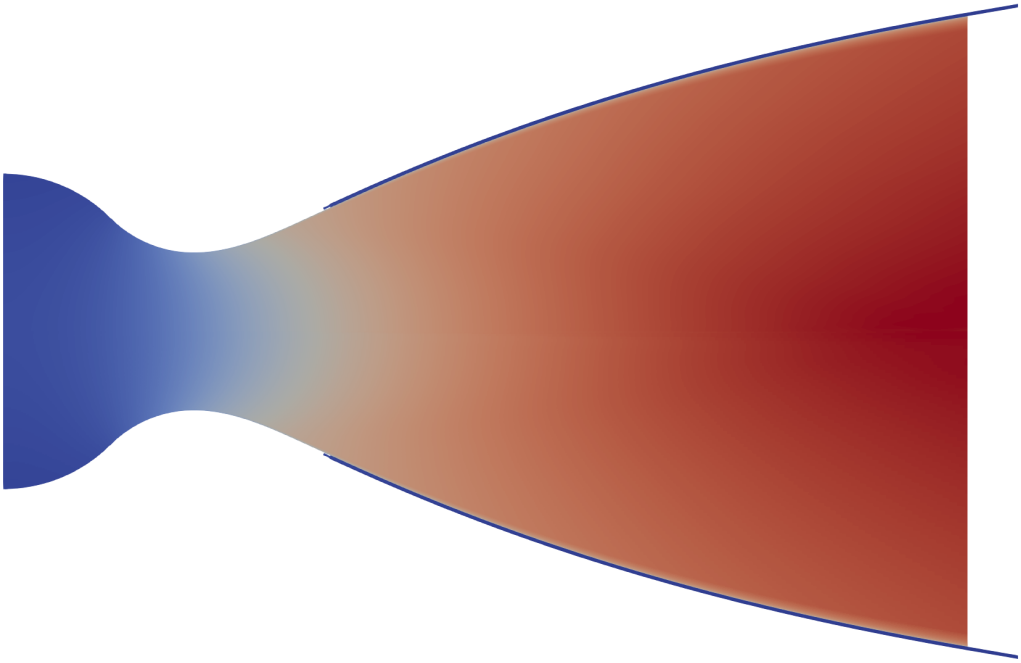

Holistic Simulation of Rocket Engine Nozzle

A Master Thesis at GKN Aerospace

Master's Thesis in Master's Programme Mobility Engineering

Alexander Hansson
Atle Nyberg

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Holistic Simulation of Rocket Engine Nozzle

Alexander Hansson, Atle Nyberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology
Gothenburg, Sweden 2025

Holistic Simulation of Rocket Engine Nozzle
ALEXANDER HANSSON ATLE NYBERG

Supervisor 1: Jan Östlund, GKN Aerospace
Supervisor 2: Alexandre Capitao Patrao, GKN Aerospace
Examiner: Thomas Grönstedt, Department of Mechanics and Maritime Sciences

Master's Thesis 2025
Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of a simulated rocket nozzle, showing the Mach contour of the nozzle and the cooling channels . The bottom part is simulated in STAR-CCM+ and the top part in ANSYS Fluent.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Holistic Simulation of Rocket Engine Nozzle
ALEXANDER HANSSON
ATLE NYBERG
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Methane-fueled rocket engines provide an attractive compromise between performance and engine complexity compared to engines utilizing other fuels such as kerosene or liquid hydrogen. For example, methane has a higher I_{SP} than kerosene while also having a higher storage temperature and density than liquid hydrogen, along with excellent properties for regenerative cooling.

Computational fluid dynamics is an invaluable tool for designing methane-fueled engines, but simulating a rocket nozzle can be a challenge due to the wide range of spatial and temporal scales along with the large temperature gradients between the coolant and nozzle flows. A traditional simulation approach divides the flows across multiple simulations coupled through boundary conditions, requiring multiple simulations which highlights an area that can be improved.

This thesis aims to investigate the possibility of carrying out an aerothermal simulation of an entire nozzle extension, with a holistic approach accounting for both the coolant and flame, as well as the heat transfer between regions in a single simulation. The project also aimed to compare two commercial CFD code Ansys Fluent and STAR-CCM+ using three different cases: 1) a 2D geometry used to identify solver settings to reach convergence, 2) a methane validation against previous work, 3) and a 3D rocket nozzle with a cooling channel for proof of concept.

For the 2D geometry case it was concluded that both simulation tools can achieve a converged solution with similar results, with a small difference likely due to the implementations of turbulence models in the tools. The methane validation case showed a good alignment for both CFD tools against the experimental results validating the method used for simulating the methane. For the 3D rocket nozzle the previous trend continued, showcasing a close match in results between the codes and previous work.

The project reached its final goal of developing a method of acceptable accuracy. This becomes clear when comparing with previous work. Both codes were proven to be suitable for the use of such simulations, allowing for a choice based on user experience along with other factors that are beyond the scope of this project, such as cost.

Keywords: CFD, Supercritical Methane, Reacting Species Transport, Rocket Nozzle Cooling, STAR-CCM+, ANSYS Fluent

Acknowledgements

This master thesis has been carried out as a collaboration between Chalmers University of Technology and GKN Aerospace Stallbacka, from which we received material and guidance. We would like to express our gratitude to our supervisors at GKN, Alexandre Capitao Patrao and Jan Östlund, for their invaluable guidance throughout the project. We also want to thank Tomas Grönstedt who has taken on the role as our Examiner at Chalmers. Lastly, we want to thank the employees at the department at GKN, especially Alexander Rodin, Jonathan Mårtensson and Kristie Goorden, for their assistance and support.

Alexander Hansson and Atle Nyberg, Gothenburg, June 2025

Nomenclature

Below is the nomenclature containing a list of commonly used acronyms, terms and variables in the report.

Acronyms

CFD	Computational Fluid Dynamics
NE	Nozzle Extension
HTD	Heat Transfer Deterioration
CEA	Chemical Equilibrium Applications
ESGR	Equivalent Sand Grain Roughness
CFL	Courant Friedrichs Lewy number

Terminology

Flamewall	The surface where the channel wall and the flame meet.
Hotwall	The surface that is in contact with the coolant, located across from the flamewall.
STAR-CCM+ or Star	Commercial CFD application developed by Siemens Digital Industries Software. STAR-CCM+2310r8 was used throughout the project.
ANSYS Fluent or Fluent	Commercial CFD software developed by Ansys, INC. Throughout the project version Fluent2024r1 was used.

Variables

M	Mach number
I_{sp}	Vacuum specific impulse
F_N	Vacuum thrust

$\dot{m}$	Mass flow rate
$\dot{m}''$	Mass flux
ρ	Density
p	Static pressure
p_0	Total pressure
T	Static temperature
T_0	Total temperature
μ	Dynamic viscosity
ν	Kinematic viscosity
k	Turbulent kinetic energy
ϵ	Rate of dissipation
ω	Specific rate of dissipation
μ_r	Viscosity ratio
I	Turbulent intensity
P_r	Prandtl number
C_P	Specific heat
κ	Conductivity
ΔP	Pressure drop
ΔT	Heat pick up
Q_{flame}	Heat transfer rate
D_H	Hydraulic diameter
X_t	Subscript t refers to a turbulent property

Contents

1
Introduction

This chapter introduces the thesis and provides an overview of the project background, the aim and the project broken down to individual tasks and questions.

1.1 Project Background

GKN Aerospace Sweden is a leading aerospace manufacturer that specializes in high performance engine components for civil and military aircraft. The branch is based in Trollhättan and is a part of the larger global company GKN Aerospace, that serves over 90% of the world's aircraft and engine manufacturers. GKN Aerospace Sweden also has a background producing rocket components for the European Space Agency and ArianeGroup. For the Ariane 6 rocket, the company develops and manufactures the nozzle extension for the Vulcain engine as well as the turbines.

When developing a nozzle extension (NE) computational fluid dynamics (CFD) is an invaluable tool that allows for a detailed analysis of the aerothermal performance without conducting any real-life testing. In order to accurately simulate the performance of a nozzle one must accurately simulate the chemically reacting flame and the cryogenic coolant along with the heat transfer between the two regions. However, there are significant differences between the regions in terms of temperature, velocity and fluid properties, resulting in a stark difference in the behavior of the flows. The traditional approach to simulate the nozzle is to split the regions, solving each flow with its own solver while being linked with a boundary condition. Despite being an accurate way to simulate each individual region, the method requires the setup of multiple simulations and multiple iterations between them to ensure convergence. The simulations may therefore take longer than needed and demand more manual input than desired, leaving room for improvement in the approach.

1.2 Aim of the Project

The purpose of this project is to investigate the inclusion of all the relevant phenomena and regimes in one single simulation including the cryogenic coolant, the chemically reacting flame and the heat transfer between the regions. The goal is to eliminate the need of setting up multiple simulations and the need to manually iterate between them. For this purpose two CFD codes were investigated, Star-CCM+ and ANSYS Fluent, in order to compare their capabilities to simulate a nozzle extension.

1.3 Tasks and Questions

The following questions are answered in this thesis:

- How can a CFD simulation be set up to capture the flows in both regions?
- Do the CFD codes produce different results and if they do, why?
- How well do the flows converge based on mass and energy continuity?
- Which code is more accurate compared to experiments and previous work?
- Which code is preferable to use in this type of simulation?
- Is it a viable approach to simulate both of the flows in the same simulation?

2
Theory

This chapter covers the theory relevant for this project. Starting with the fundamentals of rocket nozzles, cooling and CFD and then branching out into the relevant models and schemes used within the CFD codes.

2.1 Rocket Nozzles

A convergent-divergent rocket nozzle is a device through which heated gases are accelerated from subsonic to supersonic speeds according to the Mach-area relation, shown in equation 2.1. The equation shows that for Mach numbers below one, a decrease in area will increase the fluid velocity and that an increase in area instead decreases the velocity. The exact opposite is true for Mach numbers above one. In a rocket nozzle the convergent part of the nozzle accelerates a subsonic flow to sonic speeds by decreasing the area to its critical value where the flow becomes choked. The divergent part then accelerates the sonic flow to supersonic speeds by increasing the area, resulting in a total acceleration from subsonic to supersonic velocities through the convergent-divergent nozzle. The acceleration increases the kinetic energy through the nozzle and the energy is sourced from the flow's internal energy which is why the temperature in the combustion chamber can reach 3600 K [6, 17].

$$\frac{dA}{A} = (M^2 - 1)\frac{dV}{V} \tag{2.1}$$

Convergent-divergent nozzles are often used in rockets in order to expand the exhaust gases from the combustion chamber for the purpose of maximizing thrust according to equation 2.2, which shows the formula for net thrust for a rocket engine [7]. The flow is accelerated through the nozzle, thus increasing the exit velocity and thrust.

$$F_N = \dot{m}V_{exit} + A_{exit}(p_{exit} - p_0) \tag{2.2}$$

A common measure of efficiency for rockets is the specific impulse which is the ratio of thrust over mass flow of propellant.

$$I_{SP} = \frac{F_N}{\dot{m}_p} \tag{2.3}$$

Where $\dot{m}_p$ is the propellant mass flow, including both the fuel and oxidizer. Specific impulse gives both an effective way of determining thrust for differing mass flows and provides a measure of engine efficiency which has been accepted as a standard when comparing reaction engines [12].

The exhaust gases coming out of the combustion chamber can reach temperatures of up to 3600 K [4] which introduces the need for cooling the nozzle walls. There are two main groups of methods of cooling, which are generally referred to as "steady state" and "heat sink". There are also alternative ways of cooling such as "film cooling" but they are considered as additional techniques that are often used alongside the general methods. Steady state cooling refers to methods that reach an equilibrium in heat transfer and temperature, such as regenerative, dump and radiation cooling, while heat sink refers to unsteady heat transfer techniques. Regenerative cooling, the main focus of this project, is a method that involves running either the fuel or the oxidizer through channels in the nozzle walls, picking up energy from and cooling the solid. A benefit with regenerative cooling, beyond being an effective cooling method, is that the increase in enthalpy of the coolant in turn increases the thrust and performance of the engine [17].

This project considered a methane-oxygen rocket engine with a fuel rich combustion, commonly used to increase the thrust specific impulse [17]. For the regenerative cooling the methane fuel was used due to its higher specific heat compared to the liquid oxygen [17] , allowing it to absorb more heat. Additionally, if a leak would occur from the cooling channel to the flame, the methane would be the safer option. This is because leaking oxygen would react with the remaining fuel in the nozzle and form hot-spots on the nozzle wall leading to further damage, while a fuel leak would have no oxygen to react with inside the nozzle.

2.1.1 Flow Characteristics

The coolant flow is a subsonic supercritical flow going through long and narrow channels. Due to the nature of the flow a wide range of physical phenomena must be taken into account in order to accurately capture the heat transfer. The flow is often turbulent with recirculation and mixing leading to an introduction of three-dimensional effects that will affect the heat transfer within the fluid and between wall and fluid. Additionally, the heat transfer from the wall is asymmetrical which can lead to thermal stratification. In these cases, there will be a separation of a hot and a cold part of the coolant, impacting the conductivity of the flow. In order to capture the heat flux from the wall a model needs to be used for proper conjugate heat transfer. Further, when wall roughness is taken in account, accurate modeling of its induced turbulence and pressure loss is needed for capturing its effect on heat transfer. Lastly, the flow is supercritical, meaning its pressure is above the critical pressure for the coolant, but still below the level of turning it into a solid, which results in a combination of liquid-like and gas-like properties with large variations in the fluid properties throughout the channels.

Heat transfer deterioration (HTD) is a phenomenon that occurs when a supercritical fluid reaches a certain pseudo critical temperature for a given pressure. The pseudo critical temperature is equivalent to a critical temperature when a liquid boils to a gas, but for a supercritical fluid the change happens over a temperature span where the fluid transitions from liquid-like to gas-like [4]. The span is characterized by

drastic changes in fluid properties, which can be seen in figure 2.1 for the specific heat of methane. When the temperature rises to the pseudo critical temperature close to the wall a gas-like layer will form that will act like an insulating layer, decreasing the heat transfer from the wall [4]. This can result in hot-spots along the wall and issues with convergence for CFD solvers. To avoid HTD in CFD there are generally two approaches: either the pressure can be raised making the changes in fluid properties less pronounced as can be seen in figure 2.1 or the boundary conditions can be changed to ensure the fluid stays under the pseudo critical temperature.

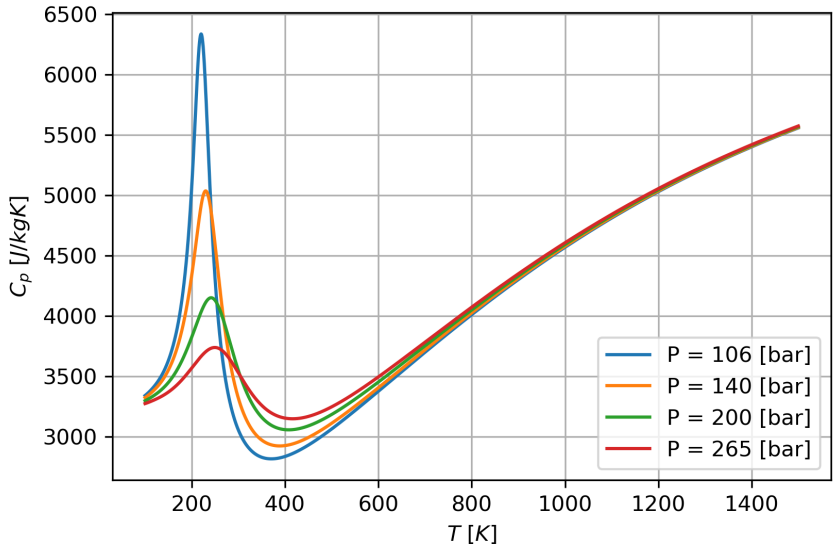

Figure 2.1: Specific heat for supercritical methane at different pressures. The plot is generated with property data of methane.

The flow of the exhaust gases, on the other hand, contains regions of both sub- and supersonic flow, thus needing a higher temporal resolution compared to the flow of the coolant in order to keep the simulation stable and accurate. Similarly to the coolant flow the exhaust gases also have a large temperature gradient close to the wall and the heat transfer between the two needs to be accurately resolved. The exhaust gases in the nozzle flame are the products of the combustion that occurred in the upstream combustion chamber and contain a multitude of different molecules that are chemically reacting with each other in the flow. In CFD the different molecules are represented with species, where the governing equations are solved for each (species transport), making simulations of the flow computationally expensive. The reactions are also captured in a reaction mechanism by calculating the reaction rates and products from temperature based expressions. The large velocities combined with the curvature of the nozzle cause turbulent flow which influences the heat transfer to the wall as well as the boundary layer. Therefore the exhaust flow, alike the coolant flow, needs a fine spatial resolution near the wall and in the boundary layer, further increasing the computational cost.

With all the above in mind there are multiple phenomena within a nozzle that adhere to widely different temporal and spatial scales. There is both sub and supersonic

flow inside of the nozzle, chemical reactions, heat transfer and a separate subsonic flow of coolant in the channel walls. This makes the problem challenging to simulate within a single CFD simulation and therefore the customary approach is to simulate the coolant and nozzle flows separately, while iterating between multiple simulations and updating the boundary conditions of each simulation depending on the output of the others, which is a manual and time-consuming procedure.

2.2 Computational Fluid Dynamics

Throughout the project Computational Fluid Dynamics (CFD) was used as a tool to simulate and analyze the different flows. Both Siemens STAR-CCM+ and ANSYS Fluent were used to solve the governing equations and to model all the relevant physics. Several cases and simulations were included in the project modeling conjugate heat transfer, turbulence, chemical reactions and supercritical fluids. In CFD the fluids are modeled as continua and represented by a set of governing equations, partial differential equations, that are the product of the physical laws of conservation of mass, momentum and energy [4]. This section covers the principles of CFD along with the theory behind the methods used to solve the cases throughout the project.

2.2.1 Governing Equations

The governing equations of fluid dynamics represent the motion and properties of a fluid flow. CFD uses spatial and temporal discretizations of the governing equations to arrive at a set of algebraic equations. This is done by breaking down the continuum equations into discrete points in space and time, making them solvable using numerical methods. This section will cover the steady state continuum equations in differential form.

The law of conservation of mass is represented by the continuity equation 2.4, it states that the mass of fluid entering a volume must equal the mass of fluid exiting it.

$$\nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.4}$$

Conservation of momentum states that the rate of change of momentum of a fluid element is equal to the sum of forces acting on it and is given by equation 2.5 where the first term represents the spatial change of momentum and the terms on the right hand side show the forces: pressure gradient force, viscous stresses and lastly body forces (usually gravity).

$$\nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f} \tag{2.5}$$

The third governing equation is for the conservation of energy which states that energy cannot be created or destroyed, only transformed from one form to another. In the context of fluid dynamics these forms can be summarized as kinetic energy,

potential energy and internal energy. See equation 2.6. Where e, k and Φ are the internal energy, thermal conductivity and the viscous dissipation respectively.

$$\nabla(\rho e\mathbf{u}) = -\nabla\cdot(p\mathbf{u}) + \nabla\cdot(k\nabla T) + \Phi \tag{2.6}$$

The energy equation is also solved for the solid domain by using Fourier's law of heat conduction, equation 2.7, which is how conjugate heat transfer was taken into account. Together these equations make up the fundamental principles of the motion and properties of fluids in a flow. As can be seen in the equations there is no governing equation quantifying pressure which is why pressure velocity coupling is needed which is covered in section 2.2.2.

$$\mathbf{q} = -k\nabla T \tag{2.7}$$

Where q is the heat flux, k is the thermal conductivity of the material and ∇T is the temperature gradient.

2.2.2 Pressure Velocity Coupling

The discretized form of the Navier-Stokes equations shows that velocity has a linear dependence on pressure with no governing equation to quantify it. To solve this a method called pressure velocity coupling is used, where an equation for the fluid flow pressure is commonly formulated from the momentum and continuity equations [4]. This section contains an overview of the methods to reach an equation for pressure, commonly starting from the momentum equation, see equation 2.8, where the viscous stresses are now represented with τ.

$$\nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \nabla\tau = -\nabla p \tag{2.8}$$

When discretized for an arbitrary fluid element "P" in a mesh with localized influence from the neighboring cells "N" the equation can be written as equation 2.9 where "a" contain the coefficients for velocity in matrix form.

$$\mathbf{a_p u_P} + \sum_N \mathbf{a_N u_N} = -\nabla p \tag{2.9}$$

Now move the term representing the influence of the neighboring cells over to the left hand side and divide with $\mathbf{a_P}$ to get an equation for the velocity vector.

$$\mathbf{u_P} = \frac{1}{\mathbf{a_p}}\sum_N \mathbf{a_N u_N} - \frac{1}{\mathbf{a_p}}\nabla p \tag{2.10}$$

Now to ensure the enforcement of conservation of mass and compressibility the equation is multiplied with the density in order to incorporate the continuity equation.

$$\rho\mathbf{u_P} = \frac{\rho}{\mathbf{a_p}}\sum_N \mathbf{a_N u_N} - \frac{\rho}{\mathbf{a_p}}\nabla p \tag{2.11}$$

The pressure equation is then given by the divergence of equation 2.11 and remember that the continuity equation states that $\nabla\cdot(\rho\mathbf{u}) = 0$, thus arriving at the pressure equation 2.12.

$$\nabla \cdot (\frac{\rho}{\mathbf{a_p}} \nabla p) = \nabla \cdot (\frac{\rho}{\mathbf{a_p}} \sum_N \mathbf{a_N} \mathbf{u_N}) \tag{2.12}$$

This equation can then be used in a series of different approaches to solve the pressure velocity coupling. There are two main methods of solving the pressure equation: 'Segregated' and 'Coupled' solvers. Segregated solvers calculate the pressure separately from velocity, while the coupled solver handles both simultaneously. The coupled solver is generally preferred when dealing with large gradients. The derivation follows the process given by [4] and [5].

2.2.3 Turbulence Modeling

Turbulence is a phenomenon encountered in the majority of flows and is characterized by chaotic and unsteady behavior. It is a three-dimensional phenomenon that enhances mixing, diffusion of fluid flow gradients and dissipation of kinetic energy. It has a wide range of both spatial and temporal scales making it difficult to simulate [5]. By resolving all the spatial and temporal scales the Navier-Stokes equation will solve for turbulence as well and this is called a direct numerical simulation (DNS). In these types of simulations the only error would be that of the discretizations which is known, but due to the wide range of scales the computational intensity exceeds practical limits for most engineering application, which is why in most cases models are used instead. The most popular of which are the Reynolds Averaged Navier-Stokes equations (RANS) and Large Eddy Simulation (LES). LES hinges upon resolving the larger scale turbulence, like DNS, while modeling the smaller scales (sub grid scales) using sub-grid models of which there are multiple. Both DNS and LES are generally considered too computationally expensive for most engineering applications, making RANS models a popular choice [4, 5].

RANS treats turbulence by averaging over all and any unsteady phenomena, i.e treating all unsteadiness as turbulence, thus arriving at a new set of governing equations containing fluctuating and mean flow properties. These equations contain non-linear terms that needs to be modeled. This approach is less computationally intensive compared to the previously mentioned which makes it the most popular in engineering applications. The fundamental assumption is that instantaneous flow quantities (ϕ) can be split up in a mean ($\overline{\phi}$) and fluctuating part (ϕ') [5]:

$$\phi(x_i, t) = \overline{\phi}(x_i, t) + \phi'(x_i, t) \tag{2.13}$$

Where $\overline{\phi}$ is the average of the instantaneous property ϕ, this averaging can be done through multiple methods but the most common are time and ensemble averaging (averaging over multiple realizations of the fluid flow). Here the non linear terms in the equations for the fluctuating part needs to be modeled which will be covered in the following section.

2.2.4 $k-\epsilon$ Turbulence Model

The $k-\epsilon$ model is a widely used 2-equation approach to close the RANS equations. It is defined by the flow properties found in its name: turbulent kinetic energy (k), which represents the sum of all normal Reynolds stresses (equation 2.14), and the dissipation rate (ϵ), which governs the conversion of turbulent kinetic energy into heat (equation 2.15)[3].

$$\frac{\partial k}{\partial t} + \overline{v}_j \frac{\partial k}{\partial x_j} = P_k + g_i \beta \frac{\nu_t}{\sigma_\theta} \frac{\partial \overline{\theta}}{\partial x_i} - \epsilon + \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial k}{\partial x_j}\right) \tag{2.14}$$

$$\frac{\partial \epsilon}{\partial t} + \overline{v}_j \frac{\partial \epsilon}{\partial x_j} = \frac{\epsilon}{k} c_{\epsilon,1} P_k + c_{\epsilon,1} g_i \frac{\epsilon}{k} \frac{\nu_t}{\sigma_\theta} \frac{\partial \overline{\theta}}{\partial x_i} - \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right) \tag{2.15}$$

Where P_k is the production term from the k equation and is calculated using equation 2.16.

$$P_k = \nu_t \left(\frac{\partial \overline{v}_i}{\partial x_j} + \frac{\partial \overline{v}_j}{\partial x_i}\right)\frac{\partial \overline{v}_i}{\partial x_j} \tag{2.16}$$

Where the turbulent viscosity, ν_t, is calculated using equation 2.17.

$$\nu_t = c_\mu \frac{k^2}{\epsilon} \tag{2.17}$$

The two transport equations, 2.14 and 2.15, are similar to each other because the equation for ϵ is derived from the k-equation using dimensional analysis. Where the terms from the k equation are scaled with constants and multiplied with $\frac{\epsilon}{k}$ to transform it into an equation for dissipation. The constants used in the equations have the standard values presented below.

$$c_\mu = 0.09, \quad c_{\epsilon,1} = 1.44, \quad c_{\epsilon,1} = 1.92, \quad \sigma_k = 1, \quad \sigma_\epsilon = 1.3$$

2.2.5 The $k-\omega$ Turbulence Model

The $k-\omega$ model is a transformed version of $k-\epsilon$, which has been adjusted to handle adverse pressure gradients without overestimating the sheer stresses, which is a common problem with the original $k-\epsilon$ model [3]. The transport equation for ω is derived from the ϵ equation using the following relation:

$$\omega = \frac{\epsilon}{k\beta^*} \tag{2.18}$$

Where $\beta^* = c_\mu$. The ω equation is an epsilon equation incorporating the relation above and it takes the same form as both the k and ϵ equations and is shown in equation 2.19

$$\frac{\partial}{\partial x_j}(\overline{v}_j \omega) = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial \omega}{\partial x_j}\right) + \alpha \frac{\omega}{k} P_k - \beta \omega^2 + \frac{2}{k}\left(\nu \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} \tag{2.19}$$

Where α and β are constants with values 0.44 and 0.0828 respectively. The turbulent viscosity ν_t is the calculated from k and ω using equation 2.20.

$$\nu_t = \frac{k}{\omega} \tag{2.20}$$

2.2.6 Shear Stress Transport

The turbulence model $k-\omega$ shear stress transport (SST) is a well established method for simulating turbulence, it combines the benefits of both $k-\omega$ and $k-\epsilon$ turbulence models. $k-\omega$ is used in the inner boundary layer as it captures the near wall effects more accurately than $k-\epsilon$, that instead is used in the outer regions of the boundary layer and the rest of the domain. A blending function is used to switch between the models depending on the location in the boundary layer and is given by equation 2.21. Where F_1 is a distance function that assumes unity in the inner parts of the boundary layer and gradually deceases to zero towards the outer regions. This way the model keeps track of where to use the respective models and utilizes that through setting β values corresponding to the respective model. The reason for blending the two methods is to avoid the weaknesses of each model. $k-\epsilon$ over predicts the stresses in adverse pressure gradients, such as over an airfoil, and it needs near wall modifications, therefore the $k-\omega$ method is used near the wall. The $k-\omega$ method on the other hand is difficult to initialize for free stream boundary conditions which is why $k-\epsilon$ is used in the bulk [3].

$$\beta_{SST} = F_1\beta_{k-\omega} + (1-F_1)\beta_{k-\epsilon} \tag{2.21}$$

2.2.7 Wall Roughness Modeling

Modeling of wall roughness is mainly relevant when considering the coolant flow. When designing the cooling channels a design parameter is the pressure loss of the fluid traveling through them, wall roughness is a contributor to increased losses and have a significant effect on the heat transfer [13]. In ANSYS Fluent and STAR-CCM+ the wall roughness is set as a boundary condition that affects drag and heat transfer on walls for turbulent wall-bounded simulations. The Fluent implementation follows an empirically derived equation that is a modified version of the the "Law of the wall" to include effects of wall roughness, see equation 2.22.

$$\frac{u_p u^*}{\tau_w/\rho} = \frac{1}{\kappa} ln(E\frac{\rho u^* y_p}{\mu}) - \Delta B \tag{2.22}$$

Where κ is the von Karman constant, $u^* = C_\mu^{1/4}\kappa^{1/2}$ and $\Delta B = \frac{1}{\kappa}ln(f_r)$. K_s is the roughness height that goes into f_r which is a roughness function that quantifies the effect of surface roughness on the velocity profile near the wall. ΔB is the addition to the equation that takes wall roughness into account by changing the intercept in the velocity profile, essentially changing where on the near wall velocity profile that the effects starts to act.

STAR-CCM+ follows the same base principal in their "Rough displaced origin" model but the implementation is different. They follow a variation of equation 2.22 with dependence on the wall y^+ value and the height of the wall displacement

(y_0^+) due to the wall roughness, see equation 2.23. Like Fluent, Star uses different equations depending on the level of roughness which depends on a wall roughness parameter R^+ which is limited by a smooth and rough value R^+_{Smooth} and R^+_{Rough} corresponding to fully smooth or fully rough surfaces. y_0^+ is then set depending on the relation between the calculated R^+ and the limiting values (i.e larger than R^+_{Rough}, in between, smaller than R^+_{Smooth}), R^+ is given by equation 2.24 where r is the ESGR and u^* is the velocity scale.

$$u^+ = \frac{1}{\kappa} ln(\frac{y^+ + y_0^+}{y_0^+}) \tag{2.23}$$

$$R^+ = \frac{r\rho u^*}{\mu} \tag{2.24}$$

The main difference between the models are the functions used to determine ΔB in Fluent and y_0^+ in Star. For further reading see [16] for STAR-CCM+ and [8] for Fluent.

2.2.8 RANS Turbulent Heat Transfer

Both ANSYS Fluent and STAR-CCM+ use the turbulent Prandtl number as an input for the calculations of turbulent thermal conductivity. The implementation is based on Fourier's law but alters the thermal conductivity using turbulent properties, see equation 2.25.

$$\overline{q} = -(\kappa + \frac{\mu_t C_p}{P_{rt}})\nabla T \tag{2.25}$$

Where κ is the thermal conductivity of the fluid, μ_t the turbulent viscosity and P_{rt} the turbulent Prandtl number. This is the most common implementation as it is the standard heat flux model employed in RANS [16]. The equation shows the importance of correctly assigning the Prandtl number when heat transfer is studied in turbulent flows as it directly influences the conductivity of the fluid flow. The turbulent Prandtl number is defined in the following way:

$$P_{rt} = \frac{C_P \mu_t}{\kappa_t} \tag{2.26}$$

where C_P, μ and κ are the pressure specific heat, viscosity and thermal conductivity of the fluid. The equation shows that the Prandtl number is the ratio between the mass and thermal diffusivity. The conductivity of the material is determined through first calculating the turbulent viscosity which is done with the help of the turbulence model then using the equation above to determine the conductivity as the Prandtl number is usually set as an input parameter and known beforehand. By changing the turbulent Prandtl number the effect will be a scaling of the turbulent thermal conductivity.

According to [19] the two CFD codes calculate the turbulent viscosity using different methods, for Fluent the following is true:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{2.27}$$

While STAR-CCM+ uses:

$$\mu_t = \rho C_\mu f_\mu k T \tag{2.28}$$

Where f_μ and T are a damping function for close wall effects and the turbulent time scale respectively. f_μ assumes unity away from the wall thus removing its effects in the $k-\epsilon$ regions assuming the simulation has a fine enough mesh resolution and uses the SST model. T takes on the following value away from the wall, i.e in the $k-\epsilon$ region.

$$T = \frac{k}{\epsilon} \tag{2.29}$$

Thus the STAR-CCM+ implementation match that of Fluent away from walls, as long as the mesh is fine enough to allow for a sufficiently good transition between $k-\epsilon$ and $k-\omega$ there should be no difference in the model leading to the codes calculating the turbulent viscosity identically. As turbulent viscosity influences the flow thermal conductivity through the turbulent Prandtl number and equation 2.25 a difference in implementation can lead to difference in thermal loads if the mesh resolution is too low. For further reading on the subject see [19]. The calculation of the turbulent viscosity is identical in the $k-\omega$ regions depending on the sub-models used in Star, according to the Fluent theory guide [8] and STAR-CCM+ documentation [16] although the wall functions differ [19] requiring a fine boundary layer resolution for the codes to compute identically.

2.2.9 Species Transport Modeling

In species transport the code simulates a gas made up of multiple gaseous components, solving the transport equations for each of them. ANSYS Fluent and STAR-CCM+ calculates the mass-fraction of each component using a convection-diffusion equation, equation 2.30, which is how the locations and ratios of gas components are captured. In the equation R_i is the net rate of production by chemical reactions, S_i includes sources from dispersed phases and user-defined sources and $\vec{J_i}$ denotes the diffusion flux of species which is prevalent in locations of concentration gradients where it acts to smooth out the concentrations.

$$\frac{\partial}{\partial t}(\rho Y_i) + \nabla \cdot (\rho \vec{v} Y_i) = -\nabla \cdot \vec{J_i} + R_i + S_i \tag{2.30}$$

The diffusion term $\vec{J_i}$ is calculated using different methods depending on if the flow is turbulent or laminar. Only the turbulent methods are shown in this report. The diffusion is calculated for each species i according to equation 2.31

$$\vec{J_i} = -\left(\rho D_{i,m} + \frac{\mu_t}{\mathrm{Sc}_t}\right)\nabla Y_i - D_{T,i}\frac{\nabla T}{T} \tag{2.31}$$

Here $D_{i,m}$ is the i:th species mass diffusion coefficient which describes how prone that species is to being diffused. The $D_{T,i}$ is the thermal (Soret) diffusion coefficient

and describes a fluid elements tendency to move along temperature gradients. For each species follows a term of diffusion induced transport of enthalpy:

$$\nabla \cdot \left[\sum_{i=1}^{n} h_i \vec{J}_i \right] \tag{2.32}$$

Which is non negligible when the species specific lewis number is far from unity, meaning that mass diffusivity of the species is far from the thermal diffusivity of the entire multi component gas [1].

2.2.10 The Laminar Finite-Rate Chemistry Model

This method used within both the STAR-CCM+ and Fluent simulations models the consumption and production of species using Arrhenius expressions, meaning equations describing the temperature dependence of reaction rates, and ignores direct effects of turbulent fluctuations. It does however take turbulence into account indirectly as the instantaneous values of velocity are used within the definition. This method is generally inaccurate for turbulent flows but in cases where the chemistry is relatively slow compared to the flow timescale and the turbulence-chemical interaction is minimal, such as super sonic flames, it delivers an acceptable accuracy [1].

The net source of each species, meaning their production or destruction, is calculated as the sum of Arrhenius reaction sources over the N_R (number of different reactions) that each species can participate in:

$$R_i = M_{w,i} \sum_{r=1}^{N_R} \hat{R}_{i,r} \tag{2.33}$$

where $M_{w,i}$ is the molecular weight of species i, $\hat{R}_{i,r}$ is the Arrhenius rate of creation/destruction of the i:th species for the given reaction r. Each reaction r is written in a general form that describes the creation and destruction of involved species along with the resulting energy released or captured in the reaction:

$$\sum_{i=1}^{N} \nu'_{i,r} \mathcal{M}_i \underset{k_{b,r}}{\overset{k_{f,r}}{\rightleftharpoons}} \sum_{i=1}^{N} \nu''_{i,r} \mathcal{M}_i \tag{2.34}$$

This equation can be seen as showing reactants on the left hand side and the products on the right, although the equation shows that the reactions can be reversed. N describes the number of species in the reaction, $\nu'_{i,r}$ and $\nu''_{i,r}$ are the stoichiometric coefficients for the reactant respectively the product i in reaction r, $\mathcal{M}_i$ denotes the species i and $k_{f,r}$ and $k_{b,r}$ are the forward and backward rate constants for reaction r, calculated using Arrhenius expressions as seen in general form in equation 2.35. These equations are imported in "CHEMKIN" formatted files to both Fluent and STAR-CCM+ from which the solvers calculate the resulting energy and products in each cell of the domain. A source term within the energy equation accounts for the release or absorption of energy due to reactions [1].

$$k = A \cdot e^{-\frac{E_a}{RT}} \tag{2.35}$$

Where k is the reaction rate constant, A is the pre-exponential factor, E_a is the activation energy, R the universal gas constant and lastly T which is the temperature [8].

3
Methods

The goal of the project was to simulate and capture all the physical phenomena in a liquid rocket nozzle within a single CFD simulation. However, as discussed in the background and theory sections, these simulations are prone to diverge due to the stark difference between the flows. To increase the chance of reaching the targets the project was therefore divided into three different cases, where the first case was a simple two-dimensional flat plate. This simulation aimed to be a starting point where different models and approaches could be applied and tested to see if the solutions would converge. These were models and approaches such as including the real gas properties of the methane, the reactions of the exhaust species in the flame and the heat transfer between the fluids. The subsequent cases would then increase step-by-step in geometrical complexity until arriving at a three-dimensional simulation of a liquid rocket nozzle including all the relevant physics.

The cases are as follows:

1. A two-dimensional simulation featuring two fluid domains separated by a flat plate of steel. One domain contains the hot supersonic flow of a rocket flame and the other the cryogenic, supercritical methane coolant flow. The goal of this case was to identify and rectify any differences in results between the two CFD codes along with developing a method for simulating both the coolant and nozzle flows.

2. A representative three-dimensional cooling channel geometry with wall roughness based of the work of Pizzarelli et al. [13]. The simulation aimed to be an introduction to accurately simulating a cryogenic and supercritical fluid, where the results was compared to Pizzarelli's and Ricci et al [14] for validation.

3. A simulation of a generic rocket nozzle internally cooled with methane. The simulation was done using the method developed in the previous cases to represent the nozzle flame and coolant. This case was included for the evaluation of the holistic approach, discerning if it was a viable method for simulating a rocket engine nozzle.

All the cases were simulated in both ANSYS Fluent and in STAR-CCM+ with the same models, meshes and schemes to allow for a fair comparison between them. The comparison was done with properties relevant to the flame such as I_{sp}, velocity profiles and Mach contours, but also for properties relevant for the other materials like wall temperatures, coolant heat pickup and pressure loss.

3.1 General Methodology

This section presents the general methods shared by the cases of this report, this includes material properties such as how the coolant, flame and the walls were modeled, but also how the turbulence was modeled and how convergence was measured. The simulation methods used throughout the project, that is covered in this section, was a result of the evaluation step of the first case, where models were investigated in order to match the results of both CFD codes.

3.1.1 Coolant

The liquid methane used to cool the nozzle extension was both cryogenic and supercritical, meaning that the ideal gas law no longer applied. Instead the methane was represented as a real gas, in CFD this was done using tabulated data of the fluid properties in the form of a real gas property table (rgp). The rgp file used in this project was supplied by GKN Aerospace and contains fluid properties as functions of temperature and pressure. The file also contained saturation levels that are used to describe the transition between liquid and gas-state in the critical region, however in this project the methane was treated as a single phase liquid due to it being supercritical.

One drawback with rgp tables was that they were not supported in STAR-CCM+. To represent the super critical methane a set of tables were constructed according to the specifications in the STAR-CCM+ user manual [16] using a Python script developed during the project. One table per fluid property was constructed and imported into STAR-CCM+. To assign the fluid properties the "real gas" model was enabled along with "User defined EOS" within which the "compressible" option needed to be enabled. The interpolation within the tables was automatically done by STAR-CCM+. The interpolation was compared to manual interpolation within the tables to verify the results and was shown to be accurate.

3.1.2 Flame

The flame was described using a multi-component gas with reacting species transport. The model used for the reactions was the finite rate "Laminar flame concept" under the assumption that most reactions take place in the supersonic region where it delivers an acceptable accuracy. The reactions in the exhaust gases were described using the "Zhukov-Kong Mechanism" [18] with 23 species, which is a reduced reaction model of a larger one for methane and oxygen developed specifically for liquid rocket engine combustion. The data contains tabulated material properties for each of the individual species along with data of the possible reactions. Tests of the data have been carried out by Zhukov [18] and shows good agreement with experimental results within temperatures of $870-1700\ K$ and pressures of $2-500\ atm$ $(2.03-506.6\ bar)$.

The inlet species mass fractions were obtained using NASAs "Chemical Equilibrium

Applications" (CEA) simulation code. CEA considers a large amount of molecules, but in the first case, only molecules with a mass fraction larger than 10^{-4} were included in the inlet boundary conditions.

The material properties for the mixture and for each species can be solved in different ways, ranging from tabulated data to empirical equations and relations. Below in table 3.1 and 3.2 the settings for both the mixture and the species are shown. The settings were selected to match the two programs as well as possible, but there were some differences between them. First of all, to be able to describe the thermal conductivity and viscosity with kinetic theory in STAR-CCM+ the "Mathur-Saxena Averaging" option had to be used. However, the averaging method slightly differs from "ideal-gas-mixing-law". Both of them uses mole fractions but the "Marthur-Saxena" includes an additional averaging term. Secondly, STAR-CCM+ uses "Chapman-Enskog" to calculate the viscosity for the species which is almost identical to Fluents implementation of "kinetic-theory" but it uses a slightly smaller constant in the equation. Additionally, in Fluent a "last species" had to be selected, this species is typically the species with the largest mass fraction, which was H_2O in the exhaust gases. In cases where the mass fraction of all species do not sum to one, the difference is added to the "last species" to ensure continuity. This is in contrast to STAR-CCM+ where the species mass fractions are normalized ensuring that they sum to one.

Table 3.1: Settings for the material properties of the mixture as they are named in the respective software.

Material property	**Fluent**	**STAR-CCM+**
Density	"Ideal-gas"	"Ideal gas"
Specific heat	"Mixing-law"	"Mass-weighted mixture"
Thermal conductivity	"Ideal-gas-mixing-law"	"Mathur-Saxena Averaging"
Viscosiy	"Ideal-gas-mixing-law"	"Mathur-Saxena Averaging"
Mass diffusivity	"Kinetic-theory"	"Kinetic-theory"

Table 3.2: Settings for the material properties of the species as they are named in the respective software. The properties not mentioned are set to their default values.

Material property	**Fluent**	**STAR-CCM+**
Specific heat	"piecewise-polynomial"	"Thermodynamic Polynomial Data"
Thermal conductivity	"kinetic-theory"	"Kinetic-theory"
Viscosity	"kinetic-theory"	"Chapman-Enskog"

3.1.3 Walls

The walls in the first and third cases were made of steel with the standard material properties from ANSYS Fluent [9], see table 3.3. They were also modeled as no-slip

walls with thermally coupled interfaces to the fluids. Steel was chosen as the thermal conductivity was similar to that of the material employed by GKN Aerospace.

Table 3.3: Material properties of steel.

ρ $[kg/m^3]$	C_p $[J/(kgK)]$	k $[W/(mK)]$
8030	502.48	16.27

3.1.4 Turbulence

To simulate the turbulence in the flows the $k-\omega$ SST turbulence model was used throughout all cases. While both CFD codes utilized the standard coefficients for $k-\omega$ presented by Menter in 1994 [11], Fluent had a alternative approach and defined them by their inverses.

An important parameter when simulating turbulence with heat flux is the turbulent Prandtl number, as it is used to calculate the turbulent contribution to the fluid thermal conductivity and therefore also the heat flux [9], see section 2.2.7 and equation 2.25. The default values for the turbulent Prandtl number was 0.85 in Fluent and 0.9 in STAR-CCM+. Because this parameter was seen to offset the temperature profiles of the walls between the CFD codes, the Prandtl number was set to the same value of 0.85 in both of the CFD codes. The Prandtl number typically varies between 0.7 to 0.9 [10]. Neither of these values were inherently more correct than the other, but when comparing CFD codes it is important that they have the same value. With similar reasoning the turbulent Schmidt number was also changed to 0.7 in both programs, where previously it had been 0.9 in STAR-CCM+ and 0.7 in Fluent.

Between the CFD codes there were some differences in the default turbulence settings. The first difference was that viscous heating (viscous dissipation) was included in the governing equations in STAR-CCM+ through the viscous stress tensor and could not be disabled [16]. Viscous dissipation is the transformation of kinetic energy of the flow to thermal energy. The phenomena is important for high speed flows and is recommended to include for compressible flows such as the flame [9]. In Fluent viscous heating needed to be enabled under "Viscous Model". Another difference was that the "compressibility correction" setting was included in STAR-CCM+ and not in Fluent by default. The "compressibility correction" setting is used to improve the prediction of free shear layers at high mach numbers, but it has a negative impact on the boundary layers [9]. Since there are no free shear layers in the nozzle or channel flow the setting was deactivated.

3.1.5 Convergence

To determine convergence in the simulations some engineering quantities was utilized such as the heat pick up and pressure drop in the channel, the average temperature of the walls and I_{sp} for the nozzle flow. All the quantities should stabilize and converge to a certain value when the solution had converged.

Mass and energy imbalances of the flows were also utilized to ensure that continuity and the energy were conserved. The mass imbalances were calculated according to equation 3.1 and is the difference between the inlet and outlet mass flow rate normalized by the inlet. The energy imbalances are calculated with built in functions in the programs. In fluent the "HeatTransferRate" was used at the boundaries and is a conserved quantity for reacting flows [8]. A similar function was used in STAR-CCM+ called "Boundary Heat Flux" and is recommend to use to check energy conservation [16]. Both of the functions were normalized with the heat transfer across the walls to get the percentage of the error.

$$\left|\frac{\dot{m}_{inlet} - \dot{m}_{outlet}}{\dot{m}_{inlet}}\right| = \Delta\dot{m} \tag{3.1}$$

3.2 Case 1: Flat Plate

This section covers the numerical approach for simulating the first case including all relevant physics. This case was kept computationally light to allow for experimentation; different models, schemes and functions were investigated in order to find where the differences between Fluent and STAR-CCM+ lied. By keeping the case light, results could be generated faster and allow for a more efficient study of the two CFD codes before moving on to the more computationally heavy cases.

3.2.1 Meshing

A structured two-dimensional mesh was generated using ANSYS ICEM and divided into three regions, one for each of the fluid regions and one for the solid. The channel and wall height were both set to 1 mm and the flame was set to 100 mm in height. The domain length was set to 500 mm and the cooling channel was elongated 50 hydraulic diameters upstream from the wall to ensure that the boundary layer would be fully developed before reaching the wall where heat transfer occurs, see figure 3.1 for a representation of the geometry. The hydraulic diameter was calculated using equation 3.2, where b was equal to one since the channel is two-dimensional.

$$D_H = \frac{2ab}{a+b} = \{b = 1\} = \frac{2a}{a+1} \tag{3.2}$$

The mesh was set up with 300 cells in the flow direction for the whole domain. To properly capture the boundary layer flow and heat transfer the mesh was refined in the vertical direction ensuring a maximum y^+ value below 1. This ensured a good resolution of the viscous boundary layer and can therefore capture its effects on the flow and heat transfer.

Table 3.4: Case 1 mesh settings.

Region	Vertical cells	First layer height	Growth ratio
Flame	75	$2.5 \cdot 10^{-7}$	1.2
Wall	10	10^{-4}	1
Channel	30	$5 \cdot 10^{-8}$	1.2

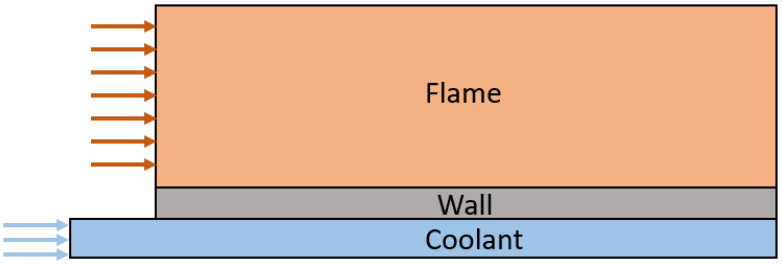

Figure 3.1: Case 1 geometry showing the different regions. The figure is not to scale.

Figure 3.1 also shows what will be referred to as the 'flamewall' and 'hotwall' throughout the thesis. The flamewall refers to the wall where the flame and the channel wall meet, while the hotwall refers to where the coolant and the channel wall meet, across from the flamewall.

3.2.2 Models and Schemes

A coupled solver was utilized in order to capture the high temperature gradient close to the wall more accurately. The simulations were steady state, using second order upwind as the spatial discretizations scheme. Energy was turned on to account for the heat transfer. The solver was implicit and pressure-based.

3.2.3 Boundary Conditions

Tabulated and separated by region below are the boundary conditions used for simulating the flat plate case, see table 3.5. For the coolant flow, the inlet and outlet conditions were set to resemble those of [13] in order to set up a realistic operating condition for supercritical methane. However, the heat flux presented in [13] was lower than the heat flux in the simulation and to avoid HTD the mass flow rate was doubled. The inlet temperature from the Pizzarelli report [13] was a static temperature and not a stagnation temperature. However when calculating the true stagnation temperature it only slightly differed from the static temperature because of the low velocity and the two can therefore be approximated to be equal at the inlet.

The boundary conditions for the flame were largely based on [15] that specified an inlet stagnation pressure of 106 *bar*, an expansion ratio of 16 : 1 and a fuel oxidizer ratio of 3.5. These parameters were then ran through NASAs CEA (Chemical Equilibrium Applications) website that calculated species mass fractions, seen in table 3.7, and the rest of the boundary conditions needed for the simulation, table 3.6. From the simulation output the values at the throat were taken as inlet conditions and values at the exit as the outlet for the CFD simulations. For the channel wall domain all boundaries were set to walls with the flamewall and hotwall being thermally coupled while the other walls being left as adiabatic.

Table 3.5: Coolant boundary conditions.

Boundary	**Type**	**Values**
Inlet	Mass flux inlet	$\dot{m}^{"} = 14000\ [kg/m^2 s]$, $T_0 = 136.5\ [K]$, $I = 5\%$, $\mu_r = 10$
Outlet	Pressure outlet	$P = 103.9\ [bar]$, $I = 5\%$, $\mu_r = 10$
Hot wall	no-slip, coupled heat transfer, no surface roughness	-
Cold wall	Symmetry	-

Table 3.6: Flame boundary conditions.

Boundary	**Type**	**Values**
Inlet	Pressure-far field	$M = 3.323$, $P = 89091[Pa]$, $T = 2264.98\ [K]$, $I = 5\%$, $\mu_r = 10$
Outlet	Pressure outlet	$P = 89091\ [Pa]$, $I = 5\%$, $\mu_r = 10$
flamewall	no-slip, coupled heat transfer, no surface roughness	-
Symmetry wall	Symmetry	-

Table 3.7: Species mass fraction for the flame inlet.

Species	**Inlet**
H2O	0.41234
CO2	0.29079
CO	0.20289
OH	0.04653
O2	0.03437
H2	0.00618
O	0.00597
H	0.00077
HO2	0.00016

3.2.4 Evaluation

To identify differences between the two codes, this case was repeated with various settings, models and schemes, leading to the method described in this section. This approach aimed to find an accurate and efficient method for the final case 3 simulation. The primary focus was the implementation of the turbulence models along

with their coefficients and constants. Additionally, interpolation methods, species transport and heat conductivity in materials were investigated as potential sources of error. This was mainly conducted through cross checking between the Fluent theory guide [8] and Star-CCM+ user guide [16], along with cross referencing field properties from the results of both codes.

3.3 Case 2: Representative Cooling Channel

The models for the material properties were kept the same for the coolant flow and was now investigated with greater detail. The flame was excluded from the simulation and a three-dimensional mesh was utilized. This case used the open geometry and operating conditions from test 2 of [14] to compare against their experimental results to determine the precision of both codes.

Two STAR-CCM+ simulations were run using different turbulent Prandtl number, the standard values of STAR-CCM+ (0.7) and Fluent (0.85), to determine which was most suitable for simulating the coolant flow and yielded the most realistic result. The chosen value would then be used in both codes throughout the project.

3.3.1 Meshing

The geometry and mesh was provided by GKN Aerospace and was based on the open geometry of a test rig from [14]. The mesh was constructed to replicate the mesh used for their numerical investigation and contains approximately three million cells with refinement towards the channel walls to keep a wall y^+ value below 1. The domain consists of a solid body with a channel running through it, see figure 3.2, to decrease the computational intensity the domain was split in half using a symmetry boundary condition, which is marked with a blue color in the figure.

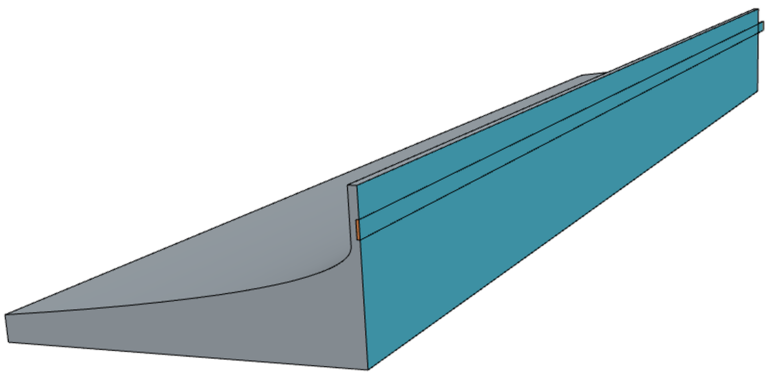

Figure 3.2: Case 2 geometry.

3.3.2 Models and Schemes

The $k-\omega$ SST turbulence model was used for the fluid domain. The energy was turned on according to the requirements of heat transfer and using a real gas to simulate the supercritical methane. The solid material was copper, described using a constant heat transfer coefficient of $k_s = 345\ W/m^2$ given by [13] with the other properties given by the standard material database in ANSYS Fluent. SIMPLEC was used to solve pressure velocity coupling in a pressure based solver as instructed in [14] where 2nd order upwind was used as the spatial scheme. STAR-CCM+ did not allow the use of SIMPLEC in steady simulations and therefore SIMPLE was utilized instead. Both CFD codes were set to use the "rough displaced origin" method for simulation of wall roughness, in STAR-CCM+ the setting shares the name with the model while in Fluent it's the standard "rough" model. Fluent was run using viscous heating enabled as it was shown to cause an offset in wall temperatures during the first case. STAR-CCM+ was run using both its standard turbulent Prandtl and Schmidt numbers, then with those of Fluent along with a deactivated compressibility correction for both simulations.

3.3.3 Boundary Conditions

All boundary conditions on the solid body were set to walls, one handling the influence of the flame by setting a constant heat flux across the boundary and the rest being adiabatic or coupled for heat transfer between fluid and solid. The in and out flow were described using a velocity inlet and pressure outlet with the values given in [14], test 2, which also provided the flamewall heat flux. The equivalent sand grain roughness (ESGR), like in [14], was attained through completing a cold case without any heat flux and iterating upon the wall roughness height in order to match the pressure drop from the experiment, roughly 3.2%. Temperatures, pressures and mass flows were taken from experiment "0A" in [14] . A cold case was completed for all simulations meaning it was repeated for Fluent and both Star simulations.

3.3.4 Convergence

Both Fluent and STAR-CCM+ had difficulties converging using the standard settings which required a method for getting the solver to initiate correctly and minimize the computational time. One physical phenomenon was introduced at a time, starting with enabling surface roughness then the flame-wall heat flux, both of which were gradually increased towards their final values to keep the solver stable. The under relaxation values were ramped for a stable initiation of the simulation, the start and end values of each under-relaxation coefficient is itemized below.

- **Pressure:** 0.15-0.3
- **Velocity:** 0.5-0.7
- **Energy:** 0.5-0.8
- **Turbulent viscosity:** 0.5-0.8
- **Turbulence:** 0.5-0.8
- **Other settings**: 0.5

3.3.5 Evaluation

The wall temperatures from the simulations were plotted against each other along with the experiment data from [14], at three different locations showed in figure 3.3. The two Star simulations, using the standard settings and adapted to fit Fluent, were both compared to the results of Fluent to showcase the difference in the outcome. Apart from the temperatures the boundary heat flux was investigated along with the wall y^+ to identify any differences.

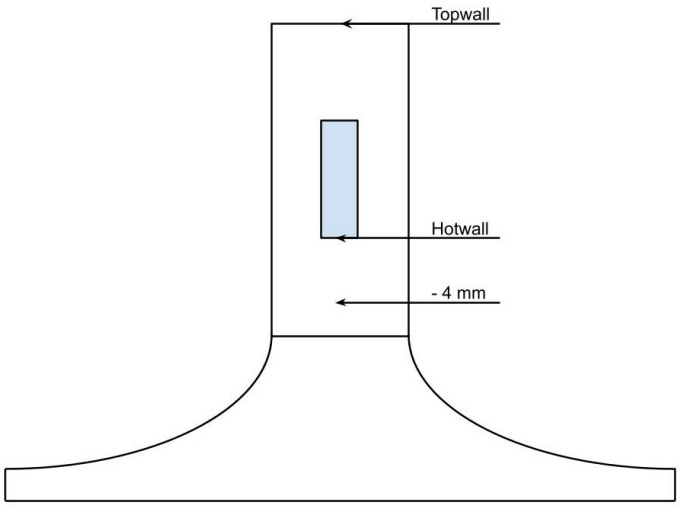

Figure 3.3: Sensor positions for wall temperature.

3.4 Case 3: Rocket Nozzle

The final case of the project featured a complete holistic simulation with a representative nozzle and nozzle wall geometry in which both the coolant, flame and heat transfer between them were simulated. The reaction scheme and real gas properties from the first case were now used in a more complex geometry. Alike the second case this case was compared against a reference [15], where performance measures such as specific impulse, thrust and mass flow were compared for both CFD codes.

3.4.1 Geometry and Meshing

The nozzle geometry was sourced from [15] and was based on a function describing the shape of the nozzle flame wall. The channel geometry was provided by GKN Aerospace and corresponds to the angular segment of a single cooling channel, see figure 3.4. ANSYS ICEM was used to generate the three-dimensional meshes, with a 1° angular segment of the geometry being meshed to make the simulations less demanding. The meshing strategy for the flame can be seen in figure 3.4 for the radial-axial plane and figure 3.5 for the radial-tangential plane. Figure 3.4 shows that the flame has been split in 3 blocks in the axial direction in order to describe the boundary of each block using a circle segment arch that stays orthogonal to the walls it connects to, this allows for the cells in the mesh to follow the shape

of the nozzle wall, ensuring the flow enters orthogonally to the faces of each cell to minimize numerical errors.

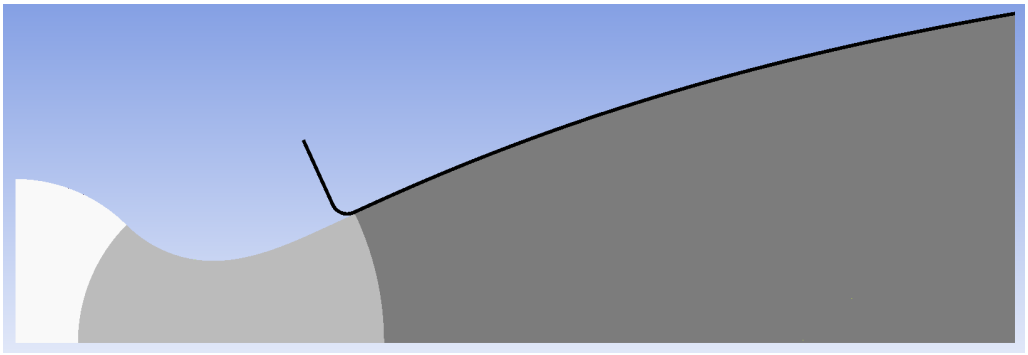

Figure 3.4: Blocking of Nozzle periodic faces showing the combustion chamber, throat and the nozzle extension.

Figure 3.5 shows the mesh strategy for the cross section of the flame where the goal was to have a fine tangential resolution close to the nozzle wall and decrease the tangential cell count in the rest of the domain. A "y-block" was used to capture the wedge connecting to the periodicity axis to ensure hexa-elements everywhere in the domain. The "o-grid" at the wall of the nozzle was used in order to decrease the number of cells in the tangential direction approaching the bulk. This blocking allowed for a fine tangential resolution at the nozzle wall while keeping the tangential resolution lower in the rest of the flame domain where gradients in the tangential direction were considered to be negligible.

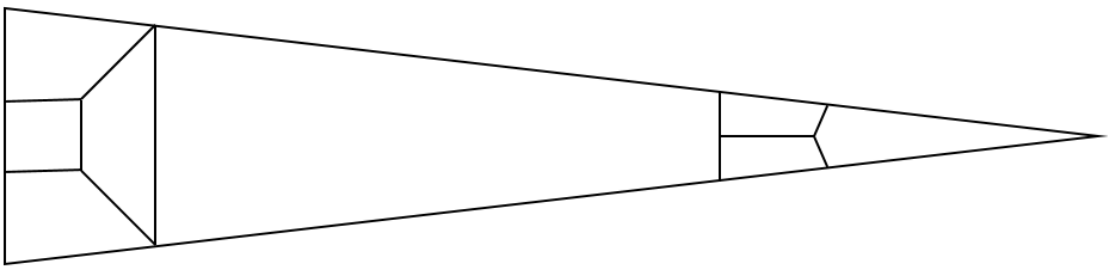

Figure 3.5: Blocking inlet/outlet plane.

The cooling channel was set to maintain contact with the entirety of the nozzle extension flamewall, at the end of which it would dump out the coolant. This simplification of the regenerative cooling method was done in order to decrease the computational cost. To initiate the boundary layer in the coolant flow and minimize any numerical errors both the inlet and outlet was extended. Initially, a circle arch shaped pipe was used to initiate the inlet, as can be seen in figure 3.4, but this

was seen to induce flow separation. To mitigate the separation the direction of the flow was reversed and was extended, like the outlet, using a straight pipe instead of a bend, see figure 3.6, where the sections colored gray are the extensions of the channel and coolant.

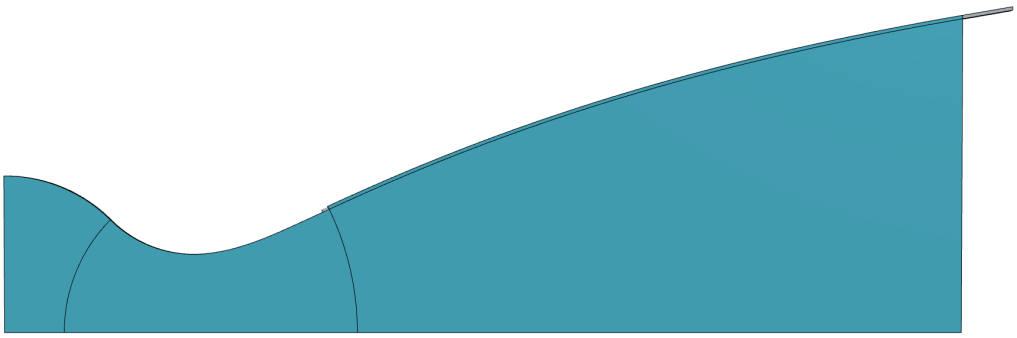

Figure 3.6: Geometry after changing channel flow direction.

The blocking of the channel was dictated by that in the flame, to achieve a 1-to-1 cell mapping, which is favored when trying to capture conjugate heat transfer, the cells making up the base of the channel was required to match those on the top of the flame. The resulting blocking setup can be seen in figure 3.7 where blue represents the coolant domain, gray the channel walls and orange the flame domain. Figure 3.7 also show the main areas of refinement: The top of the flame and the bottom of the coolant, as the majority of the heat transfer was shown to occur over these boundaries during preliminary simulations of the third case.

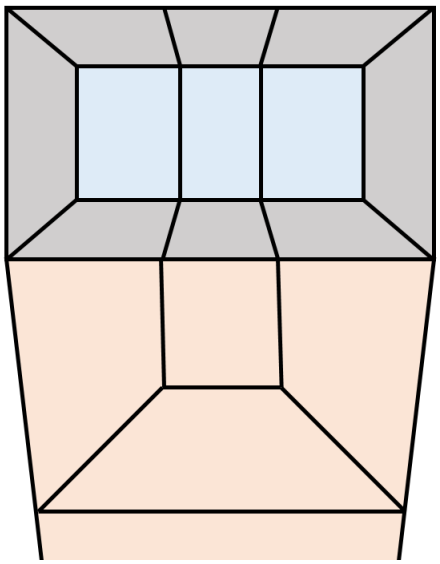

Figure 3.7: Blocking of channel inlet/outlet plane, including the flame wall.

The mesh quality was also assessed through comparison with the article for the design of a representative rocket nozzle [15]. The simulation was run to be compared against the performance measures in the report. This check was considered

necessary as the geometry that they supplied was, in this case, adapted to a three-dimensional domain and the fact that only a one degree section of the nozzle was simulated, which could cause high aspect ratio cells thus a lower resolution of certain flow features and increased numerical smearing [9].

General mesh information can be seen in table 3.8. These parameter values were attained through a mesh study where the first layer height and cell number was iterated upon until both the average flame wall temperature and pressure drop changes were below 1% between meshes. The cell numbers ranged from 5.1 to 12.8 million summed over all regions for the five meshes made for the third case. The final mesh only had 1-to-1 cell matching between the channel and coolant in contrast to everywhere in the previous meshes. This was because the 1-to-1 mapping caused a bloated cell number (in the flame and channel wall domains) when making necessary refinements to the coolant domain.

Table 3.8: Nozzle mesh information.

Region	**First layer height** [m]	**Growth ratio**	**Cells**
Flame	10^{-6}	1.2	1M
Coolant	10^{-8}	1.2	4.5M
Channel	10^{-4}	1	2.7M

To allow for the low cell height in the channel the geometric tolerances within ICEM and STAR-CCM+ were adjusted. This involved decreasing the minimum allowable wall distance in Star and the interface geometric tolerances, as well as the the general "Model/units" settings in ICEM. Fluent had a minimum wall distance of 10^{-12} as default and did therefore not need to be modified.

3.4.2 Boundary Conditions

The coolant was simulated using a mass flow inlet and a pressure outlet, allowing the mass flow rate to be controlled. Using density, velocity and area obtained from CEA, the mass flow rate through each channel as calculated to be 0.149 kg/s, with equation 3.3 and 3.4. Given that the whole of the cooling system is not included and that parts such as the throat also required cooling, 65% of the flow might typically be routed towards cooling the throat and combustion chamber of the nozzle but due to the occurrence of HTD the full mass flow was used. The assumption that the flow would loop back according to the design of most regenerative cooled nozzles the mass flow was set to 0.3 kg/s. This would be an unrealistic design as the methane would be split up into multiple flows cooling the whole nozzle but as the optimization and design of the system lay outside the bounds of this project the simplification was made.

The pressure at the coolant outlet was estimated to be 2.5 times larger than the combustion chamber pressure, based on a rough estimation of the pressure loss through the channels. This did not need to be entirely realistic, as the project mainly focused on method development. The 2.5 pressure ratio was then assumed to make up for

the pressure loss throughout the entire channel and keep the coolant at a sufficient pressure for injection into the combustion chamber. All the boundary conditions for the coolant are presented in table 3.9.

$$\dot{m} = \rho \cdot v \cdot A \tag{3.3}$$

$$\frac{\dot{m}_{oxidizer}}{\dot{m}_{fuel}} = 3.5, \qquad \dot{m}_{oxidizer} + \dot{m}_{fuel} = \dot{m}_{flame} \tag{3.4}$$

Table 3.9: Coolant boundary conditions.

Boundary	**Type**	**Values**
Inlet	Mass flux inlet	$\dot{m} = 0.3$ $[kg/s]$, $T_0 = 137.1$ $[K]$
Outlet	Pressure outlet	$P = 265$ $[bar]$
Coolant walls	no-slip, coupled heat transfer, no Surface roughness.	-
Sides	Rotationally periodic	-

Similarly to case 1 the boundary conditions for the flame were based on [15] with the value for stagnation pressure, expansion ratio and O/F used as an input for a CEA simulation. However, for this case the values were taken at the chamber inlet and at the nozzle exit since the entire nozzle was simulated. The values for the boundary conditions for the flame are shown in table 3.10.

Table 3.10: Flame boundary conditions.

Boundary	**Type**	**Values**
Inlet	Pressure inlet	$P = 106$ $[bar]$, $T = 3615.03$ $[K]$
Outlet	Pressure outlet	$P = 89091$ $[Pa]$
Chamber and throat flamewall	no-slip, constant temperature, no surface roughness	$T_{wall} = 700[K]$
Nozzle extension flamewall	no-slip, coupled heat transfer, no surface roughness	-
Sides	Rotationally periodic	-

3.4.3 Simulation Strategy

To save time when initializing the simulation a feature called "Patching" was used, where user defined field functions representing flow properties (e.g. velocity, pressure, temperature) dependent on the x-position were defined. The CEA simulations could be used to get values from the inlet, throat and outlet which could then be

fitted to a function and used to describe the field properties at each x-coordinate. This method gave a rough first guess from which the solver could begin from.
Due to the computational cost of the third case the simulation of the channel and flame were initially split up in STAR-CCM+. Simulating the flame and the channel alone without any coupling to let the different flows converge before being allowed to interact would save time and allow for any error in the simulation or mesh to be more easily found and rectified. A temperature of 1000 K was initially set as a boundary condition on the wall separating the channel and the flame, both sides of the wall were then allowed to converge separately before coupling the wall and removing the thousand Kelvin condition. This method enabled the control of the CFL according to the domain which also saved time and stabilized the simulation. Additionally the flame was initialized using air in STAR-CCM+ to establish a starting point for field properties, then the multi-component gas material was implemented, lastly the reactions were activated.

In STAR-CCM+ the function to turn on and off specific continua has allowed for iteration between the two regions. By turning off a continua the field variables remain constant thus allowing for the iteration to function without exporting and importing boundary values. This iteration was done in order to get each region to converge as effectively as possible, due to the large difference in velocities between the regions it was beneficial to have a variable CFL number depending on the region. Fluent did not feature this functionality leading to running a coupled simulation for the entirety of the simulation. The switch between regions was done every time the active regions imbalances settled, approximately three iterations between regions were needed for the flow to converge enough for running the simulation fully coupled effectively.

4

Results

The results section covers the simulation outputs and is divided into sub-sections belonging to the different cases. The results mainly focus on the difference in results from the two CFD codes and the rate of convergence for each program. The second and the third case sections contain comparisons with previous work.

4.1 Flat Plate

The following section contains the results from the simulations of the flat plate and a comparison between STAR-CCM+ and Fluent. The comparison includes the temperature profiles of the walls, species mass fractions at the outlet, y^+ values, convergence and lastly an analysis of the turbulent quantities.

4.1.1 Wall Temperature

In figure 4.1 the flame and hotwall temperatures along the x-axis are shown for both codes. The wall temperatures produced in STAR-CCM+ and in Fluent followed a similar profile, but there was an average difference of 13.8 K at the flamewall and 3.5 K at the hotwall. The temperature profiles from STAR-CCM+ also had some fluctuations that can be seen in figure 4.1, while the temperature profiles from Fluent was smoother.

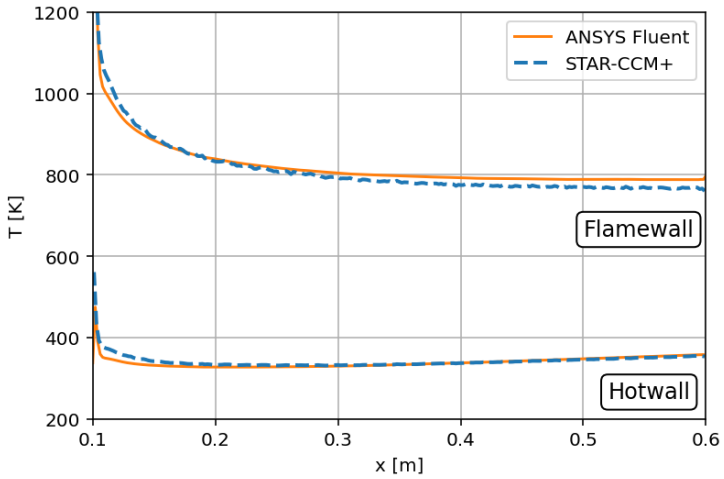

Figure 4.1: Wall temperatures at the flame- and hotwall.

4.1.2 Species Mass Fraction

In Table 4.1 the averaged species mass fractions are shown at the outlet of the flame. The table only shows the mass fraction of the species that were used at the inlet of the flame and there are more species in the domain that has formed due to the reactions. The simulations show similar results for the dominant species such as H2O, CO2 and CO. One notable difference is that STAR-CCM+ have less OH and O2 at the outlet, which is also reflected as an increase in the mass fractions of both O and H as the molecules break apart.

Table 4.1: Species mass fraction at the outlet compared to the inlet.

Species	Inlet	Ansys outlet	Star outlet	Difference
H2O	0.41234	0.431	0.431	$< 0.1\%$
CO2	0.29079	0.345	0.345	$< 0.1\%$
CO	0.20289	0.168	0.168	$< 0.1\%$
OH	0.04653	0.0314	0.0232	26%
O2	0.03437	0.0165	0.0229	27%
H2	0.00618	$5.6 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$	$< 0.1\%$
O	0.00597	$3.25 \cdot 10^{-3}$	$3.28 \cdot 10^{-3}$	0.9%
H	0.00077	$6.51 \cdot 10^{-4}$	$6.56 \cdot 10^{-4}$	0.8%
HO2	0.00016	$1.27 \cdot 10^{-5}$	$1.26 \cdot 10^{-5}$	0.8%

4.1.3 Boundary Layer

In figure 4.2, the y^+ values are shown for the flamewall, hotwall and the extended channel wall. The y^+ values on each of the walls follow the general trend of the other software, but the flamewall has an offset between the CFD codes. However, all the y^+ values are below one ensuring that the mesh is fine enough to capture the viscous sublayer.

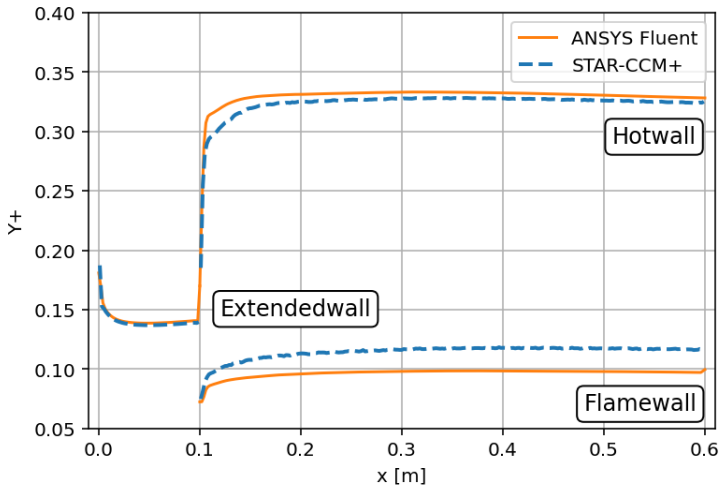

Figure 4.2: y^+ value of the flamewall, hotwall and the extended wall along the x-direction.

4.1.4 Convergence

In table 4.2 it can be seen that the CFD codes converged at similar amount of iterations but that Fluent had a worse heat imbalance for the flame at 0.21% while in STAR-CCM+ it was below 0.1%. STAR-CCM+ did however have a higher channel heat imbalance at 0.14%. The rest of the imbalances were all below 0.1% along with normalized residuals under 1 for both CFD codes. The simulation was stopped when all convergence parameters (imbalances, residuals and engineering quantities) had leveled out at a sufficiently low value. However, the heat imbalance for the flame in Fluent was still decreasing when the simulation was stopped. This was done because the wrong expression was used to measure the heat imbalance showing that the simulation had converged at 3%. The expression did not capture the released heat from the reactions, and therefore could not converge towards zero. Later on, the "HeatTransferRate" function in Fluent was used, revealing that the simulation was stopped at a heat imbalance of 0.21%.

Table 4.2: Convergence measurements.

Parameter	**Fluent**	**STAR-CCM+**
Iterations	10000	9900
Mass imbalance flame	$4.0 \cdot 10^{-8}$	$4.8 \cdot 10^{-6}$
Mass imbalance channel	$1.4 \cdot 10^{-6}$	$2.0 \cdot 10^{-5}$
Heat imbalance flame	$2.1 \cdot 10^{-3}$	$6.7 \cdot 10^{-4}$
Heat imbalance channel	$4.6 \cdot 10^{-6}$	$1.5 \cdot 10^{-3}$

When the heat imbalance of the flame in Fluent was believed to be at 3%, a number of simulation methods were tried to decrease it. Firstly, the simulation was ran for a longer time, with 50000 iterations in total, resulting in a lower heat imbalance at 0.03%. The CFL number was also altered through the fluid time scale factor, which showed that too low a CFL number led to unstable behavior in the simulation. An increase in CFL, however, decreased the heat imbalance of the flame, and after 30000 iterations in total the heat imbalance had reached $3.9 \cdot 10^{-6}$. Neither of these changes had a significant impact on the thermal loads of the walls such as the temperature profiles.

In STAR-CCM+, the species residuals were initially large due to that the normalization value was chosen by the code within the first 5 iterations. At the start of the simulation the reactions were disabled leading to small species residuals. However, later when the reactions were enabled the normalization was done with the small values from the start of the simulation leading to inflated residuals. Including the start of the reactions in the normalization led to a better portrayal of the normalized residuals.

4.1.5 Turbulent Inlet Conditions

To explain the differences between the simulations the turbulence entering the domain through the inlets was investigated. In figure 4.3 and 4.4 turbulent parameters such as turbulent viscosity, turbulent kinetic energy and specific rate of dissipation can be seen. The figures show that the turbulent quantities were not the same at the inlets between the CFD codes, especially the turbulent viscosity in the flame, even though the same turbulent intensity and viscosity ratio had been specified in the CFD codes. In figure 4.3, the turbulent quantities for Star cuts off near the symmetry boundary condition in the channel and at the same location in Fluent the value changes drastically. The change was limited to the cell furthest away from the hotwall and the difference between the CFD codes arose due to differences in how the symmetry boundary condition is treated. This effect can also be seen at the flame for k, but it was less pronounced.

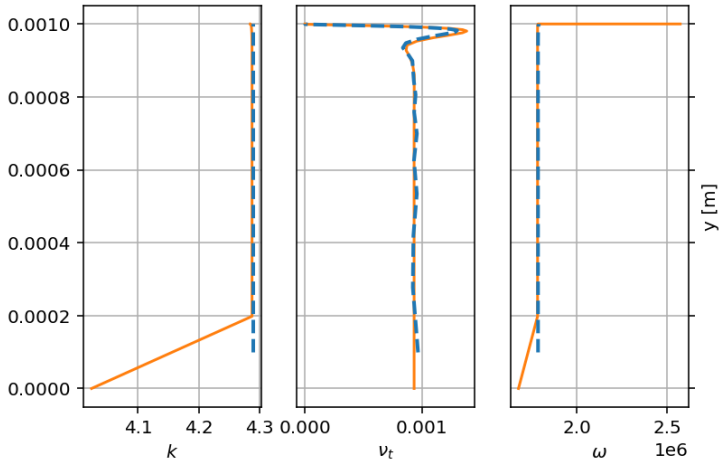

Figure 4.3: Turbulent quantities for the channel inlet. The orange line is the Fluent inlet and the blue dashed line is the STAR-CCM+ inlet.

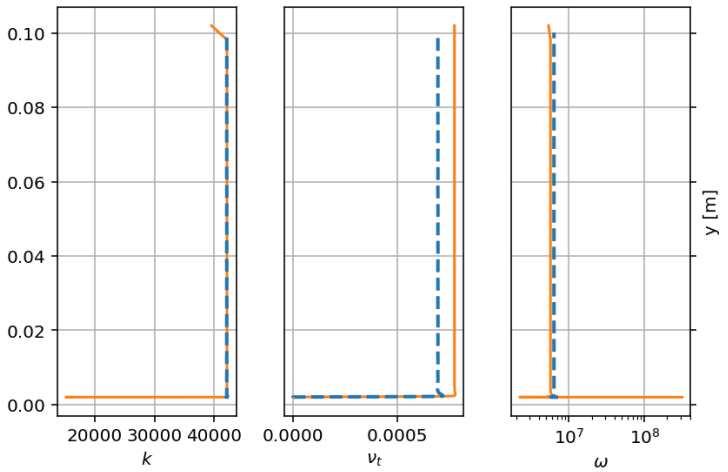

Figure 4.4: Turbulent quantities for the flame inlet. The orange line is the Fluent inlet and the blue dashed line is the STAR-CCM+ inlet.

To address the difference, the simulation in STAR-CCM+ was ran again with profiles of specific rate of dissipation and turbulent kinetic energy from the Fluent simulation at the inlets. This ensured that both CFD programs had identical inlet conditions. The result from the simulation however showed little to no impact on the wall temperature profiles indicating that the turbulent boundary conditions were not the source of the discrepancy.

4.2 Representative Cooling Channel

The following section covers the results of the second case with results from both Fluent and STAR-CCM+ along with the experimental results from [14] for verification. The focus of this section was to compare both the accuracy and the efficiency of the CFD codes.

4.2.1 Cold Case

Iteration upon the wall roughness was done within both CFD codes following the procedure of [14] and the conditions matching that of the "0A" experiment. The resulting ESGR for Fluent and Star were 12.08 μm and 7.083 μm respectively, to achieve a pressure drop of 3.2% in both codes.

4.2.2 Wall Temperature

Figure 4.5 shows a plot along the x-direction showing the channel wall temperatures at the locations of the sensors in the experiments by [14]. The results from the Fluent simulation showed better agreement with the experiments but the temperatures of STAR-CCM+ are essentially offset from those of Fluent while following the same behavior.

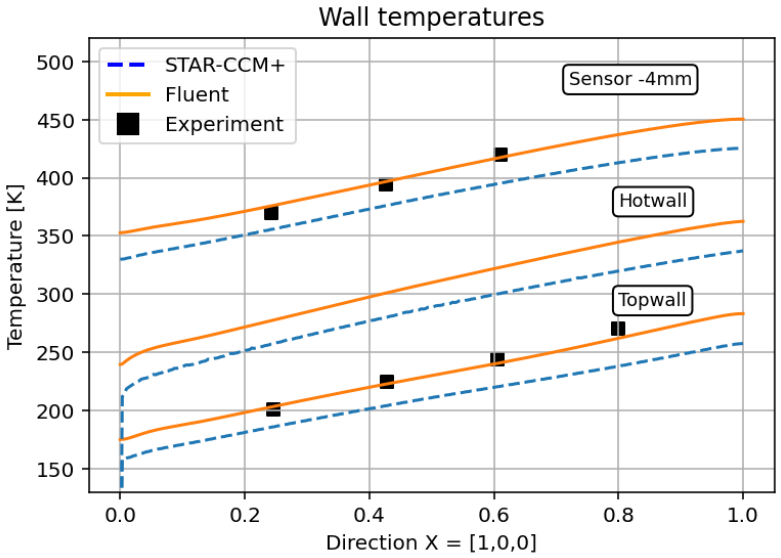

Figure 4.5: Wall temperatures from the experiment and the two CFD codes.

Adjusting the turbulent Prandtl number of the fluid within STAR-CCM+ to fit those of Fluent resulted in the temperature profiles seen in figure 4.6. The agreement between STAR-CCM+, the experiments and Fluent increased across the whole domain as the temperature offset diminished.

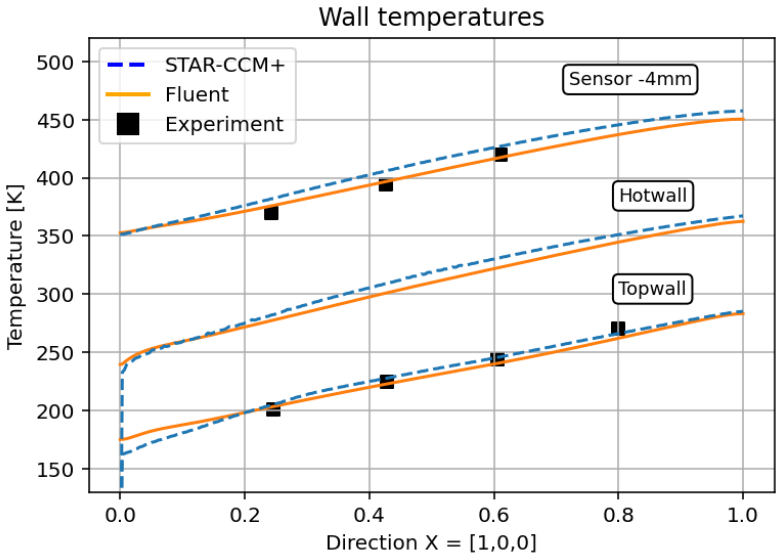

Figure 4.6: Wall temperatures using the Fluent turbulent Prandtl number in both codes.

4.2.3 Heat Transfer

Table 4.3 shows the heat transfer rate from the channel wall to the coolant flow along the x-direction integrated over each surface. Figure 4.7 shows the average wall y^+ of the channel bottom wall along the x-direction, where a near perfect match between the two CFD codes can be seen.

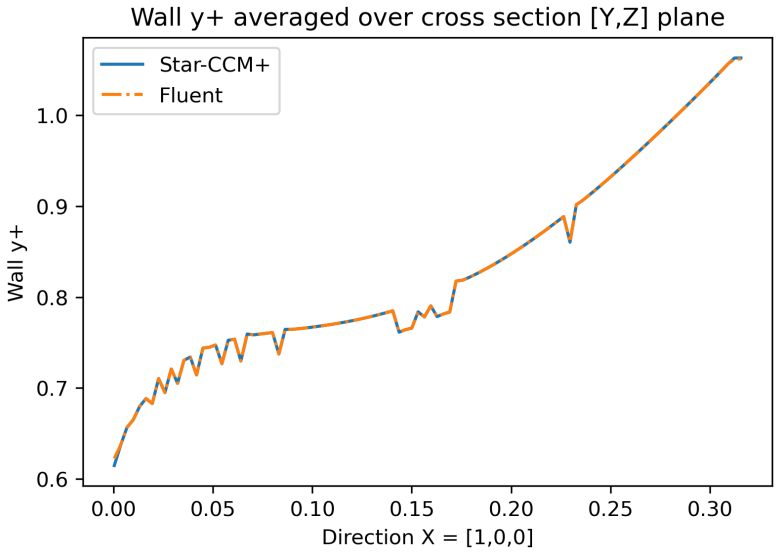

Figure 4.7: Wall y^+ comparison between the two CFD codes.

Table 4.3: Wall heat transfer rate comparison.

Location	Heat flux Fluent [W]	Heat flux STAR-CCM+ [W]
Top Wall	860.2	755.8
Side Walls	8327.8	8402.5
Bottom Wall	1467.3	1542.0

4.2.4 Convergence

Table 4.4 shows the imbalances at the end of the final simulation of the case for each code, they converged when they had leveled out below 0.1% in error. Apart from imbalances the pressure loss and heat pickup were considered converged when leveled out and differ less than 0.1% from their respective mean values. STAR-CCM+ required fewer iterations to converge but the simulation time was approximately the same.

Table 4.4: Amount of iterations and imbalance for each software.

Values	Fluent	STAR-CCM+
Iterations	$30,000$	$25,000$
Mass imbalance	$3.7 \cdot 10^{-8}$	$4.0 \cdot 10^{-11}$
Heat imbalance	$3.8 \cdot 10^{-4}$	$2.3 \cdot 10^{-4}$

4.3 Rocket Nozzle

The following section contains the results from Fluent and STAR-CCM+ regarding the third case. Apart from what has been shown for the other sections a performance measure comparison between Star, Fluent and a reference was included to evaluate the quality of the results. Worth noting is that running the case with the final channel geometry while having a mass flow of 0.149 kg/s did not cause an occurrence of HTD, running at lower mass flows was possible but did cause unrealistic wall temperatures and operating conditions for the nozzle and channel flows.

4.3.1 Wall Temperature

Below in figure 4.8 the wall temperatures for the NE flamewall and the channel hotwall are shown. The temperature profiles were taken from the middle of the slice ($Z = 0$) and follows the nozzle wall contour. The difference between the CFD codes were on average 3.27 K and 3.14 K for the flame and hotwall respectively.

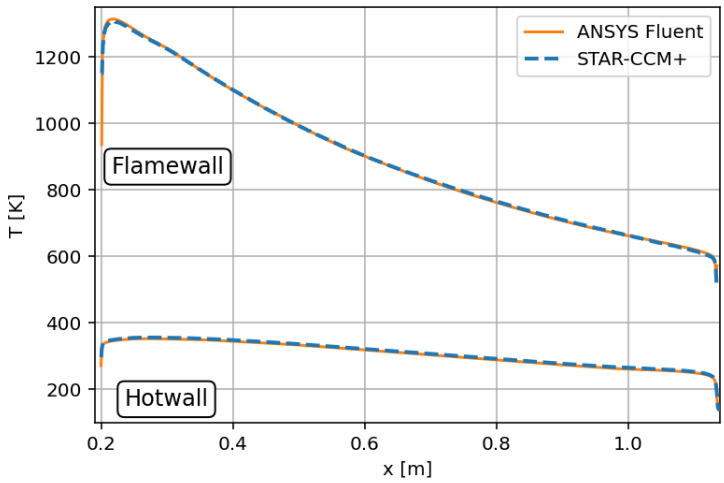

Figure 4.8: Wall temperature profiles along the NE contour.

The difference in percentage are shown in figure 4.9 along the NE, the largest difference can be seen at the start of the hotwall. The points are every 10th data point and the data from Star needed to be interpolated to get the same x-coordinates as Fluent, due to them storing temperature values in a node based and cell based manner respectively.

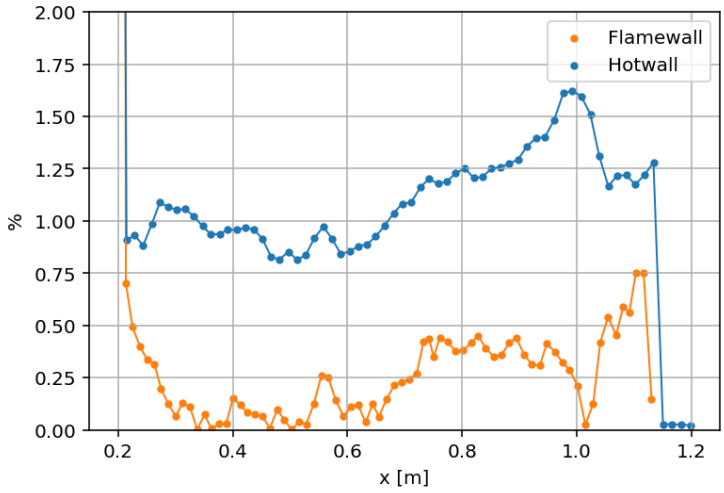

Figure 4.9: The difference of the wall temperatures between the CFD codes in percentage.

The simulation was also run with the original Prandtl and Schmidt numbers in STAR-CCM+ which gave the results seen in figure 4.10, with a larger temperature difference at the walls.

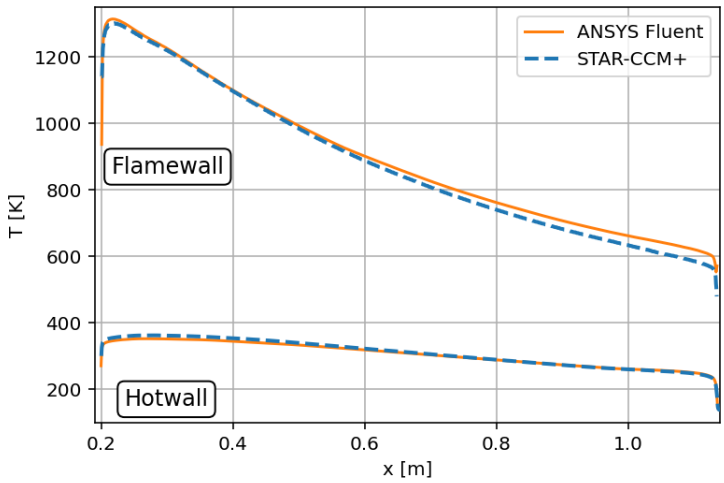

Figure 4.10: Case 3 wall temperatures, with Star's original Prandtl and Schmidt numbers.

4.3.2 Boundary Layer

The y^+ values throughout the report were calculated using built in functions in each software. However, in figure 4.11 it can be seen that the functions do not align at the flamewall, where the y^+ value in Star was almost double that of Fluent while showing a good agreement in wall temperatures. To control that the built in functions worked as intended the y^+ values were calculated manually, resulting in the profiles in figure 4.12 showing a smaller difference for the flamewall.

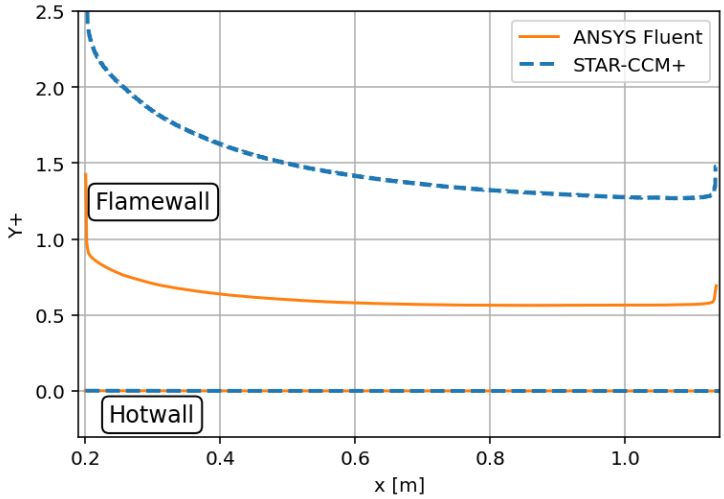

Figure 4.11: y^+ calculated with built in functions.

The y^+ values presented in Figure 4.12 show that the boundary layer was sufficiently resolved for the hotwall with y^+ below one for the whole wall. The majority of the flamewall was below a y^+ of one, except the first 0.1 m. In the profiles a clear offset was also seen between the CFD codes with Star displaying a higher y^+ at the flamewall.

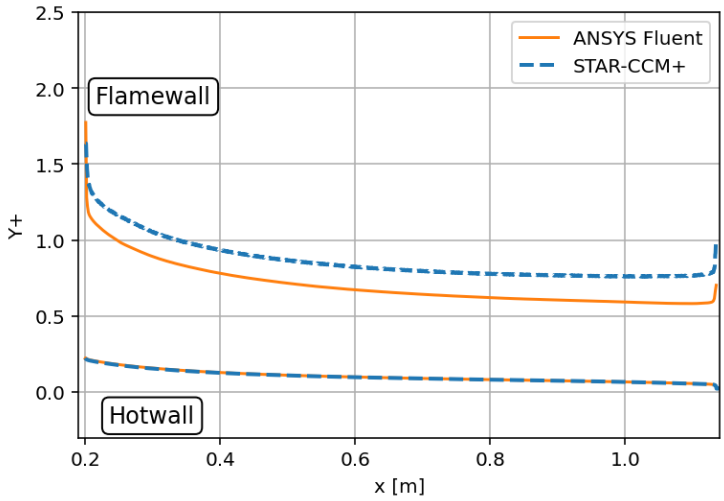

Figure 4.12: y^+ calculated according to $y^+ = \frac{yu_\tau}{\nu}$.

4.3.3 Mach Contour

Figure 4.13 shows a contour plot comparing the mach number of the Fluent and STAR-CCM+ simulations using the updated settings for a close match in results. The figure is split horizontally where the top half displays the results of Fluent and the bottom that of STAR-CCM+. The mach contours are shown to be very similar with smaller differences towards and connecting to the centrum line of the nozzle flow.

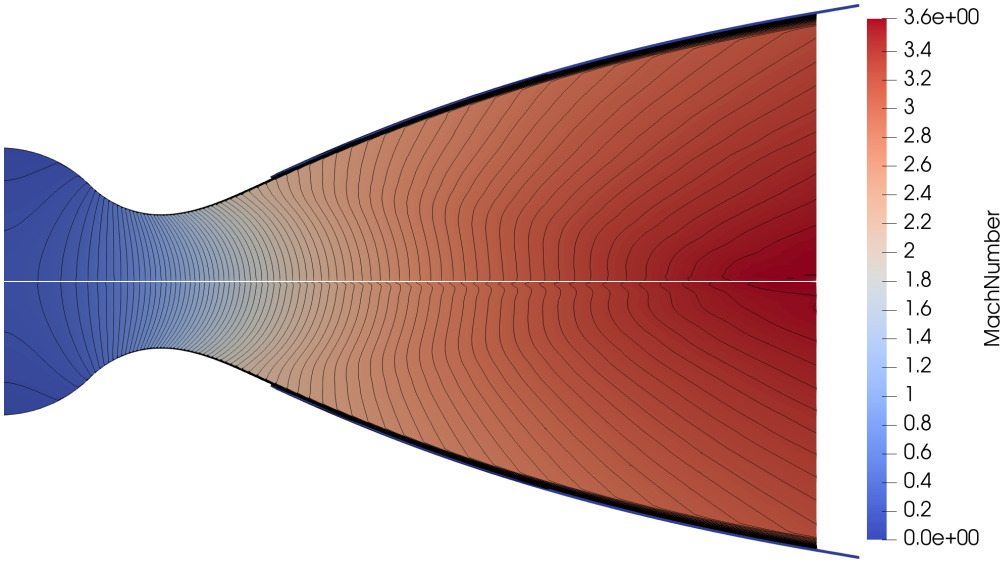

Figure 4.13: Mach number contour, showing Fluent at the top part and STAR-CCM+ at the bottom, separated by the white line.

4.3.4 Convergence

Table 4.5 shows the number of iterations and resulting imbalance values from the simulations within both codes. The simulations were stopped when the imbalances and engineering properties, such as channel pressure loss and average flame wall temperature, had leveled out. The convergence criterion for the imbalances were that they should be below 0.1% for all of the regions.

Table 4.5: Convergence measurements.

Parameter	Fluent	STAR-CCM+
Iterations	33700	35000
Mass imbalance flame	$1.29 \cdot 10^{-6}$	$1.0 \cdot 10^{-5}$
Mass imbalance channel	$4.75 \cdot 10^{-7}$	$1.5 \cdot 10^{-4}$
Heat imbalance flame	$3.76 \cdot 10^{-4}$	$3.5 \cdot 10^{-4}$
Heat imbalance channel	$2.29 \cdot 10^{-6}$	$2.5 \cdot 10^{-5}$

4.3.5 Nozzle Performance

To compare the overall mesh quality the performance measures of the nozzle were compared with the parabolic TIC nozzle from [15] with non equilibrium reactions. The reference was simulated using an axis-symmetric mesh with 500 K on the flame wall, additionally the reference used another reaction scheme; hence an apparent difference was expected. The comparison is shown in table 4.6, where Star 1 refers to the STAR-CCM+ simulation using standard settings and Star 2 refers to the simulation with Fluent's Prandtl number of 0.85. The comparison of the CFD codes reveals similar results with minor differences. The differences become slightly more pronounced as the settings are changed in Star (Star 2), but are minor compared to the improvement in the wall temperature profiles.

Table 4.6: Performance comparison with reference nozzle.

Performance	Star 1	Star 2	Fluent	Reference
F_N	-0.19%	-0.21%	-0.16%	809.3 [kN]
I_{sp}	-3.7%	-3.7%	-3.8%	3487 [m/s]
$\dot{m}_{flame}$	$+3.9\%$	$+3.8\%$	3.9%	231.8 [kg/s]

Table 4.7 shows additional system level values used to compare the codes against each other, such as the pressure drop and heat pick up of the coolant, along with the heat transfer through the nozzle extension.

Table 4.7: Code comparison.

Performance	Star 1	Star 2	Fluent
ΔP	7.06%	7.11%	7.16%
ΔT	$47.26K$	$48.80K$	$49.63K$
Q_{Flame}	$48749.37W$	$50363.02W$	$51120.14W$

5
Discussion

This section contains the discussions regarding what was presented in the results chapter along with an evaluation of the user experience and time consumption of simulating in each separate CFD code. This section is split up in two main parts, the general discussion relating to the assembled results of all cases and a second part with case specific discussions.

The first case was handled like a sandbox for investigating the mismatch in the results between STAR-CCM+ and ANSYS Fluent. As this project had a clear focus on the calculation of thermal loads, matching wall temperatures and heat fluxes were a priority. An important setting that needed to be enabled in Fluent was "viscous heating", that accounts for the dissipation of turbulent kinetic energy to heat. The setting had a significant impact on the wall temperature, shifting its profile up to 200 K in the first case. The main trend regarding the differences between Star and Fluent was an offset in the wall temperatures while the temperature gradients in the wall remained similar, indicating the fluid conductivities as a point of interest. The laminar contribution to the fluid conductivity was described by kinetic theory in both CFD programs, but the turbulent contribution was calculated using equation 2.25 that assumes a turbulent Prandtl number (P_{rt}). Since the programs assume different turbulent Prandtl numbers this part was investigated. The results show that P_{rt} has a large impact on the fluid conductivity and thus the heat flux, since it directly scales the turbulent contribution to the fluid conductivity. Matching P_{rt} between the programs gave a significantly better alignment in terms of wall temperature. Other differences, though with smaller impact, were identified to be the "Compressibility correction" that was enabled in Star and the turbulent Schmidt number of the flame. The compressibility correction was disabled as it can worsen the resolution of the boundary layers and the Schmidt number was matched between the programs to maintain consistency even though it had a smaller impact than the turbulent Prandtl number.

During the project Fluent was used in "Batch mode" while Star was used in "Server mode", which was not available for Fluent at GKN Aerospace during the project. Server mode allows the user to interact with the simulation setup during the simulation, as if running it on a local machine. In terms of user experience server mode has generally been appreciated by the authors as it allowed for live changes in the solver settings thus maintaining higher convergence rate and stability. Star also allowed for the deactivation of regions which can accelerate convergence by briefly decoupling the flows from each other. It is worth noting that Fluent allows for the

deactivation of regions but it is, in general, a cumbersome process, where report expressions would error and not allow the user to run the simulation until they were resolved, it was therefore concluded to not be recommended. The method of deactivating regions was also well suited for server mode as it enabled the user to make live changes. Another benefit with Star was the ease of replacing imported meshes and interpolating the results over to the new mesh which accelerated the mesh study. Overall it was found more comfortable to work within STAR-CCM+ compared to ANSYS Fluent.

ANSYS Fluent, in contrast to STAR-CCM+, supports the rgp format files for simulations involving supercritical and cryogenic fluids. In this project the files were imported into Star by converting them into csv files formulated according to Star standards by using a python script. The python script was general enough to be used for different materials, pressure and temperature ranges etc., meaning future translation of rgp files should not pose a problem. The process of converting and importing the tables was an additional sequence in setting up the simulation that was easier in Fluent, and could potentially introduce errors if done incorrectly. However, the developed script worked well as proven in case 2 and produces results with the same quality as a rgp file if implemented correctly. A similar trend has been seen in STAR-CCM+ as a whole with numerous settings and functions that needed to be changed in order to match the results of Fluent and the experimental data of the second case. A template sim file was constructed, for the third case, to reduce the workload of setting up future simulations in Star, where the user only needs to import the mesh and assign the boundary condition before running. However, it is important to note that STAR-CCM+ has been observed to be more sensitive to HTD for coarser meshes, and that Star needed a finer mesh to match Fluent. This is especially true at higher pressure levels like in case 3 (265 *bar* in the coolant flow), where the boundary layer became thinner.

The results from the second case indicates that a turbulent Prandtl number of 0.85 produces a realistic result for the flow of the coolant compared to the experiment. This conclusion can not be drawn about the flame since the flamewall temperature of the other cases have not been compared to any experimental results. A relevant question could be if both of the flows require the same Prandtl number to be modeled accurately, if not then Star would have a clear advantage since it allows for different P_{rt} in each fluid while Fluent does not.

Two important settings throughout the project have been the CFL number and the time scale factor used in Star and Fluent respectively. Both of the parameters are used to change the pseudo-time step of the solver and therefore serve the same purpose. The general trend that has been seen in Fluent when using the default time scale factor of 1 has been slow convergence for the flame heat imbalance. However, it has been seen in case 1 that an increase of the time scale factor can speed up the convergence. The simulations in Star were run with server mode allowing for live changes in the solver settings and a manually variable CFL number. The trend seen in Star aligns with that of Fluent, where an increase in CFL will increase the

convergence rate of the solver. One difference between the CFD codes was that in Star the value of the CFL number was set, which in turn meant that the pseudo-time step was set. In Fluent only a factor to scale the pseudo-time step was used resulting in less frequent changes as the solver would automatically determine the pseudo-time step.

5.1 Case 1: Flat Plate

In case 1 it was shown that the two CFD codes were able to produce similar results in almost the same amount of iterations which stood as an argument for continuing to the following cases with both codes and the developed method. It also gave a forecast of where differences might still appear. Considering figure 4.1 the largest differences can be seen at the location where heat transfer starts to occur between the regions which is made more apparent in figure 4.2. As heat starts to pass into the fluid the boundary layer starts to thicken and a thermal boundary layer starts to form, which might be a topic for further work to investigate as this was the source of the largest remaining error between codes and persisted in the following cases. See figure 4.6 for the hotwall curves, keep in mind that the viscous boundary layer of the coolant flow was initiated for all three cases. The cause of this was not investigated as the match in results remained close enough for the purpose of this project.

Additionally, it was shown in case 1 that the turbulence at the inlet boundary conditions were not equal between the CFD programs, especially for the turbulent viscosity of the flame outside the boundary layer. This was shown for a turbulent intensity and for a viscosity ratio boundary. According to [2] this is also true for turbulent lengths scale where STAR-CCM+ uses the Wilcox approach and Fluent uses a mixing-length derivation to arrive at a different definitions. This example shows that the implementation of turbulence models can vary between each software and there might be more hidden differences accounting for the slight disparity in the results. As the first case had a simple and straight geometry the dominating source of turbulence was that of the inlet boundary which might have caused the difference in the results to propagate beyond the proximity of the inlet, this effect was not as pronounced in the last case which may be due to the bend of the geometry and sources of turbulence beyond the inlet boundary.

5.1.1 Heat Transfer Deterioration

In the beginning of the project Star could not converge and the wall temperature profiles had lumps on them. Later on in the project it was discovered that the lumps in Star were a result of HTD, which was avoided in Fluent because the viscous heating setting was wrongly inactive, resulting in a significantly lower wall temperature. The result of the HTD in Star was poor convergence with high energy imbalances and unstable behavior of the wall temperature profiles that moved in a wavelike pattern and required many iterations to stabilize. When viscous heating was enabled in Fluent the solver had similar problems to Star but manged to converge faster. The results also clearly highlight the importance of including viscous heating

in the flame as it made the temperature profiles approximately 200 K hotter. Due to the convergence difficulties associated with HTD it was avoided throughout this project, mainly by increasing the mass flow and thus keeping the coolant flow below its pseudo-critical temperature. This did result in non-realistic operating conditions in the third case, but as the design and optimization of the cooling channel geometry was outside the bounds of this project the increased mass flow was retained.

5.2 Case 2: Representative Cooling Channel

During this case neither of the CFD codes had problems converging although Fluent required a slightly higher number of iterations to do so but in roughly the same amount of time.

The offset in wall temperatures between the solvers had previously been shown to arise from a difference in the settings for the turbulence model. Changing the turbulent settings in both codes to fit those of the first case resulted in stronger agreement between the CFD codes and the experimental data. The final simulation displayed a close match between the codes as seen in figure 4.6. It had good agreement with the experimental values of [14], which verified the accuracy of the method to simulate the coolant.

A precise match in y^+ at the bottom wall of the coolant display a similar buildup of the boundary layers when comparing the codes, yet despite this, the heat transfer showed some differences. Both codes showed the same trend, with the side walls being the boundaries where the most heat transfer occurred. The sum heat flux between the codes had a 0.8% difference in effect going through the boundaries, this indicates a lasting discrepancy that could be investigated and reduced in future work but was considered small enough to neglect in this project.

5.3 Case 3: Rocket Nozzle

The performance measure comparison in the results for the third case could have been improved by running a simulation using identical boundary conditions to that of [15] as it would be a more telling comparison and indication of the mesh quality. The results from this project were attained through three-dimensional periodic simulations with cooling which differ from the reference case. For a fair comparison the simulations should have been performed without cooling, then compared with the reference. For STAR-CCM+ this would have required multiple simulations as both the standard and improved settings would need testing. The comparison was still included to check the order of magnitude of the results which proved to be same as those from [15] thus proving the validity of the methods used for simulating the flame to some degree.

Another implication of the method accuracy is that the two CFD codes arrived at nearly identical solutions when the settings of the turbulence model were adjusted.

As of the end of the project Fluent had a slightly lower rate of convergence compared to Star, though this might be the result of differently powerful cores used to perform the simulations as the number of iterations to reach convergence was of the same order of magnitude. Lastly, the difference in wall y^+ remained, further strengthening the argument of STAR-CCM+ needing a finer mesh. Figure 4.13 showed a difference connecting to the center of the nozzle flow, this could be a result of the high aspect ratio cells connecting to the rotational axis, due to the thin 1° angular segment of the nozzle. This can be seen as the iso-lines show a gradient over the centrum line, indicating boundary effects as the cause. There are strategies to minimize this error, such as excluding the middle and replacing it with a free stream boundary condition, but these were not tested in this project as the main focus lied on the thermal loads on the nozzle walls which seem to have been undisturbed by these errors. The nozzle geometry was sourced from [15] which had a similar problem with nozzles of inferior design and the effect of that on the formation of the boundary layer. With that in mind the error could be a result of an insufficient resolution of the boundary layer which would explain why the error was more pronounced in Star as it showed a higher flamewall y^+ value compared to Fluent. If this was the case, the error would be mitigated with a refinement of the flame mesh close to the nozzle wall.

The time to simulate the third case was around 4320 or 5520 core hours for Star-CCM+ and Fluent respectively, indicating that Fluent takes slightly longer to converge. The comparison was made with the same mesh and considering that Star needs a finer mesh to achieve the same y^+ as Fluent, this difference would decrease. It should also be noted that Fluent had some warning messages during the whole simulation of unrealistically high and low temperatures that could have slowed down the simulation. These errors were limited to a couple of cells at the coolant outlet and should not have an impact on the overall quality of the simulation. The errors were a result of some extremely thin cells in the boundary layer, leading to a small amount of cells being skewed. To avoid this problem a lower pressure could have been used in the cooling channel to increase the thickness of the boundary layer, so that the mesh could be coarser.

When the mesh refinements were compared in the third case, it was observed that the codes used different methods to calculate the wall $y+$. The difference was most notable at the flamewall where Star produced a $y+$ that was almost double that of Fluent. In order to make a fair comparison the wall y^+ was instead calculated using user-specified field functions which gave a much closer match between the codes, however Star still had a higher value than Fluent on the flamewall. Neither of the codes' documentation was forthcoming with what implementation of the wall y^+ was used, therefore it is recommended to calculate certain properties manually in order to confirm their validity. During the project multiple examples of lacking documentation of the implementation of models were noticed, among them the positioning of probes and the definition of turbulence at the inlets. These differences are likely to account for an unknown quantity of error in the results for both of the CFD codes observed throughout the project.

6
Conclusion

This chapter goes through the conclusions drawn about the methods used in the project along with the user experience and time consumption for each CFD code.

Both CFD codes were able to perform a holistic simulation of a rocket engine nozzle including all relevant physics. STAR-CCM+ was slightly faster compared to Fluent. The manual input for both programs was low though higher for Star when activating and deactivating the regions for the iteration method, this could however be minimized using a macro within STAR-CCM+. The results were accurate when comparing to the previous work of [15] and [14] but they have yet to be compared against GKN Aerospace internal methods. The final conclusion is that the simulations are viable though heavy and time consuming when performed in fully three-dimensional meshes. To accelerate the process a combination between an axisymmetric mesh of the flame with a three-dimensional mesh for the channel and coolant could be utilized, interpolating the values from boundary to boundary and using the mean temperature of the channel flame wall to set on the boundary of the flame, although the inclusion of an axisymmetric and 3D mesh within the same simulation might not be possible in the two CFD codes. But as for the methods in this project they have been proven to be reliable and accurate.

The two CFD codes feature both strengths and weaknesses. The iteration method used in Star required a larger amount of manual input, this method was used to save time but a run using the same method as Fluent was never tested due to its higher time consumption. Star cases took longer to set up due to the abundance of settings needing to be changed, this problem was partially solved as a template file was set up where meshes could be directly imported and retained all the necessary settings. Using the template file equals out the time to set up a case. With all above in mind, both codes were easy to use and accurate, but in this project Star was found to be superior in the first regard and required less time to converge, thus being the code recommended for this application based on the experience of the authors.

As stated in the introduction the aim of this project was to *"investigate the inclusion of all the relevant phenomena and regimes in one single simulation including the cryogenic coolant, the chemically reacting flame and the heat transfer between the regions. The goal is to eliminate the need of setting up multiple simulations and the need to manually iterate between them"*. By the end of the project the aim was achieved for both CFD tools, with a developed simulation method that was verified against previous work.

Bibliography

[1] ANSYS Inc. *ANSYS FLUENT 12.0 Theory Guide*, 2009. Accessed: 2025-03-24.

[2] CFD Online. *Turbulence length scale*, 2024. Accessed: 2025-05-13.

[3] Lars Davidson. *fluid mechanics, turbulent flow and turbulence modeling*. 2025. [Accessed 18-02-2025].

[4] L. Denies. Regenerative cooling analysis of oxygen/methane rocket engines. Master's thesis, Delft, 2015.

[5] J. H. Ferziger and M. Peri´c. *Computational Methods for Fluid Dynamics*. Springer-Verlag, 2002.

[6] N. Hall. Nozzle design - converging/diverging (cd) nozzle — grc.nasa.gov. `https://www.grc.nasa.gov/www/k-12/airplane/nozzled.html`.

[7] N. Hall. Rocket thrust equation — grc.nasa.gov. `https://www.grc.nasa.gov/www/k-12/airplane/rockth.html`.

[8] ANSYS Inc. *ANSYS FLUENT 12.0/12.1 Documentation*, 2009. Accessed: 2025-04-14.

[9] ANSYS Inc. *ANSYS FLUENT 2024 R1 User's Guide*, 2024. Accessed: 2025-04-14.

[10] William M. Kays. Turbulent prandtl number—where are we? *Journal of Heat Transfer*, 116(2):284–295, May 1994.

[11] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, August 1994.

[12] NASA. Specific impulse, 2025. Accessed: 2025-03-28.

[13] Marco Pizzarelli, Francesco Nasuti, Raffaele Votta, and Francesco Battista. Validation of conjugate heat transfer model for rocket cooling with supercritical methane. *Journal of Propulsion and Power*, 32:1–8, 03 2016.

[14] Daniele Ricci, Pasquale Natale, Francesco Battista, Emanuela D'Aversa, Rocco Pellegrini, Hiroya Asakawa, and hideo Negishi. Investigation on the transcritical behaviour of methane and numerical rebuilding activities in the framework of asi/jaxa cooperation project. 05 2016.

[15] Tamas Bykerk Sebastian Karl and Mariasole Laureti. Design of a truncated ideal nozzle for a re-usable first stage launcher. In *HiSST: 3rd International Conference on High-Speed Vehicle Science Technology*, 2024.

[16] Siemens Digital Industries Software. *STAR-CCM+ User Guide*, 2025. Accessed: 2025-03-27.

[17] George P. Sutton and Oscar Biblarz. *Rocket Propulsion Elements*. John Wiley and Sons, Inc., Hoboken, New Jersey, 2017.

[18] Victor Zhukov. Kinetic model of alkane oxidation at high pressure from methane to n-heptane. *Combustion Theory and Modelling*, Vol. 13:427–442, 07 2009.

[19] Ying Zou, Xingwang Zhao, and Qingyan Chen. Comparison of star-ccm+ and ansys fluent for simulating indoor airflows. *Building Simulation*, 11(1):165–174, 2018.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY