and boundary conditions are symmetric, convergence to a steady-state solution becomes unfeasible at higher AFRs, due to the limitation of steady RANS modelling at high Reynolds numbers in the presence of strong swirl and recirculation. One of the contributing factors is the inherently three-dimensional nature of the mean flow, in which mechanisms such as vortex stretching play an important role in shaping the turbulence dynamics. While vortex stretching is a fundamental mechanism of three-dimensional turbulence, in RANS its effects are only indirectly reflected through the modeled turbulence quantities associated with strong mean velocity gradients. As a result, symmetric geometries with symmetric boundary conditions may still lead to asymmetric solutions in steady RANS simulations due to the nonlinear nature of the governing equations and their sensitivity to small numerical perturbations. Two-dimensional simulations have been shown to converge, as the phenomenon of vortex stretching vanishes, and high AFRs no longer lead to convergence problems. The impact of this simplification is shown to be acceptable for the thermal behaviour.

- Continuing with 2D simulations, it was shown that both the wall temperature and thermal NO emissions peaked at the fuel-rich and fuel-lean sides of the stoichiometric AFR, respectively. If the AFR continued to increase, the wall temperature decreased significantly and thermal NO emissions dropped dramatically. The results demonstrated that increasing AFR reduced thermal NO formation but also led to a significant drop in wall temperature, which could negatively impact the sintering process of the material bed.

- In an attempt to validate the AFR study, the results were compared with NO_x measurements in the fully operating kiln. The main difference between the results is that the measured data show a peak well beyond stoichiometry, around AFR 13, and it has been reported in literature that the peak can reach air-fuel equivalence ratios of 1.3 to 1.4 (~ AFR 13). The reason why the predicted NO peak remains tied to stoichiometric mean conditions is because the NO post-processor does not include a turbulence-chemistry interaction model, and therefore the fluctuating temperature and species concentrations are not captured. However, we are aware that many factors play a role that leads to this behaviour that we see in the measurements.

- Exhaust Gas Recirculation (EGR) proved to be highly effective in reducing thermal NO emissions – achieving reductions of up to 99.8% – while simultaneously maintaining a more favourable temperature profile along the kiln wall as compared with the AFR study, despite the low flame temperature. This was due to the higher emissivity of the abundant combustion products. Hence, EGR could potentially reduce thermal NO formation without compromising the kiln's productivity.

- The parametric studies with AFR and EGR were only carried out with a 2D geometry in steady-state condition. An attempt was made to include detailed chemistry in the solver and conduct a semi-transient simulation using the Local Time Stepping approach, starting from a steady-state solution. However, this fails as the flame blows off, a problem that does not occur with a mixed-is-burned combustion model. It is found that the annular fuel jet (in 2D simulation) does not mix well with its surroundings. In the 3D geometry with 16 circular fuel inlets, the jets mix well, and no lift-off is observed. However, the simulations still fail with LTS. Full transient URANS simulations were not conducted as the computational resources were limited during the time of research.

7.5. FUTURE WORK

Although the solver `multiRegionReactingFoam` has shown to be able to successfully simulate the combustion process and wall heat transfer in a 3D full-scale kiln, further progress can be made with the simulation methods and applied mathematical models. The big challenge was to reach a steady-state solution with higher airflow rates than the nominal condition, which led to critically high Reynolds numbers; too high to be dampened by RANS. In this view, the following recommendations originate from the current work:

- To capture the varying effects of the vortices, it is recommended to carry out transient simulations. However, it will lead to much longer run times, even with URANS. This is mainly due to the sheer size of the kiln, with a flow residence time of about 30 seconds, while the CFL condition (<0.2) requires that the time steps should be of the order of 10^{-8} seconds. The latter is due to the limitation by the fuel inlets, where the inlet speed reaches Mach 0.6 and the cell sizes are a fraction of a millimeter. However, during the time of the PhD research, the computational resource was limited to a 20-core machine. Currently the university has the DelftBlue supercomputer at its disposal, with about 17 thousand compute cores in total [135]. Successful attempts were made in [136] to conduct a transient simulation in reasonable time, where they were able to increase the time-step size to 10^{-5} seconds, be it with a global reaction mechanism.

- Note that in a transient simulation, as the flow, temperatures, and emissions fluctuate irregularly, the NO post-processor will produce different results at different moments that the transient simulation is frozen. There are several ways to overcome this. First option is to conduct a follow-up simulation, starting from a nearly converged simulation, and to employ the thermal NO post-processor as a function object, in order to calculate the NO formation after each time-step. The simulation will run much slower than without the post-processor, but average NO values can be determined after a few fluctuations. Alternatively, NO formation can be resolved directly with a detailed reaction mechanism, such as the GRI-3.0 or GRI-2.11, but will lead to very long run-times. Recently however, Dynamic Load Balancing (DLB) was introduced in [137] as a promising technique to accelerate chemistry in parallel simulations. Along with a fully analytical chemistry Jacobian formulation and optimised ODE solution routines, a speed-up of up to x250 can be achieved, as compared to the standard OpenFOAM chemistry model. DLB-Foam can easily be compiled in the OpenFOAM environment and linked to OpenFOAM's combustion solvers [138].

- The main limitation of the NO post-processor is that it does not include a turbulence-chemistry interaction model, and therefore the fluctuating temperature and species concentrations are not captured, which are the major source of NO formation. Room for improvement is to include the effects of temporal fluctuations using a presumed Probability Density Function approach.

- It is worthwhile to experiment with the Flamelet Generated Manifold (FGM) as combustion model (Sec.2.5.3), which is superior to the EDM as it includes detailed chemistry, and the thermal NO post-processor will not be necessary. Compared with the PaSR, the stiff ODEs are avoided and (much) fewer transport equations are resolved, which provides for better numerical robustness and computational efficiency.

- Some characteristics of the rotary kiln process that have been excluded in this dissertation are the kiln's rotation and both the granular motion and heat transfer of the material bed. These were out of the scope of this dissertation due to the complexity of building such a solver during the time of OpenFOAM v5. As of 2023, OpenFOAM v11 was released, which transitioned from the tens of different solvers to a modular approach [139], addressing the fact that developers had to build a new solver for each new application. The modular solvers in v11 are simpler and more flexible, which makes it easier to extend, e.g., basic fluid flow with CHT, chemistry, Multiple Reference Frames (MRF) and Lagrangian particles, hence capturing the complete process inside the rotary kiln.

- In case the structural durability of the wall is not a focus point for rotary kilns, the solid region can be left out. As the cylindrical kiln walls consist of uniform layers of refractory material and steel, their thermal resistance can be incorporated with the `externalWallHeatFluxTemperature` function where the thicknesses and conductivities of the layers can be included. This speeds up the convergence as well.

BIBLIOGRAPHY

[1] A.A. Boateng. *Rotary kilns: transport phenomena and transport processes.* 2nd. Oxford: Elsevier and Butterworth-Heinemann, 2016.

[2] N.B. Winter. *Basic principle of a wet-process kiln.* https://www.understanding-cement.com/kiln.html. Copyright 2005-2021 by WHD Microanalysis Consultants Ltd. Reprinted with permission.

[3] *Climate change: The massive CO2 emitter you may not know about.* https://www.bbc.com/news/science-environment-46455844. accessed: 2018-12-30.

[4] A.H. Lefebvre and D.R. Ballal. *Gas Turbine Combustion: Alternative Fuels and Emissions.* 3rd. Boca Raton: CRC Press., 2010.

[5] J.H. Seinfeld and S.N. Pandis. *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change.* 2nd. Hoboken, New Jersey: John Wiley and Sons, Inc., 2006.

[6] European Commission. *Zero Pollution: Modernised rules on industrial and livestock rearing emissions come into effect.* https://environment.ec.europa.eu/news/revised-industrial-emissions-directive-comes-effect-2024-08-02_en. accessed: 2024-09-15.

[7] A. Habibi, B. Merci, and G.J. Heynderickx. "Multiscale Modeling of Turbulent Combustion and NOx Emission in Steam Crackers". In: *AIChE Journal* 53.9 (2007), pp. 2384–2398. DOI: 10.1002/aic.11243.

[8] A. Ashouri and K. Mazaheri. "Effects of burner configuration modification on CH4-H2 fueled industrial steam cracking furnaces: Heat transfer, pollutant emission, and furnace performance". In: *Energy* 335 (2025), p. 138023. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2025.138023. URL: https://www.sciencedirect.com/science/article/pii/S0360544225036655.

[9] C. Cravero, D. Marsano, and G. Milanese. "A CFD Model for the Direct Coupling of the Combustion Process and Glass Melting Flow Simulation in Glass Furnaces". In: *Energies* 18.7 (2025). DOI: 10.3390/en18071792.

[10] IA Sofia Larsson, T Staffan Lundström, and B Daniel Marjavaara. “Calculation of kiln aerodynamics with two RANS turbulence models and by DDES”. In: *Flow, turbulence and combustion* 94.4 (2015), pp. 859–878.

[11] M. Ditaranto and J. Bakken. “Study of a full scale oxy-fuel cement rotary kiln”. In: *International Journal of Greenhouse Gas Control* 83 (2019), pp. 166–175. ISSN: 1750-5836.

[12] Morten Nedergaard Pedersen. “Co-firing of Alternative Fuels in Cement Kiln Burners”. PhD thesis. Technical University of Denmark, 2018.

[13] H.F. Elattar et al. “Study of Parameters Influencing Fluid Flow and Wall Hot Spots in Rotary Kilns using CFD.” In: *The Canadian Journal of Chemical Engineering* 94 (2016), pp. 355–367.

[14] B. Manickam et al. “Enriched Oxygen Combustion Simulation for Rotary Kiln Application”. In: *4th European Combustion Meeting, Vienna* (2009).

[15] Hassan F. Elattar et al. “Swirled Jet Flame Simulation and Flow Visualization Inside Rotary Kiln—CFD with PDF Approach”. In: *Processes* 8.2 (2020).

[16] W. Rong et al. “Numerical study of the solid flow behavior in a rotating drum based on a multiphase CFD model accounting for solid frictional viscosity and wall friction”. In: *Powder Technology* 361 (2020), pp. 87–98. ISSN: 0032-5910.

[17] B.-J .R. Mungyeko Bisulandu and F. Marias. “Modeling of the Thermochemical Conversion of Biomass in Cement Rotary Kiln”. In: *Waste and Biomass Valorization* 12 (2021), pp. 1005–1024.

[18] Peter J. Witt et al. “A hierarchical simulation methodology for rotary kilns including granular flow and heat transfer”. In: *Minerals Engineering* 119 (2018), pp. 244–262. ISSN: 0892-6875.

[19] Tobias Oschmann and Harald Kruggel-Emden. “A novel method for the calculation of particle heat conduction and resolved 3D wall heat transfer for the CFD/DEM approach”. In: *Powder Technology* 338 (2018), pp. 289–303. ISSN: 0032-5910.

[20] C. Pieper et al. “Interaction of the combustion of refuse derived fuel with the clinker bed in rotary cement kilns: A numerical study”. In: *Fuel* 266 (2020), p. 117048. ISSN: 0016-2361.

[21] K. Jang, W. Han, and K.Y. Huh. “Simulation of a Moving-Bed Reactor and a Fluidized-Bed Reactor by DPM and MPPIC in OpenFOAM”. In: (2019), pp. 419–435.

[22] M.B. Rysgaard. “CFD modelling of NOx formation and control in mineral processing.” PhD thesis. Technical University of Denmark, 2023.

[23] P. Dillon. “A Dual-Solver CFD Model for Conjugate Heat Transfer in Continuous Thermal Processing”. In: *Case Studies in Thermal Engineering* 49 (2023), p. 103337. DOI: `10.1016/j.csite.2023.103337`.

[24] S. Moghaddam et al. “Investigating the Potentials to Improve the Energy Consumption of a Lime Rotary Kiln Using Measurement and CFD Simulation”. In: *Heat and Mass Transfer* 59.5 (2023), pp. 949–970. DOI: `10.1007/s00231-022-03309-9`.

[25] Y. Shu et al. “Numerical Study on Oxy-Biomass Co-Firing in a Cement Rotary Kiln”. In: *Thermal Science* 28.5 (2024), pp. 4407–4419. DOI: `10.2298/TSCI2405407S`.

[26] M. C. Siame et al. “Dynamic Simulation of Heat Distribution and Losses in Cement Kilns for Sustainable Energy Consumption in Cement Production”. In: *Sustainability* 17.2 (2025). DOI: `10.3390/su17020553`.

[27] M. Pisaroni, R. Sadi, and D. Lahaye. “Counteracting ring formation in rotary kilns”. In: *Journal of Mathematics in Industry* 2 (2012), p. 3.

[28] H. Gao et al. “Reducing the recycle flue gas rate of an oxy-fuel utility power boiler”. In: *Fuel* 140 (2015), pp. 578–589.

[29] B. Danon et al. “Emission and efficiency comparison of different firing modes in a furnace with four HiTAC burners”. In: *Combustion Science and Technology* 183 (2011), pp. 686–703.

[30] D.A. Granados et al. “Effect of flue gas recirculation during oxy-fuel combustion in a rotary cement kiln”. In: *Energy* 64 (2014), pp. 615–625.

[31] B. Krause et al. “Coupled three dimensional DEM–CFD simulation of a lime shaft kiln—Calcination, particle movement and gas phase flow field”. In: *Chemical Engineering Science* 134 (2015), pp. 834–849.

[32] H.I. Kassem et al. “Implementation of the eddy dissipation model of turbulent non-premixed combustion in OpenFOAM”. In: *International Communications in Heat and Mass Transfer* 38 (2011), pp. 363–367.

[33] J.J. Keenan, D.V. Makarov, and V.V. Molkov. “Modelling and simulation of high-pressure hydrogen jets using notional nozzle theory and open source code OpenFOAM”. In: *International Journal of Hydrogen Energy* 30 (2016), pp. 1–10.

[34] A.C. Benim et al. “Numerical investigation of turbulent swirling flames with validation in a gas turbine model combustor”. In: *Applied Thermal Engineering* 110 (2017), pp. 202–2012.

[35] E. Fooladgar, C.K. Chan, and K.J. Nogenmyr. “An accelerated computation of combustion with finite-rate chemistry using LES and an open source library for In-Situ-Adaptive Tabulation”. In: *Computers and Fluids* 146 (2017), pp. 42–50.

[36] K.M. Pang et al. “Development and validation of a local time stepping-based PaSR solver for combustion and radiation modeling”. In: In Proceedings of 8th International OpenFOAM Workshop, 2013.

[37] Z.H. Che Daud et al. “3D electro-thermal modelling and experimental validation of lithium polymer-based batteries for automotive applications”. In: *International Journal of Energy Research* 40 (2016), pp. 1144–1154.

[38] G. Brösigke et al. “Fundamental investigation of heat transfer mechanisms between a rolling sphere and a plate in OpenFOAM for laminar flow regime.” In: *International Journal of Thermal Sciences* 111 (2017), pp. 246–255.

[39] S.Sandler et al. “Evaluation of the impact of the thermal shunt effect on the U-pipe ground borehole heat exchanger performance”. In: *Geothermics* 65 (2017), pp. 244–254.

[40] C. Cintolesi et al. “Numerical simulation of conjugate heat transfer and surface radiative heat transfer using the P1 thermal radiation model: Parametric study in benchmark cases.” In: *International Journal of Heat and Mass Transfer* 107 (2017), pp. 956–971.

[41] T. Holzmann. *Mathematics, Numerics, Derivations and OpenFOAM(R)*. 4th. Leoben: Holzmann CFD, 2017.

[42] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. 2nd. Philadelphia: R.T. Edwards, Inc., 2005.

[43] H.M. Ali. *Hybrid nanofluids for convection heat transfer*. Academic Press, 2020.

[44] C. Greenshields and H. Weller. *Notes on Computational Fluid Dynamics: General Principles*. Reading, UK: CFD Direct Ltd, 2022.

[45] D.R. Stull. *JANAF Thermochemical Tables*. Vol. 1. Clearinghouse, 1965.

[46] C.K. Westbrook and F.L. Dryer. “Simplified Reaction Mechanisms for the Oxidation of Hydrocarbon Fuels in Flames”. In: *Combustion Science and Technology* 27 (1981), pp. 31–43.

[47] *GRI-Mech 3.0*. `http://combustion.berkeley.edu/gri-mech/version30/text30.html`.

[48] C.P. Stone and F. Bisetti. “Comparison of ODE Solver for Chemical Kinetics and Reactive CFD Applications”. In: *52nd Aerospace Sciences Meeting*.

[49] S.B. Pope. “Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation”. In: *Combustion Theory and Modelling* 1.1 (1997), pp. 41–63.

[50] L. Liang, J. G. Stevens, and J. T. Farrell. “A dynamic adaptive chemistry scheme for reactive flow computations”. In: *Proceedings of the Combustion Institute* 32.1 (2009), pp. 527–534. ISSN: 1540-7489.

[51] F. Contino et al. “Coupling of in situ adaptive tabulation and dynamic adaptive chemistry: An effective method for solving combustion in engine simulations”. In: *Proceedings of the Combustion Institute* 33.2 (2011), pp. 3057–3064. ISSN: 1540-7489.

[52] S. R. Turns. *An Introduction to Combustion: Concepts and Applications*. 3rd. New York: McGraw-Hill, 2012.

[53] R.K. Hanson and S. Salimian. “Survey of Rate Constants in the N/H/O System”. In: (1984), pp. 361–421.

[54] A.H. Kadar. “Modelling Turbulent Non-Premixed Combustion in Industrial Furnaces: Using the Open Source Toolbox OpenFOAM”. MA thesis. TU Delft, 2015.

[55] ANSYS. *ANSYS FLUENT Theory Guide*. Canonsburg, Pennsylvania: ANSYS Inc., 2011.

[56] A. Kadar. *Source code of NOxFOAM*. `https://github.com/nae9on/OpenFOAM_MSc_Thesis_Project/tree/master/NOxFoam`. accessed: 2018-02-15.

[57] R. Borghi. “Turbulent combustion modelling”. In: *Progress in Energy and Combustion Science* 14.4 (1988), pp. 245–292. ISSN: 0360-1285.

[58] R. W. Bilger. “A Note on Favre Averaging in Variable Density Flows”. In: *Combustion Science and Technology* 11.5-6 (1975), pp. 215–217.

[59] T.H. Shih et al. “A New k-epsilon Eddy-Viscosity Model for High Reynold Number Turbulent Flows - Model Development and Validation”. In: *Computers & Fluids* 24(3) (1995), pp. 227–238.

[60] Y. Tominaga and T Stathopoulos. “Turbulent Schmidt numbers for CFD analysis with various types of flowfield”. In: *Atmospheric Environment* 41.37 (2007), pp. 8091–8099.

[61] V. Golovitchev, N. Nordin, and F. Tao. *Modeling of Spray Formation, Ignition and Combustion in Internal Combustion Engines, Annual Report*. Tech. rep. Chalmers University of Technology, Department of Thermo and Fluid Dynamics, 1998.

[62] O.M. Knio, H.N. Najm, and P.S. Wyckoff. “A Semi-implicit Numerical Scheme for Reacting Flow: II. Stiff, Operator-Split Formulation”. In: *Journal of Computational Physics* 154.2 (1999), pp. 428–467. ISSN: 0021-9991.

[63] A. Cuoci et al. “Numerical Modeling of Laminar Flames with Detailed Kinetics Based on the Operator-Splitting Method”. In: *Energy Fuels* 27 (2013).

[64] B.F. Magnussen and B.H. Hjertager. “On mathematical modeling of turbulent combustion with special emphasis on soot formation and combustion”. In: *Symposium (International) on Combustion* 16 (1977), pp. 719–729.

[65] J.A. VAN OIJEN and L.P.H. DE GOEY. “Modelling of Premixed Laminar Flames using Flamelet-Generated Manifolds”. In: *Combustion Science and Technology* 161.1 (2000), pp. 113–137.

[66] L. Ma and D. Roekaerts. “Modeling of spray jet flame under MILD condition with non-adiabatic FGM and a new conditional droplet injection model”. In: *Combustion and Flame* 165 (2016), pp. 402–423.

[67] L. Ma. “Computational modelling of turbulent spray combustion.” PhD thesis. Delft University of Technology, 2016.

[68] O. R. Darbyshire and N. Swaminathan. “A Presumed Joint PDF Model for Turbulent Combustion with Varying Equivalence Ratio”. In: *Combustion Science and Technology* 184.12 (2012), pp. 2036–2067.

[69] M.F. Modest and D.C. Haworth. *Radiative Heat Transfer in Turbulent Combustion Systems, Theory and Applications*. Cham: Springer, 2016. ISBN: 978-3-319-27291-7.

[70] M. Mancini, P.J. Coelho, and D.J.E.M. Roekaerts. *ERCOFTAC Best Practice Guide on CFD of combustion, chapter 4: Radiative Heat Transfer*. Tech. rep. ERCOFTAC, 2015.

[71] *greyMeanAbsorptionEmissionCoeffs for calculating the absorption coefficients of the gray gas assumption in OpenFOAM*. `https://github.com/OpenFOAM/OpenFOAM-5.x/blob/master/tutorials/combustion/reactingFoam/RAS/SandiaD_LTS/constant/radiationProperties`. accessed: 2026-01-05.

[72] H.K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. 2nd. Harlow: Pearson Education Limited, 2007. ISBN: 978-0-13-127498-3.

[73] Aokomoriuta. *The figure of law of the wall*. `https://en.wikipedia.org/wiki/Law_of_the_wall#/media/File:Law_of_the_wall_(English).svg`. accessed: 2016-10-15.

[74] B.E. Launder and D.B. Spalding. “The Numerical Computation of Turbulent Flows”. In: *Computational Methods Applied Mechanical Engineering* 3 (1974), pp. 269–289.

[75] T. von Kármán. “Mechanische Ähnlichkeit und Turbulenz”. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Fachgruppe 1 (Mathematik)* 5 (1930), pp. 58–76.

[76] A. E. Perry, J. B. Bell, and P. N. Joubert. “Velocity and temperature profiles in adverse pressure gradient turbulent boundary layers”. In: *Journal of Fluid Mechanics* 25.2 (1966), pp. 299–320.

[77] C.L.V Jayatilleke. “The influence of Prandtl number and surface roughness on the resistance of the laminar sublayer to momentum and heat transfer”. In: *Prog. Heat Mass Transfer* 1 (1969), p. 193.

[78] D.B. Spalding. *Monograph on turbulent boundary layers*. Tech. rep. Imperial College Mechanical Engineering Department.

[79] A.A. Townsend. *The Structure of Turbulent Shear Flow*. 2nd. Cambridge University Press, 1976.

[80] G. Kalitzin et al. “Near-wall behavior of RANS turbulence models and implications for wall functions”. In: *Journal of Computational Physics* 204 (2004), pp. 265–591.

[81] A.S. Dorfman. *Conjugate Problems in Convective Heat Transfer*. Taylor and Francis Group, LLC, 2010. ISBN: 978-1-4200-8237-1.

[82] S.V. Pantakar. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, 1980.

[83] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[84] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. 3rd. Berlin, Germany: Springer, 2002. ISBN: 3-540-42074-6.

[85] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics, An Advanced Introduction with OpenFOAM and MATLAB*. Cham: Springer, 2016. ISBN: 978-3-319-16873-9.

[86] C.J. Greenshields. *OpenFOAM user guide*. OpenFOAM Foundation Ltd., 2017.

[87] B.P. Leonard. “A stable and accurate convective modelling procedure based on quadratic upstream interpolation”. In: *Computer Methods in Applied Mechanics and Engineering* 19.1 (1979), pp. 59–98. ISSN: 0045-7825.

[88] C. Rhie and W. Chow. “A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation”. In: 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, 1982.

[89] *Description of SimpleFoam on the OpenFOAM wiki forums*. https://openfoamwiki.net/index.php/SimpleFoam. accessed: 2020-05-01.

[90] R.I Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62.1 (1986), pp. 40–65. ISSN: 0021-9991.

[91] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th. The Johns Hopkins University Press, Baltimore, 2013.

[92] S.R. Fulton, P.E. Ciesielski, and W.H. Schubert. “Multigrid Methods for Elliptic Problems: A Review”. In: *Monthly Weather Review* 114 (1986), p. 943.

[93] R.Tielen et al. “p-multigrid methods and their comparison to h-multigrid methods within Isogeometric Analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 372.37 (2020), p. 113347.

[94] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. 1st. London: Academic Press, 2001.

[95] P. Wesseling and C.W. Oosterlee. “Geometric multigrid with applications to computational fluid dynamics”. In: *Journal of Computational and Applied Mathematics* 128.1 (2001), pp. 311–334. ISSN: 0377-0427.

[96] T. Behrens. *OpenFOAM’s basic solvers for linear systems of equations*. Tech. rep. Technical University of Denmark, 2009.

[97] P. Wesseling. *Principles of Computational Fluid Dynamics*. 1st. Berlin: Springer, 1991. ISBN: 3-540-67853-0.

[98] H. A. van der Vorst. “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CGfor the Solution of Nonsymmetric Linear Systems”. In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), pp. 631–644.

[99] H. A. van der Vorst and C. Vuik. “GMRESR: a family of nested GMRES methods”. In: *Numerical linear algebra with applications* 1.4 (1994), pp. 369–386.

[100] A. Amritkar et al. “Recycling Krylov subspaces for CFD applications and a new hybrid recycling solver”. In: *Journal of Computational Physics* 303 (2015), pp. 222–237. ISSN: 0021-9991.

[101] S. Thomas, A. Baker, and S. Gaudreault. *Augmented Block-Arnoldi Recycling CFD Solvers*. 2023. arXiv: `2301.09480 [math.NA]`.

[102] P. Sonneveld and M.B. van Gijzen. "IDR(s): A Family of Simple and Fast Algorithms for Solving Large Nonsymmetric Systems of Linear Equations". In: *SIAM Journal on Scientific Computing* 31.2 (2009), pp. 1035–1062.

[103] P. Sonneveld. "On the convergence behaviour of IDR (s)". In: *Reports of the Department of Applied Mathematical Analysis, 10-08* (2010).

[104] S. Bnà et al. "PETSc4FOAM: A Library to plug-in PETSc into the OpenFOAM Framework". In: *PRACE White paper* (2020).

[105] *Sandia flame D test description on the ERCOFTAC QNET-CFD wiki forums*. `https://www.kbwiki.ercoftac.org/w/index.php/Description_AC2-09`. accessed: 2016-07-01.

[106] *Sandia flame D experimental data on the ERCOFTAC QNET-CFD wiki forums*. `https://kbwiki.ercoftac.org/w/index.php?title=Test_Data_AC2-09`. accessed: 2016-07-01.

[107] E.A. Daymo and M. Hettel. "Chemical Reaction Engineering with DUO and chtMultiRegionReactingFoam". In: 4th OpenFOAM User Conference 2016, Cologne, Germany, 2016.

[108] *Source code of multiRegionReactingFoam*. `https://github.com/TonkomoLLC`. accessed: 2017-01-15.

[109] M. el Abbassi, D.J.P. Lahaye, and C. Vuik. "Modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM". In: Proceedings of the Tenth Mediterranean Combustion Symposium, Naples, Germany, 2017.

[110] M. el Abbassi, D.J.P. Lahaye, and C. Vuik. "Modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM". In: *Numerical Mathematics and Advanced Applications ENUMATH 2019: European Conference, Egmond aan Zee, The Netherlands*. Springer. 2020, pp. 1137–1145.

[111] B. Damstedt et al. "Biomass cofiring impacts on flame structure and emissions". In: *Proceedings of the Combustion Institute* 31 (2007), pp. 2813–2820.

[112] R.S. Barlow and J.H. Frank. "Effects of turbulence on species mass fractions in methane/air jet flames". In: *Symposium (international) on combustion*. Vol. 27. 1. Elsevier. 1998, pp. 1087–1095.

[113] M. el Abbassi et al. "Non-premixed combustion in rotary kilns using OpenFOAM: the effect of conjugate heat transfer and external radiative heat loss on the reacting flow and the wall". In: *ICHMT Digital Library Online*. Begel House Inc. 2018.

[114] A. Revuelta et al. "Laminar Craya–Curtet jets". In: *Physics of Fluids* 16.1 (2004), pp. 208–211.

[115] J.D. Anderson Jr. *Fundamentals of Aerodynamics*. 5th. New York: McGraw-Hill, 2011.

[116] D.J.P. Lahaye et al. "Mitigating Thermal NOx by Changing the Secondary Air Injection Channel: A Case Study in the Cement Industry". In: *Fluids 2020* 5 (2020), p. 220.

[117] M. El Abbassi, D.J.P. Lahaye, and C. Vuik. "The Effect of Variable Air-Fuel Ratio on Thermal NOx Emissions and Numerical Flow Stability in Rotary Kilns Using Non-Premixed Combustion". In: *Processes* 9.10 (2021), p. 1723.

[118] F. Juretic. "cfMesh version 1.1, user guide".

[119] A. Banerjee et al. "NOx Minimization in Staged Combustion Using Rich Premixed Flame in Porous Media". In: *Combustion Science and Technology* 192.9 (2020), pp. 1633–1649.

[120] G. Scribano and M.-V. Tran. "Numerical investigation of a confined diffusion flame in a swirl burner". In: *European Journal of Mechanics - B/Fluids* 82 (2020), pp. 1–10. ISSN: 0997-7546.

[121] R. Edland et al. "Evaluation of NOx-Reduction Measures for Iron-Ore Rotary Kilns". In: *Energy & Fuels* 34.4 (2020), pp. 4934–4948.

[122] H. Wu et al. "Experimental Investigation of Cutting Nitrogen Oxides Emission from Cement Kilns using Coal Preheating Method". In: *Journal of Thermal Science* 30 (2021), pp. 1097–1107.

[123] S.B. Pope. *Turbulent flows*. New York: Cambridge University Press, 2000. ISBN: 9780521598866.

[124] P.A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. Oxford: Oxford University Press, 2004. ISBN: 9780198529491.

[125] D.C. Wilcox. *Turbulence Modeling for CFD*. 3rd. La Cañada, CA: DCW Industries, 2006. ISBN: 9781928729082.

[126] F.T.M. Nieuwstadt, B.J. Boersma, and J. Westerweel. *Turbulence*. Cham: Springer, 2016. ISBN: 978-3-319-31599-7.

[127] L. Davidson. *An Introduction to Turbulence Models*. Chalmers University of Technology, Department of Thermo and Fluid Dynamics, 2016.

[128] Christian Del Taglia et al. “Numerical and experimental investigation of an annular jet flow with large blockage”. In: *J. Fluids Eng.* 126.3 (2004), pp. 375–384.

[129] M. Chaabane, B. Adouane, and A. Benzaoui. “Composition and Stoichiometry Effects of Biogas as Fuel in Spark Ignition Engine”. In: *INTERNATIONAL JOURNAL OF AUTOMOTIVE AND MECHANICAL ENGINEERING* 15 (Mar. 2018), pp. 5036–5052. DOI: 10.15282/ijame.15.1.2018.11.0390.

[130] M. Mohamed et al. “A Comprehensive Experimental Investigation of NOx Emission Characteristics in Hydrogen Engine Using an Ultra-Fast Crank Domain Measurement”. In: *Energies* 17.16 (2024). ISSN: 1996-1073. DOI: 10.3390/en17164141. URL: https://www.mdpi.com/1996-1073/17/16/4141.

[131] S.S. Kulkarni and S.S. Barve. “CFD Simulation of Flue Gas Recirculation in a Rotary Kiln for NOx Reduction”. In: *International Journal of Innovative Research in Science, Engineering and Technology* 5.10 (2016), pp. 17539–17545.

[132] M. Zhang et al. “Numerical Investigation of Combustion Characteristics under Oxygen-Enriched Combustion Combined with Flue Gas Recirculation in a Cement Rotary Kiln”. In: *Energy* 64 (2014), pp. 615–625.

[133] C.E. Baukal. *Industrial combustion pollution and control*. CRC Press, 2003.

[134] D. Zhu et al. “Co-benefits of CO2 emission reduction and sintering performance improvement of limonitic laterite via hot exhaust-gas recirculation sintering”. In: *Powder Technology* 373 (2020), pp. 727–740.

[135] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2. 2024.

[136] D. Lahaye, F. Juretić, and M. Talice. “Modelling a Turbulent Non-Premixed Combustion in a Full-Scale Rotary Cement Kiln Using reactingFOAM”. In: *Energies* 15.24 (2022). ISSN: 1996-1073.

[137] B.Tekgül et al. “DLBFoam: An open-source dynamic load balancing model for fast reacting flow simulations in OpenFOAM”. In: *Computer Physics Communications* 267 (2021), p. 108073. ISSN: 0010-4655.

[138] *Source code of DLBFoam*. https://github.com/Aalto-CFD/DLBFoam. accessed: 2025-02-05.

[139] C. Greenshields. *Description of Modular Solvers in OpenFOAM*. https://cfd.direct/openfoam/free-software/modular-solvers. accessed: 2020-06-01.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my promoter and supervisor Prof. Kees Vuik for his unconditional support and patience, who kept pushing me through the difficult stage of my life long after my research. Thank you for always being there, and without judgment.

I would like to acknowledge my co-promoter, Dr. Domenico Lahaye, for his critical perspective, which consistently challenged me to sharpen my work, and for his continuous, careful review of my all my reports and publications with particular attention to mathematical detail.

I would like to express my acknowledgement to the examination committee members, for reading the dissertation and providing valuable feedback.

I am deeply grateful for the help of Prof. Dirk Roekaerts whenever I needed it and for his careful and unconditional review of this dissertation. His extensive knowledge on the topic, wisdom, commitment to education, and honest, constructive criticism are worthy of great respect. I wish we could have worked together more closely.

I would also like to thank Almatis B.V. for the financial support of my research and providing me a high performance computer where all colleagues at the faculty still benefit from long after my research. I would especially like to thank my industrial supervisor, Dr. Stefan Kuiper, for sharing his knowledge of rotary kilns, for his guidance within the company, and for his pragmatic perspective without losing sight of scientific relevance. His flexibility helped me find the right balance.

Lastly but not least, I would like to express my sincere gratitude to Dr. Gabriela (Gaby) Diaz Cortes and Dr. Roel Tielen for reviewing Chapters 2 and 3, respectively, and Dr. Qiyao (Alice) Peng for her guidance on the doctoral process. Beyond this, I am grateful to them for being my friends.

LIST OF PUBLICATIONS

SCIENTIFIC ARTICLES IN JOURNALS

3. **M. el Abbassi**, D.J.P. Lahaye and C. Vuik, *The Effect of Variable Air-Fuel Ratio on thermal NOx Emissions and Numerical Flow Stability in Rotary Kilns Using Non-Premixed Combustion.*, Processes 2021, 9, 1723, MDPI.

2. D.J.P. Lahaye, **M. el Abbassi**, C. Vuik, M. Talice and F. Juretić, *Mitigating thermal NOx by changing the secondary air inlet channel: A case study in the cement industry.*, Fluids 2020, 5, 220, MDPI.

1. **M. el Abbassi**, D.J.P. Lahaye and C. Vuik, *Modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM.*, Numerical Mathematics and Advanced Applications ENUMATH 2019, Vol 139, Springer, Cham, 22 August 2020.

SCIENTIFIC ARTICLES IN PROCEEDINGS [PEER-REVIEWED]

1. **M. el Abbassi**, D.R.A. Fikri, D.J.P. Lahaye and C. Vuik, *Non-premixed combustion in rotary kilns using OpenFOAM: the effect of conjugate heat transfer and external radiative heat loss on the reacting flow and the wall.*, Proceedings of the Ninth International Symposium On Turbulence Heat and Mass Transfer, Rio de Janeiro, Brazil, 10-13, July 2018.

SCIENTIFIC ARTICLES IN PROCEEDINGS

1. **M. el Abbassi**, D.J.P. Lahaye and C. Vuik, *Modelling turbulent combustion coupled with conjugate heat transfer in OpenFOAM.*, Proceedings of the Tenth Mediterranean Combustion Symposium, Naples, Italy, 17–21 September 2017.

TALKS AT CONFERENCES

2021

- OpenFOAM Workshop 2021, Dutch OpenFOAM User Group, Delft, The Netherlands, June 18, 2021 (virtual)

2019

- OpenFOAM Workshop 2019, Dutch OpenFOAM User Group, Delft, The Netherlands, June 20, 2019

2018

- Burgers Symposium, Conference Centre ‘De Werelt’, Lunteren, the Netherlands, 5-6 june, 2018
- Combura Symposium, NVV 2019 - Conference hotel Kontakt der Kontinenten Amersfoortsestraat, Soesterberg, the Netherlands, October 9-10, 2018
- 43rd Woudschoten conference 2018, Dutch-Flemish Scientific Computing Society, Woudschoten, the Netherlands, October 3-5, 2018
- Turbulence, Heat and Mass Transfer (THMT) 2018, 9th international symposium, The International Centre for Heat and Mass Transfer (ICHMT), Rio de Janeiro, Brazil, July 10-13, 2018

2017

- PhD Energy Event 2017, TU Delft, Delft, the Netherlands, November 2, 2017
- Combura Symposium, NVV 2017, Conference hotel Kontakt der Kontinenten Amersfoortsestraat, Soesterberg, the Netherlands, October 11-12, 2017
- Tenth Mediterranean Combustion Symposium, School of Polytechnic and Basic Sciences, College of Engineering, Naples, Italy, September 17-21, 2017

2016

- 2016 SIAM Annual Meeting, Society for Industrial and Applied Mathematics, Boston, Massachusetts, USA, July 11-15, 2016

CURRICULUM VITÆ

Mohamed EL ABBASSI

WORK HISTORY

2025 - ...	Energy Grid Engineer Stedin, Delft, The Netherlands
2022 - 2025	Simulation Engineer Aero/Thermal Mitsubischi Turbocharger and Engine Europe B.V., Almere, The Netherlands
2020 - 2022	Lecturer Automotive Engineering (Full-time) Hogeschool Rotterdam, Rotterdam, The Netherlands
2015 - 2019	Process Research Engineer Almatis B.V., Rotterdam-Botlek, The Netherlands
2012 - 2013	Gas Turbine Performance Engineer Alstom Power, Baden-Aargau, Switzerland

EDUCATION

2015-2025	PhD. in Applied Mathematics Delft University of Technology, Delft, The Netherlands *Thesis:* Mathematical modelling of turbulent combustion and conjugate heat transfer in rotary kilns
2012-2015	MSc. in Aerospace Engineering Delft University of Technology, Delft, The Netherlands *Thesis:* Conceptual design of a ’Flameless’ combustion chamber for aircraft engines.
2006 - 2011	BSc. in Aerospace Engineering Delft University of Technology, Delft, The Netherlands