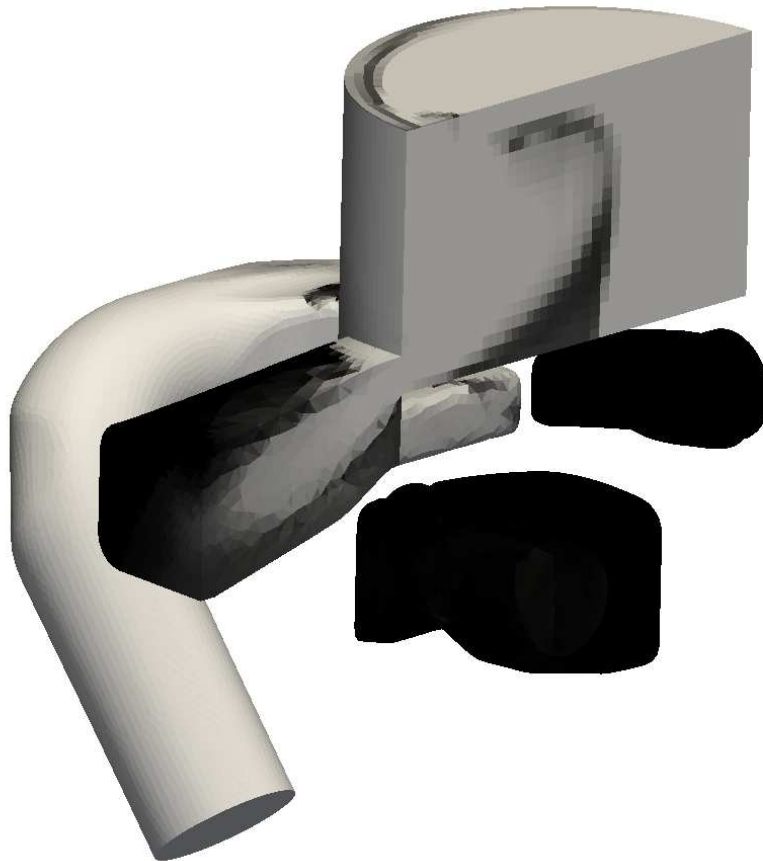


CHALMERS



Stratified scavenging computations in two-stroke engines using OpenFOAM

Master's Thesis in Automotive Engineering

MIKAEL JÖNSSON

Department of Applied Mechanics
Division of Fluid Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010
Master's Thesis 2010:24

MASTER'S THESIS 2010:24

Stratified scavenging computations in two-stroke engines using
OpenFOAM

Master's Thesis in Automotive Engineering
MIKAEL JÖNSSON

Department of Applied Mechanics
Division of Fluid Dynamics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2010

Stratified scavenging computations in two-stroke engines using OpenFOAM
MIKAEL JÖNSSON

©MIKAEL JÖNSSON, 2010

Master's Thesis 2010:24
ISSN 1652-8557
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:

A cross-section of the H576X engine showing the scavenging of the air fuel mixture.

Chalmers Reproservice
Göteborg, Sweden 2010

Stratified scavenging computations in two-stroke engines using OpenFOAM
Master's Thesis in Automotive Engineering
MIKAEL JÖNSSON
Department of Applied Mechanics
Division of Fluid Dynamics
Chalmers University of Technology

Abstract

This report presents the methodology and the results of simulations of the flow in a stratified charged two-stroke engine. The focus is put on the process and performance of the scavenging. To simulate the flow through the engine the finite volume method in three dimensions is used. The OpenFOAM-1.5-dev computational fluid dynamics software is used to perform the calculations. The performance and possibilities that OpenFOAM-1.5-dev gives for this kind of applications is evaluated.

Keywords: Two-stroke engine, OpenFOAM, Mesh operations, Topology changes, Trapping efficiency

Contents

Abstract	I
Contents	III
Nomenclature	V
Preface	VII
Acknowledgements	VIII
1 Introduction	1
1.1 Background	1
1.2 Aim and purpose	1
1.3 Theory	2
1.3.1 The two-stroke engine	2
1.3.2 Conventional two-stroke engine	5
1.3.3 Stratified charged two-stroke engine	6
1.3.4 The governing equations	9
1.3.5 The RANS-equations and turbulence modelling	9
1.3.6 The finite volume method	12
1.3.7 Fluids	14
1.4 Approach	15
1.5 Limitations	16
2 Method	17
2.1 Engine geometry	17
2.1.1 Engine geometry modifications	18
2.2 Mesh generation	19
2.2.1 The H576 engine geometry	20
2.2.2 The block geometry	21
2.3 Dynamic mesh with topology changes	21
2.3.1 Liner and ports	22
2.3.2 Deforming mesh	22
2.3.3 Layer addition and removal	23
2.3.4 Sliding interfaces	24
2.3.5 Mesh quality	25
2.4 Indata from 1-D software	27
2.5 Discretisation schemes	28
2.5.1 Time derivatives	28
2.5.2 Gradient terms	28
2.5.3 Divergence terms	28
2.5.4 Laplacian terms	29
2.6 Boundary conditions	29
2.6.1 Walls	29
2.6.2 Inlets and outlets	30
2.7 Time step and convergence criterion	32
2.8 Setup for calculations in parallel	33
2.8.1 Layer addition and removal in parallel	33
2.8.2 Sliding interfaces in parallel	34

2.8.3	Domain decomposition	34
2.9	Assumptions and approximations	36
2.9.1	Fluids and combustion	36
2.9.2	Initial field values	36
3	Results	38
3.1	Scavenging and trapping efficiency	38
3.2	Validation	45
3.2.1	Commercial 1-D software	45
3.2.2	Turbulence modelling	46
3.2.3	Commercial 3-D software	49
4	Conclusions	51
5	Discussion and recommendations	52

Nomenclature

\dot{m}	Mass flow
$\eta_{tr,fuel}$	Trapping efficiency, fuel
η_{tr}	Trapping efficiency, air fuel mixture
\mathbf{U}	Velocity vector
\mathbf{u}	Velocity vector
μ	Dynamic viscosity
ν	Kinematic viscosity
ρ	Density
σ	Skewness
θ	Crank angle degrees
ε	Dissipation of turbulent kinetic energy
a	Crank radius
B	Cylinder bore
c	Clearance
c_p	Specific heat capacity
c_t	Trapped compression ratio
h_f	Heat of fusion
k	Turbulent kinetic energy
L	Stroke
l	Connection rod length
L_c	Characteristic length
$m_{d,fuel}$	Delivered fuel
$m_{dr,fuel}$	Delivered retained, fuel
m_{dr}	Delivered retained air fuel mixture mass
m_d	Delivered air fuel mixture mass
n	Engine speed
p	Pressure
Pr	Prandtl number
r_c	Geometric compression ratio

Re	Reynolds number
S	Source term
t	Time
U_c	Characteristic velocity
V_c	Dead volume
V_d	Displaced volume
V_{cv}	Cell volume
z	Piston position
ABDC	After bottom dead center
ATDC	After top dead center
BBDC	Before bottom dead center
BDC	Bottom dead center
BTDC	Before top dead center
CAD	Crank angle degrees
RANS	Reynolds-Averaged Navier-Stokes
TDC	Top dead center

Preface

The aim of this work is to perform simulations of the flow in a two-stroke engine. This aim has two main purposes. One is to study the results of the simulations and thereby learn more about two-stroke engines. The second main reason is to validate the software with which the simulations will be performed. This software is named OpenFOAM-1.5-dev and is used to perform flow calculations in three dimensions.

The work started in January 2010 and was finished in June 2010. It was carried out at Husqvarna AB in Jönköping for the Division of Fluid Dynamics at the Department of Applied Mechanics at Chalmers University of Technology in Gothenburg. Associate Professor Håkan Nilsson is the examiner and has also been supervising. At Husqvarna AB have Erik Lindén and Johan Spång been supervising.

Göteborg June 2010

Mikael Jönsson

Acknowledgements

I would like to thank my supervisors Erik Lindén and Johan Spång at Husqvarna AB for their great support and supervision throughout this work. I would also like to thank Associate Professor Håkan Nilsson at Chalmers University of Technology. As my examiner he has not only directed this project but also been of great help with his expertise during the work. Tommaso Lucchini at Politecnico di Milano is greatly acknowledged for his technical support and feedback. During the work Mikael Bergman at Husqvarna AB has provided me with support and his knowledge about two-stroke engines. During the work I have been provided with resources and knowledge from Husqvarna AB.



1 Introduction

This section gives the background to this work and describes the purpose of it. The aim is defined together with the theory which is required to reach it. The approach to solve this problem is also presented. The approach will introduce some limitations which are described in this section.

1.1 Background

The two-stroke engine was introduced in the 1870s and 1880s. In 1875 an american by the name George Brayton presented his two-stroke engine. A couple of years later two englishmen, Dugald Clerk and James Robson, presented their engine. The german engineer Karl Benz had also developed his two-stroke engine, so work was going on worldwide.

Combustion engines can be distinctly categorised in two types. These two categories are based on which type of ignition is used to start the combustion. The two types are the compression ignited, CI-, engine and the spark ignited, SI-, engine. In the CI-engine the charge, containing fuel and air, is ignited due to high pressure. A SI-engine charge is ignited by a spark. Both compression ignited and spark ignited two-stroke engines are available. From now on only the latter of the two will be discussed and the SI two-stroke engine will simply be referred to as the two-stroke engine. The two-stroke engine is characterised by its high power to mass ratio. It has also a low complexity thanks to its simple construction which contains few moving parts. These properties makes it suited for handheld products and lighter vehicles such as motorcycles and mopeds, which are the main areas of application for the two-stroke engine.

Designing an engine is not a simple task. There are many parameters that will affect the performance of the engine. But performance, in terms of power and torque, is not the only target. When an engine is developed for commercial purposes the number of challenges rises. Legislation as well as the consumers will put demands on noise, fuel consumption, levels of emissions etcetera. To design an engine which fulfills these requirements a number of tools could be useful. Often an engine is developed using a combination of practical experiments, computer aided simulations and of course previous experiences. These tools are used to predict the engine characteristics but also to evaluate the design. Focus will now be put on computer aided simulations and flow simulations in particular.

Calculations which models the flow of fluids in an engine can be used to predict and model a number of properties. A commonly used tool is the 1-dimensional flow calculation softwares, [2] [3]. Among all the properties such a program can predict are engine power, torque and levels of emissions. When a more detailed description of a certain flow is required, a 3-dimensional flow calculation method is preferred, [6]. A 3-dimensional flow simulation can give information such as weak spots in the engine design in terms of flow characteristics. The 3-dimensional flow calculations will from now on be referred to as computational fluid dynamics, CFD. The CFD is used to calculate temperature, pressure, velocities and other properties used to characterise a flow. The combustion process can also be modelled by CFD. The information which might be provided by these features could be very helpful when designing or performing research on two-stroke engines.

1.2 Aim and purpose

The aim of this work is to simulate two-stroke engine scavenging using CFD. Scavenging is the part of the two-stroke engine operating cycle when the combustion products, named exhaust gases, inside the cylinder are supposed to be replaced by a fresh charge of mixed

air and fuel. The engine which will be the object of evaluation is the Husqvarna H576XP. From now on this engine will be referred to as H576.

There are two main purposes of performing these simulations. One of the reasons is the evaluation of scavenging performance of the H576 engine. The results can be used as recommendations and guidelines for future engine design projects or research. The second main purpose is to evaluate the use of the CFD software OpenFOAM for this kind of application. OpenFOAM is an open source CFD software written in C++. OpenFOAM-1.5-dev, which is an extended version of OpenFOAM-1.5, contains a library which is developed for two-stroke engine simulations.

1.3 Theory

One of the purposes of this work is, as mentioned in the previous section, to evaluate the scavenging of the H576 engine. This requires knowledge about the two-stroke engine operating cycle. The criteria upon which the performance will be judged determines the requirements on the simulation. The evaluation methodology and the corresponding procedure, is presented in this section. First the two-stroke engine operating cycle is described. For further information about two-stroke engines and stratified charged two-stroke engines see [3], [2] and [1]

1.3.1 The two-stroke engine

Before describing the two-stroke engine operating cycle the construction and the included parts are presented. The parts are described according to the numbering in Figure 1.1. A two-stroke engine basically contains three moving parts, the piston (1) which is connected to the crankshaft (8) by the connection rod (3). The crankshaft (8) is located in the crankcase (9). The crankcase (9) volume and the cylinder (2) volume are separated by a dashed line in Figure 1.1. The cylinder walls are also known as the liner. Figure 1.1 also shows the scavenging port (4), the scavenging channel (5), the scavenging inlet (11), the exhaust port (6), the exhaust channel (7) and the intake channel (10).

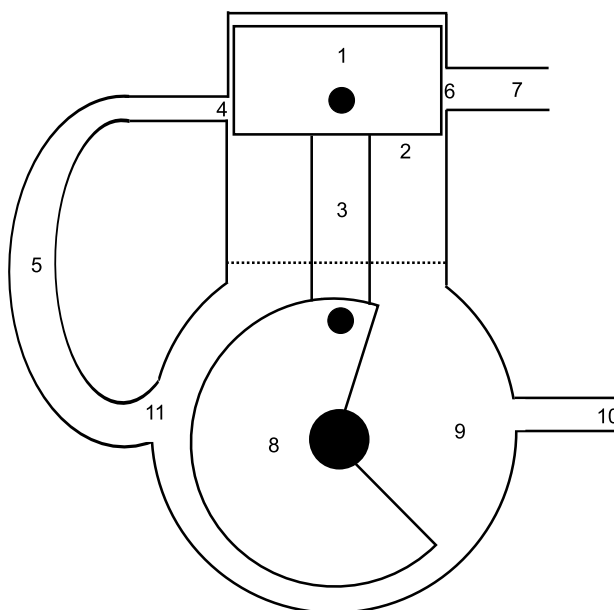


Figure 1.1: This figure shows a schematic of a two-stroke engine.

A two stroke engine completes a cycle in two strokes which means one revolution of the crankshaft, hence the name of the engine. The two strokes are named as the compression and expansion strokes. As the crankshaft turns the piston will move in the cylinder. The piston has two directions of motion. The first direction, downwards, is the case when the piston moves towards the crankcase. The cylinder volume above the piston is now expanding. This part of the cycle is defined as the expansion. The second direction of motion, upwards, is when the piston is moving away from the crankcase. Now the cylinder volume above the piston is decreasing. This is defined as compression. The uppermost position of the piston is defined as the top dead center. This is indicated in Figure 1.2 as TDC with a dashed line. From now on will the top dead center be referred to as TDC. The piston is at TDC in Figure 1.2. In Figure 1.2 is also the lowermost position of the piston indicated. It is called the bottom dead center and is shown with a dashed line and the acronym BDC in figure 1.2. The bottom dead center is now referred to as BDC. Figure 1.3 shows the engine when the piston is at BDC. The crankshaft has turned half a revolution compared to the crankshaft position in Figure 1.2. To indicate the crankshaft position the crank angle degrees, CAD, are introduced. In this work the starting point, 0 CAD, is set to TDC. This results in that BDC occurs at 180 CAD. The CAD is indicated as θ [deg] in Figure 1.3.

The cylinder volume captured between TDC and BDC is defined as the swept, or displaced volume, V_d [m³]. The cylinder volume which is captured above TDC is called the dead volume, V_c [m³].

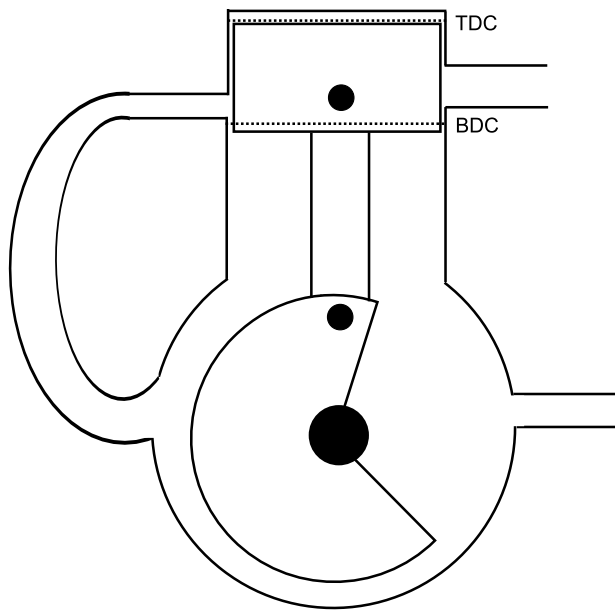


Figure 1.2: A two-stroke engine with the piston at TDC.

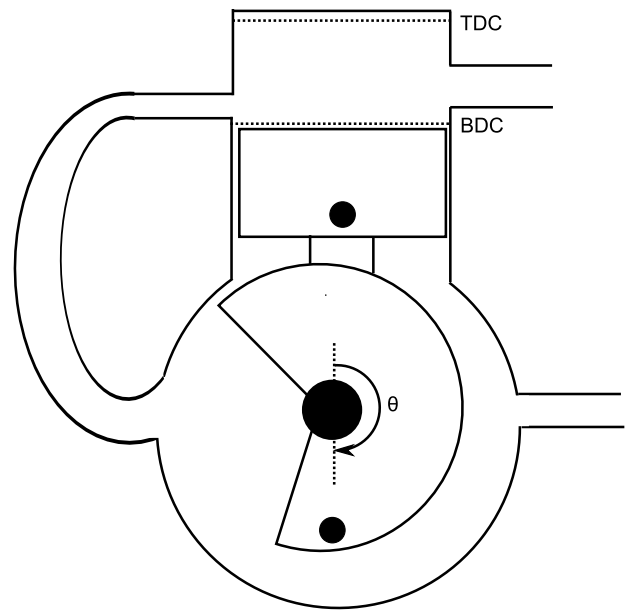


Figure 1.3: A two-stroke engine with the piston at BDC.

The displaced volume, V_d , and the dead volume, V_c are used to calculate the engine's geometric compression ratio, r_c [-]. The compression ratio is a common number when an engine is specified. Equation (1.1) shows the definition of the compression ratio. However, another type of compression ratio is more interesting for two-stroke engines, named the trapped compression ratio. The trapped compression ratio, c_t [-], is defined as the ratio of the volume above TDC and the volume from the top of the exhaust port to the top of the cylinder.

$$r_c = \frac{V_c + V_d}{V_c} \quad (1.1)$$

When working with engines, CAD are rather used as time measurement than seconds. The CAD, θ [deg], as a function of time, t [s], and engine speed, n [rpm], is shown in equation (1.2). The engine speed, n [rpm], is defined as the rotational speed of the crankshaft.

$$\theta(t) = \frac{n360t}{60} \quad (1.2)$$

The piston position at a current θ depends on the measures of the engine parts. Figure 1.4 shows the information required to calculate the piston position. In Figure 1.4 is l [m] the connection rod length, a [m] the crank radius, c [m] the clearance, B [m] the cylinder bore. The cylinder bore is equal to the diameter of the cylinder. Using this information the piston position, z [m], can be calculated as shown equation (1.3).

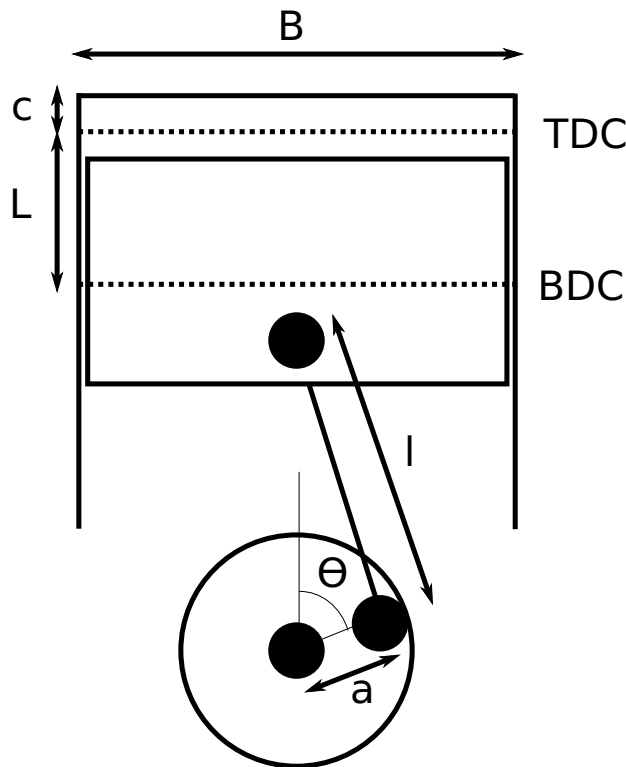


Figure 1.4: Definition of engine parameters for calculating the piston position.

$$z(\theta) = l + a + c - a \cos(\theta) - (l^2 - a^2 \sin^2(\theta))^{\frac{1}{2}} \quad (1.3)$$

The relation between a [m] and the stroke L [m] is found in equation 1.4. When looking at equation (1.3) it is obvious that the distance travelled for each crank angle degree will vary over the stroke, since the function is non-linear.

$$a = \frac{L}{2} \quad (1.4)$$

The piston position as a function of CAD for the H576 engine is shown in Figure 1.5.

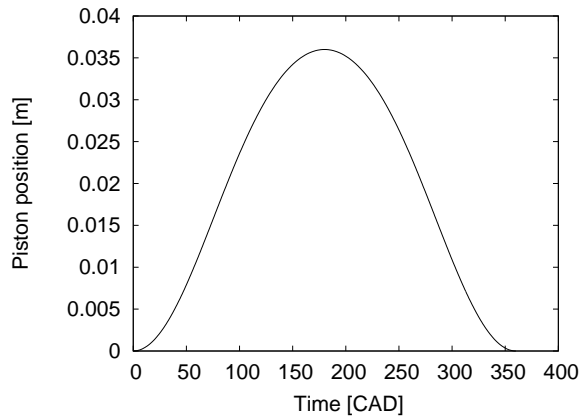


Figure 1.5: Piston position over crank angles for the H576 engine.

The definitions and concepts described so far in this section are useful when describing the two-stroke engine operating cycle.

1.3.2 Conventional two-stroke engine

This section is based on the concepts and nomenclature found in the previous section, section 1.3.1. The expansion stroke starts at 0 CAD. The expansion is a result of the combustion. Commonly the combustion starts a couple of CAD before TDC. The combustion is initiated by a spark from the spark plug. The spark plug is located in the top of the cylinder. The spark ignites the fresh charge, which is a mixture of air and fuel, trapped in the cylinder.

As the cylinder volume expands due to the combustion the piston will move towards the crankcase. The crankshaft and the crankcase are designed to create a lower pressure in the crankcase than in the intake channel. This difference in pressure will force fresh mixture into the crankcase, see Figure 1.6. Upstream of the intake channel is the carburettor located. The carburettor is used to evaporate the fuel so it mixes with the fresh air. Upstream of the carburettor is a throttle located. The throttle is used to control the amount of air supplied to the engine.

Eventually will the piston reach the exhaust port. When the exhaust port is reached it will begin to open. The pressure in the cylinder is much higher than the pressure in the exhaust channel. This pressure difference is a result of the combustion and will force the combustion products out of the cylinder into the exhaust channel.

The expansion continues and eventually will also the scavenging port begin to open. The pressure of the fresh mixture in the crankcase has now increased. The pressure has increased when the crankshaft has turned in the crankcase. Since the cylinder volume has expanded and the exhaust port has opened the pressure in the cylinder has decreased to a level below the pressure in the scavenging channel.

The scavenging will now start as the fresh mixture is entering the cylinder, while the combustion products are exiting the cylinder through the exhaust port into the exhaust channel. The piston will continue toward the crankcase until BDC is reached. BDC is reached at 180 CAD. As BDC is reached the direction of motion will change and the compression stroke will begin, see Figure 1.7.

The compression begins and the cylinder volume is now decreasing. The area of the scavenging port and exhaust port decreases as the piston moves. First the scavenging port is entirely closed and later on the exhaust port is entirely closed. The trapped fresh mixture is now being compressed as the piston moves towards TDC. Before the piston reaches

TDC the fresh mixture will be ignited by a spark from the spark plug. The combustion starts and the cycle will start over.

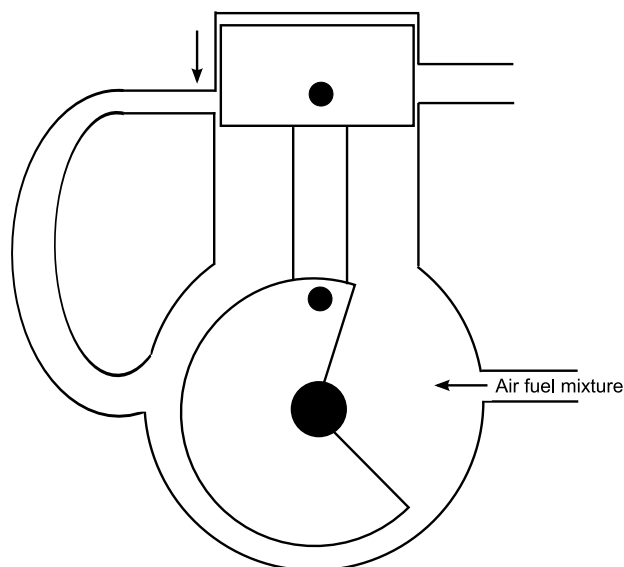


Figure 1.6: The beginning of the expansion stroke.

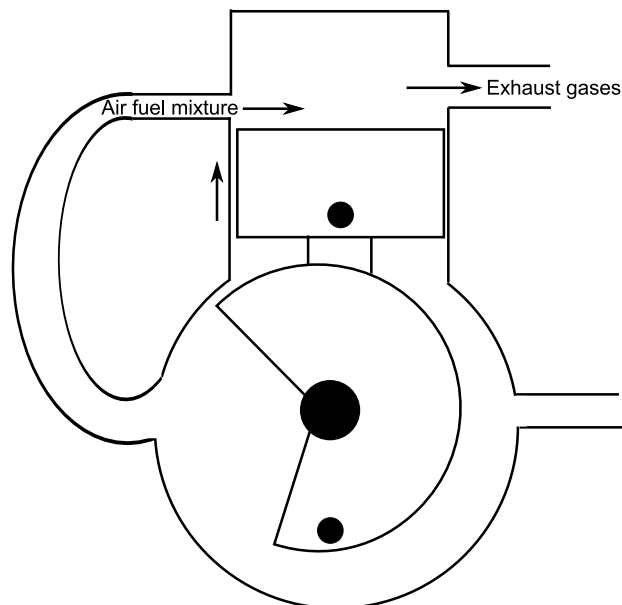


Figure 1.7: The beginning of the compression stroke.

1.3.3 Stratified charged two-stroke engine

The stratified charged two-stroke engine is now introduced. The H576 engine is a stratified charged engine. The purpose of the stratified charged two-stroke engine is to reduce the amount of fresh mixture going out in the exhaust channel uncombusted. As seen in section 1.3.2, there is a phase in the two-stroke engine cycle when the exhaust and scavenging ports are open at the same time. This allows fresh mixture to go through the cylinder uncombusted, which is a drawback for the two-stroke engine since the fuel consumption and level of emissions will increase. To determine how well the scavenging works, i.e. how much of the fresh mixture that is delivered in to the cylinder will be trapped, the trapping efficiency is introduced. The trapping efficiency, η_{tr} [-] is defined as the ratio between the mass of the delivered mixture retained in the cylinder after the exhaust port has closed, m_{dr} [kg], and the mass of the mixture delivered to the cylinder, m_d [kg]. Equation (1.5) is used to calculate the trapping efficiency.

$$\eta_{tr} = \frac{m_{dr}}{m_d} \quad (1.5)$$

To increase the trapping efficiency, and thereby reduce the emissions and lower the fuel consumption, the stratified charged engine was developed. This engine is based on the conventional two-stroke engine, see section 1.3.2. The engine geometry is however modified in two major ways. An additional channel is added to the cylinder. This channel is named the stratifying channel. The stratifying channel is, in difference from the intake channel, not supplied with neither a carburettor nor a throttle. The channel is thereby only supplied with fresh air at ambient, (atmospheric), pressure.

The second major difference from the conventional two-stroke engine is the piston design. The piston in the stratified charged engine has a pocket. This pocket's function is to create a pass between the stratifying channel and the scavenging channel. Figure 1.8 shows the parts of the stratified charged engine. The piston (1), the crankshaft (8), the

connection rod (3), the crankcase (9), the , the cylinder (2), the scavenging port (4), the scavenging channel (5), the scavenging inlet (11), the exhaust port (6), the exhaust channel (7) and the intake channel (10) are all found in the two-stroke engine. The stratifying port (12) and the stratifying channel (13) are not present in the conventional two-stroke engine. The piston pocket (14) which creates the pass between the scavenging port and the stratifying port is not present in the conventional two-stroke engine either.

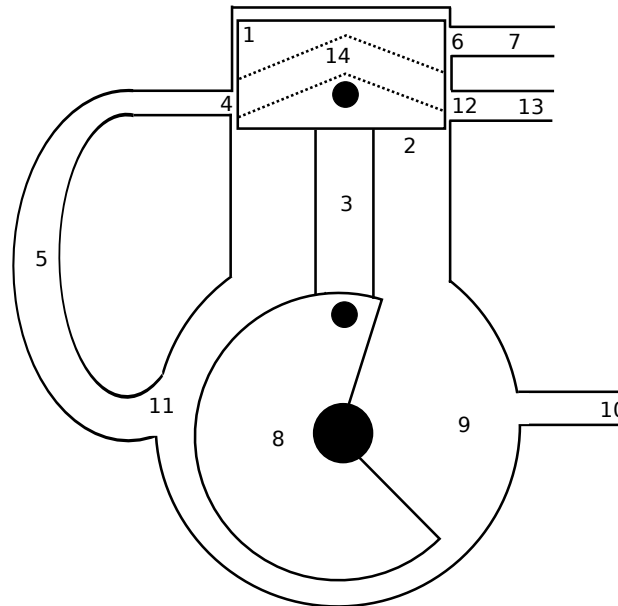


Figure 1.8: Piston position over crank angles for the H576 engine.

Due to the movement of the piston and the shape of the pocket, the pass will only be present within a certain interval of CAD. At this interval, the pressure in the crankcase and scavenging channel is below the pressure in the stratifying channel. This pressure difference will force fresh air from the stratifying channel through the pass in the piston, into the scavenging port and down the scavenging channel. The air fuel mixture in the scavenging channel will be surpressed by the fresh air, see Figure 1.9. Eventually will the pass be closed and a buffert of air will remain in the top of the scavenging channel.

When the scavenging port opens towards the cylinder the scavenging port contents will start to flow into the cylinder. Of course has the fresh air and the air fuel mixture started to mix. However will the fluid that first enters the cylinder have a rather low concentration of fuel and a high concentration of air. This will result in a lower concentration of fuel in the part of the fresh air fuel mixture that will not be trapped in the cylinder, see Figure 1.10. This is the idea of how the stratified charged engine reduces emissions and lowers the fuel consumption.

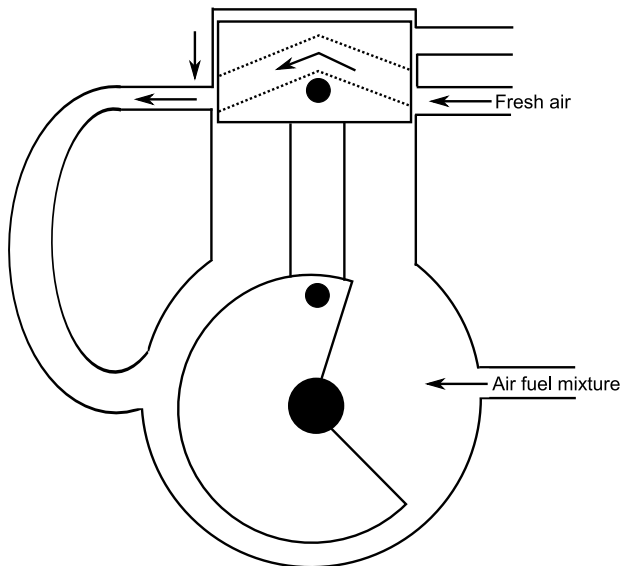


Figure 1.9: The beginning of the expansion stroke.

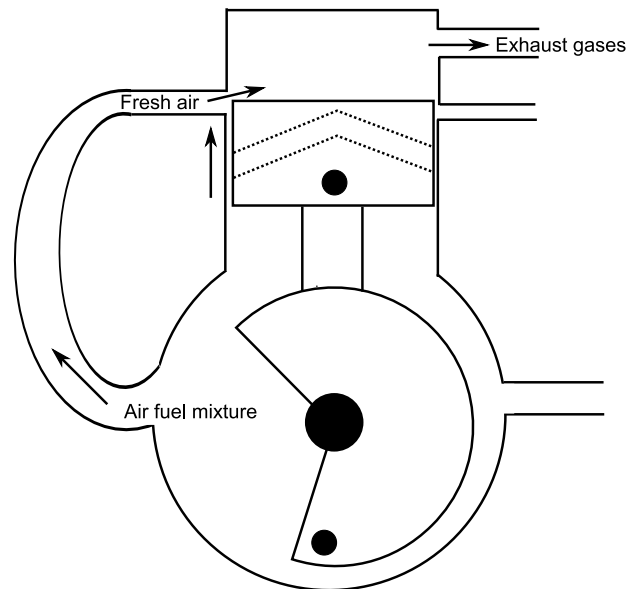


Figure 1.10: The beginning of the compression stroke

Below is a schedule over these instances for the H576 engine.

- 0 CAD ATDC. The piston starts to move down.
- 103 CAD ATDC. The exhaust port starts to open.
- 116 CAD ATDC. The intake ports start to open.
- 180 CAD ATDC. BDC is reached. The piston direction of motion changes. The exhaust and intake ports begin to close.
- 244 CAD ATDC. The intake ports are completely closed.
- 257 CAD ATDC. The exhaust port is completely closed.
- 360 CAD ATDC. TDC is reached and the cycle starts over.

One must bear in mind that this is not a method of increasing the trapping efficiency. This makes sense when looking at the definition of the trapping efficiency and in particular equation (1.5). This states that the trapping efficiency is a measure of the fresh mixture trapped in the cylinder. The stratified charged engine will probably have the same trapping efficiency as a conventional two-stroke engine. The essential difference is that the fresh mixture leaving the cylinder unburned will have a much lower fuel concentration than the unburned mixture leaving the cylinder in a conventional two-stroke engine. A more interesting measure for the stratified charged engine is to calculate the trapping efficiency for the fuel rather than the air fuel mixture. The definition of the fuel trapping efficiency, $\eta_{tr,fuel}$ [-], is found in equation (1.6). The mass of the delivered fuel retained is denoted by $m_{dr,fuel}$ [kg] and the mass of the delivered fuel by $m_{d,fuel}$ [kg]. From now on the fuel trapping efficiency $\eta_{tr,fuel}$ is referred to as the trapping efficiency.

$$\eta_{tr,fuel} = \frac{m_{dr,fuel}}{m_{d,fuel}} \quad (1.6)$$

1.3.4 The governing equations

There are different ways to evaluate an engine. The method that will be used depends on what kind of information is required to make a judgement. In this work the H576 engine will be evaluated by simulations of the flow in the cylinder and the channels described in section 1.3.3. To perform simulations, some kind of model is required to describe the flow. One method to describe the flow motion is to use the Navier-Stokes equations. Before the Navier-Stokes equations are described some concepts must be introduced.

The concept of the control volume is fundamental. The control volume is the volume for which the Navier-Stokes equations will be solved. For example, the control volume could be the cylinder volume. It could also be a volume defined in space with no physical walls.

Equation (1.7) is known as the continuity equation. The continuity equation states that the rate of increase of mass in the control volume is equal to the net rate of flow of mass into the control volume. Physically this means that the mass is conserved, since mass can neither be created nor destroyed. By summing up the forces acting on the fluid in the control volume the Navier-Stokes momentum equations are obtained. The momentum equations in the x-, y- and z-direction are found in equation (1.8), (1.9) and (1.10) respectively. In equation (1.7)-(1.10), ρ [kg/m³] is the fluid density, μ [kg/ms] the dynamic viscosity, p [Pa] the pressure and \mathbf{u} [m/s] the velocity vector. The velocity vector \mathbf{u} has three component, u , v and w . These components are the velocities in the x-, y- and z-direction respectively. The S_u , S_v and S_w are the source terms in the corresponding direction.

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{u}) = 0 \quad (1.7)$$

$$\frac{\partial(\rho u)}{\partial t} + \text{div}(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + \text{div}(\mu \text{grad}(u)) + S_u \quad (1.8)$$

$$\frac{\partial(\rho v)}{\partial t} + \text{div}(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + \text{div}(\mu \text{grad}(v)) + S_v \quad (1.9)$$

$$\frac{\partial(\rho w)}{\partial t} + \text{div}(\rho w \mathbf{u}) = -\frac{\partial p}{\partial z} + \text{div}(\mu \text{grad}(w)) + S_w \quad (1.10)$$

These equations are used to describe a compressible, non-steady, three-dimensional flow. The equations are further described in [7] and [8].

1.3.5 The RANS-equations and turbulence modelling

The governing equations in the previous section, section 1.3.4, are used to describe a flow. How these equations will be used depends on the flow. A rough categorisation is to determine if the flow is laminar or turbulent. To do this the Reynolds number, Re [-], is introduced. This is a non-dimensional number found in equation 1.11. The variables U_c [m/s] and L_c [m] are the characteristic velocity and the characteristic length respectively and ν [m²/s] is the kinematic viscosity. The kinematic viscosity is the ratio between the dynamic viscosity, μ [kg/sm], and the fluid density, ρ [kg/m³], see equation 1.12.

$$Re = \frac{U_c L_c}{\nu} \quad (1.11)$$

$$\nu = \frac{\mu}{\rho} \quad (1.12)$$

Above a certain Reynolds number, Re_{crit} , of approximately 2000 is the flow considered to be turbulent. The Reynolds number in the intake channel is estimated to 86000 for the actual case, and the flow is thereby considered turbulent. From the 1-D simulation indata, section 2.4, the maximum massflow, \dot{m} [kg/s], is found. Using the pressure, p [Pa], and temperature, T [K], at that particular CAD gives ρ and μ using equation (1.13) and (1.14). The intake diameter is used as characteristic length, L_c [m]. The intake diameter and massflow is used to find the characteristic velocity U_c [m/s], see equation (1.15). The values used to calculate the Reynolds number is found in Table 1.1

$$\rho = \frac{p}{RT} \quad (1.13)$$

$$\mu \approx \mu_0 \left(\frac{T}{T_0} \right)^{0.7} \quad (1.14)$$

Where $R \approx 287[\text{J/kgK}]$, $T_0=273[\text{K}]$ and $\mu_0=1.71 \cdot 10^{-5}[\text{kg/ms}]$.

$$U_c = \frac{\dot{m}}{\frac{L_c^2 \pi}{4} \rho} \quad (1.15)$$

Table 1.1: Values for the Reynolds number calculation.

Mass flow, \dot{m} [kg/s].	0.01474
Pressure, p [Pa].	163574
Temperature, T [K].	408
Characteristic length, L_c [m]	0.0095
Density, ρ [kg/m ³]	1.4
Characteristic velocity, U_c [m/s]	148
Dynamic viscosity, μ [Ns/m ²].	$2.265 \cdot 10^{-5}$

For turbulent compressible flows a common approach is to use the RANS equations. The acronym RANS means Reynolds-Averaged Navier-Stokes. The Navier-Stokes equations, see equation (1.7)-(1.10) in section 1.3.4, are used to construct the RANS equations. Turbulent flows are random and the variables will be considerably fluctuating. To take the fluctuations into account the Reynolds-decomposition is used. The Reynolds-decomposition separates a variable into two parts. One part is the mean value of the current variable. The second part is used to take the fluctuations into account. The Reynolds-decomposed velocity and pressure components are found in the equations (1.16)-(1.19). The part denoted by a captial letter is the mean value of the variable. The parts denoted by lower case letters with a prime are the fluctuation parts.

$$u(t) = U + u'(t) \quad (1.16)$$

$$v(t) = V + v'(t) \quad (1.17)$$

$$w(t) = W + w'(t) \quad (1.18)$$

$$p(t) = P + p'(t) \quad (1.19)$$

The Reynolds-decomposed variables, (1.16)-(1.19), are now substituted into the Navier-Stokes equations, equation (1.7)-(1.10). After the substitution the equations are time-averaged. The Reynolds-Averaged Navier-Stokes are now obtained. The RANS equations are found in equation (1.21)-(1.23). The velocity terms indicated by tilde are Favre-averaged terms. This is a way of averaging using the density. Equation (1.24) shows the

Favre-averaged velocity component in the x-direction.

$$\frac{\partial \bar{\rho}}{\partial t} + \text{div}(\bar{\rho} \tilde{\mathbf{U}}) = 0 \quad (1.20)$$

$$\begin{aligned} \frac{\partial(\bar{\rho} \tilde{U})}{\partial t} + \text{div}(\bar{\rho} \tilde{U} \tilde{\mathbf{U}}) &= -\frac{\partial \bar{P}}{\partial x} + \text{div}(\mu \text{grad } \tilde{U}) + \\ + \left[-\frac{\partial(\overline{\rho u'^2})}{\partial x} - \frac{\partial(\overline{\rho u'v'})}{\partial y} - \frac{\partial(\overline{\rho u'w'})}{\partial z} \right] + S_{Mx} & \quad (1.21) \end{aligned}$$

$$\begin{aligned} \frac{\partial(\bar{\rho} \tilde{V})}{\partial t} + \text{div}(\bar{\rho} \tilde{V} \tilde{\mathbf{U}}) &= -\frac{\partial \bar{P}}{\partial y} + \text{div}(\mu \text{grad } \tilde{V}) + \\ + \left[-\frac{\partial(\overline{\rho u'v'})}{\partial x} - \frac{\partial(\overline{\rho v'^2})}{\partial y} - \frac{\partial(\overline{\rho v'w'})}{\partial z} \right] + S_{My} & \quad (1.22) \end{aligned}$$

$$\begin{aligned} \frac{\partial(\bar{\rho} \tilde{W})}{\partial t} + \text{div}(\bar{\rho} \tilde{W} \tilde{\mathbf{U}}) &= -\frac{\partial \bar{P}}{\partial z} + \text{div}(\mu \text{grad } \tilde{W}) + \\ + \left[-\frac{\partial(\overline{\rho u'w'})}{\partial x} - \frac{\partial(\overline{\rho v'w'})}{\partial y} - \frac{\partial(\overline{\rho w'^2})}{\partial z} \right] + S_{Mz} & \quad (1.23) \end{aligned}$$

$$\tilde{U} = \frac{\overline{\rho u}}{\bar{\rho}} \quad (1.24)$$

The Reynolds-averaging gives rise to new term in the Navier-Stokes equations. The terms containing fluctuating parts are unknown. These terms must be modelled and a common approach is to use two-equation models. This is referred to as turbulence modelling. In this work the standard $k - \varepsilon$ turbulence model is used. It adds two equations to the equation system, one for k [m^2/s^2] which is the turbulent kinetic energy and one for ε [m^2/s^3] which is the dissipation of the turbulent kinetic energy. The two additional equations are found in equation (1.25) and (1.26). The values in equation (1.27) are used for the $k - \varepsilon$ coefficients.

$$\begin{aligned} \frac{\partial(\rho \varepsilon)}{\partial t} + \text{div}(\rho \mathbf{U} \varepsilon) - \nabla^2((\alpha_\varepsilon \mu_t + \mu) \varepsilon) = \\ C_{1\varepsilon} \frac{\varepsilon}{k} (\mu_t 2(S_{ij} \bullet S_{ij})) - \left(\frac{2}{3} C_{1\varepsilon} + C_{3\varepsilon} \right) \text{div}(\rho \mathbf{U} \varepsilon) - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \end{aligned} \quad (1.25)$$

$$\begin{aligned} \frac{\partial(\rho k)}{\partial t} + \text{div}(\rho \mathbf{U} k) - \nabla^2((\alpha_k \mu_t + \mu) k) = \\ \mu_t 2(S_{ij} \bullet S_{ij}) - \frac{2}{3} \text{div}(\rho \mathbf{U} k) - \rho \varepsilon \end{aligned} \quad (1.26)$$

$$\begin{aligned} C_\mu &= 0.09 \\ C_{1\varepsilon} &= 1.44 \\ C_{2\varepsilon} &= 1.92 \\ C_{3\varepsilon} &= -0.33 \\ \alpha_h &= 1 \\ \alpha_k &= 1 \\ \alpha_\varepsilon &= 0.76923 \end{aligned} \quad (1.27)$$

Special care must be taken for flows near walls. In this work the $k - \varepsilon$ -model is used with wall functions. The wall functions are used to take near wall behavior into account. The

wall functions gives the following equations for k and ε near the wall, see equation (1.28) and (1.29). To find u_τ equation (1.30) is iterated by using the log-law which states that $u^+ = y^+$. The log-law is valid close to the wall in turbulent flows and the layer close to the wall is thus called the log-law layer.

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \quad (1.28)$$

$$\varepsilon = \frac{u_\tau^3}{\kappa y} \quad (1.29)$$

$$u^+ = \frac{U}{u_\tau} = \frac{1}{\kappa} \ln(Ey^+) \quad (1.30)$$

where

$$\begin{aligned} E &= 9 \\ \kappa &= 0.4187 \end{aligned} \quad (1.31)$$

In this work the PISO-algorithm is used for pressure velocity coupling. PISO is an acronym for Pressure Implicit with Splitting of Operators. The PISO-algorithm is described in [7]. See [7] and [8] for further a further description of this section.

1.3.6 The finite volume method

The finite volume method is used to solve the RANS-equations, see section 1.3.5, in this work. As discussed in section 1.3.4, a control volume is defined over which the equations are solved. To increase the accuracy of the results the computational domain can be split into several control volumes. A computational domain which is split into a number of control volumes is defined as a mesh. The finer the mesh is, the higher accuracy will the results have. The downside is that the equation system will grow with the resolution of the mesh. The idea of the finite volume method is to solve the governing equations, (the RANS-equations), for each control volume in the mesh. An example of how to set up a problem using the finite volume method is now presented.

The heat conduction equation is to be solved using CFD and the finite volume method as an example. It is a 1-dimensional example, (in the x-direction), and the heat conduction equation is found in equation (1.32). In equation (1.32), is k [J/smK] the thermal conductivity coefficient, T [K] the temperature and S [J/sm²] the source term. The mesh consists of three equally large control volumes, shown in Figure 1.11. The finite volume method is described more thoroughly in [7].

$$\frac{d}{dx} \left(k \frac{dT}{dx} \right) + S = 0 \quad (1.32)$$

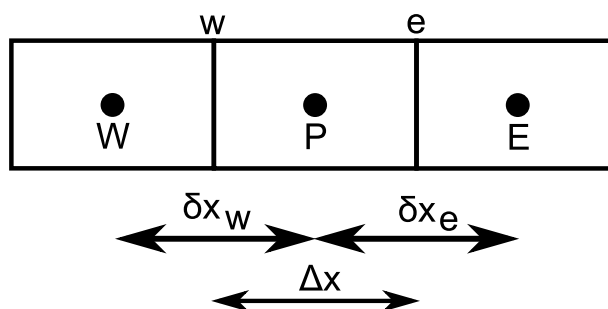


Figure 1.11: Example of a computational domain for a one dimensional finite volume case.

The heat conduction equation, equation (1.32), is discretised and integrated over the middle control volume in the mesh, see Figure 1.11. The resulting equation is found in (1.33). The center of the control volume is named node. According to the notation in Figure 1.11, node values are denoted by capital letters and control volume boundary are values denoted by lower case letters.

$$\int_e^w \left[\frac{d}{dx} \left(k \frac{dT}{dx} \right) + S \right] dx = \left(k \frac{dT}{dx} \right)_e - \left(k \frac{dT}{dx} \right)_w + \bar{S} \Delta x = 0 \quad (1.33)$$

The central-differencing discretisation scheme is used to calculate the derivatives at the control volume boundaries, see equation (1.34).

$$\left(\frac{dT}{dx} \right)_w \simeq \frac{T_P - T_W}{\delta x_w} \quad \left(\frac{dT}{dx} \right)_e \simeq \frac{T_E - T_P}{\delta x_e} \quad (1.34)$$

The equation for the middle control volume can now be rewritten as equation 1.35 using the coefficients found in 1.36.

$$a_P T_P = a_W T_W + a_E T_E + S \quad (1.35)$$

$$a_W = \frac{k_w}{\delta x_w} \quad a_E = \frac{k_e}{\delta x_e} \quad S = \bar{S} \Delta x \quad a_P = a_W + a_E \quad (1.36)$$

The steps shown in equation (1.33)-(1.36) are performed for the two remaining control volumes in the mesh. An equation system can now be set up, shown in equation (1.37). The temperatures T_A and T_B are the boundary temperatures.

$$\begin{bmatrix} a_{P1} & -a_{E1} & 0 & 0 \\ -a_{W2} & a_{P2} & -a_{E2} & 0 \\ 0 & -a_{W3} & a_{P3} & -a_{E3} \\ 0 & 0 & -a_{W4} & a_{P4} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} S_1 + a_{W1} T_A \\ S_2 \\ S_3 \\ S_4 + a_{E4} T_B \end{bmatrix} \quad (1.37)$$

An example of an iterative method that can be used to solve this equation system is the Gauss-Seidel iteration method. See equation (1.38), where the superscript 0 indicates values from the initial guess if it is the first iteration. If it is not the first iteration, superscript 0 indicates values from the previous iteration. Values with superscript 1 are taken from the current iteration. Hence calculated values are used in the following equations in the current iteration.

$$\begin{aligned} T_1^1 &= \frac{a_{E1} T_2^0 + a_{W1} T_A + S_1}{a_{P1}} \\ T_2^1 &= \frac{a_{E2} T_3^0 + a_{W2} T_1^1 + S_2}{a_{P2}} \\ T_3^1 &= \frac{a_{E3} T_4^0 + a_{W3} T_2^1 + S_3}{a_{P3}} \\ T_4^1 &= \frac{a_{E4} T_B + a_{W4} T_3^1 + S_4}{a_{P4}} \end{aligned} \quad (1.38)$$

To determine when to stop the iterative process some kind of criterion is needed. The residuals are calculated to determine when the solution has converged. After each iteration are the residuals calculated. The residuals are found by calculating the difference between the left hand side and right hand side of the equations in (1.38), using the values from the current iteration. The residuals are normalised, commonly by the flux of the current variable. In this example case the heat flux is used to normalise the residuals. The purpose of the normalisation is to make the residuals independent of the current scale.

1.3.7 Fluids

To perform simulations of the flow in a two-stroke engine, according to the theory presented so far, the properties of the present fluids are required. The first step is to determine which fluids to use, and to do this the concept of the cold engine-flow is introduced. A simulation of a cold engine flow is a way of simplify the problem. In this work the cold engine-flow means that no combustion will occur, while in a real engine a number of chemical reactions will take place due to the combustion process. These reactions are not modelled in this work and a further simplification of the problem is to use one fluid only. As a consequence of this there will be no chemical reactions and the only specie that is used as fluid in this work is air.

A number of parameters are used to model the specie. These are the molecular weight W [kg/kmol], the specific heat capacity c_p [J/kgK], the heat of fusion h_f [kJ/kg], the dynamic viscosity μ [Ns/m²] and the Prandtl number Pr [-] which is a nondimensional number found in equation (1.39). More information about the fluid and its definition is found [8] and [5].

$$Pr = \frac{\mu c_p}{k} \quad (1.39)$$

where k is the thermal conductivity coefficient. The coefficient values are found in Table 1.2.

Table 1.2: Fluid properties for the current fluid, air.

Molecular weight, W [kg/kmol].	28.9
Specific heat capacity, c_p [J/kgK].	1300
Dynamic viscosity, μ [Ns/m ²].	$1.84 \cdot 10^{-5}$
Prandtl number, Pr [-].	0.7
Heat of fusion, h_f [kJ/kg]	$2.544 \cdot 10^6$

As mentioned in section 1.3.3, the trapping efficiency gives a good rate of the two-stroke engine scavenging performance. To calculate the trapping efficiency, see equation (1.6) in section 1.3.3, the air fuel mixture needs some kind of representation. When only one fluid is used the trapping efficiency cannot be calculated since the fresh air, the exhaust gases and the air fuel mixture are not separated. The amount of air fuel mixture that is trapped in the cylinder, the amount of air fuel mixture going out the exhaust channel etcetera, cannot be found.

To represent the air fuel mixture, and allow the trapping efficiency to be calculated, a passive scalar, ϕ , is introduced. The passive scalar is a way to mark the parts of the fluid that is supposed to be the air fuel mixture. This gives a representation of the air fuel mixture and makes it traceable. The passive scalar is incorporated with the solution by adding an additional equation to the system of governing equations, the latter equations are found in section 1.3.5. The added equation is found in equation (1.40). The equation for the passive scalar and its physical interpretation will now be discussed.

$$\frac{\partial(\rho\phi)}{\partial t} + \text{div}(\rho\mathbf{U}\phi) = 0 \quad (1.40)$$

When equation (1.40) is solved for a control volume the passive scalar, ϕ , will obtain the same velocity vector as the fluid in the control volume. The scalar is passive since it will not affect the neither fluid nor the flow. The governing equations are independent of equation

(1.40).

The passive scalar, ϕ , can adopt a value between 0 and 1. If ϕ has the value one in a control volume, the concentration of ϕ is one hundred percent. The concentration indicates the mass fraction of the control volume which is indicated by the passive scalar. That the mass is indicated by the passive scalar physically means that the fluid is marked and can be traced. This makes it possible to follow fluid particles in the flow. In this work the air fuel mixture is marked by the passive scalar and can thus be traced. This makes it possible to calculate the trapping efficiency, equation (1.6), despite that one fluid only is present. Equation (1.40) is the passive scalar transport equation and it is only convection based. The first term is the rate of change of the scalar, ϕ , and the second term is the convective term. The flow is considered to be dominated by convection. The effects of diffusion is considered negligible and the diffusion term is therefore omitted.

As previously mentioned, the passive scalar gives the concentration of the air fuel mixture in a control volume. The passive scalar indicates the volume fraction air fuel mixture of the control volume. Since the flow is compressible, the mass of the air fuel mixture in the control volume must be found. The air fuel mixture mass, $m_{af,mixture}$, in a control volume is shown in equation (1.41) where ϕ is the passive scalar concentration, V_{cv} the volume and ρ the density in the current control volume.

$$m_{af,mixture} = \phi * V_{cv} * \rho \quad (1.41)$$

1.4 Approach

The first step of the work is to set the aim and the purpose. The aim is to evaluate the scavenging performance of the H576 engine, see section 1.2. To perform calculations using the finite volume method on this kind of applications some mesh operations must be managed by the solver. The OpenFOAM-1.5-dev is provided with a library, the `twoStrokeEngine`-library, which is developed for two-stroke engine simulations. To test the functions and get familiar with the software a fictional engine geometry is created. The mesh is created from blocks only and a relatively low number of cells. This makes the geometry easy to handle and reduces time for troubleshooting. When the mesh operations are working, some requirements must be fulfilled before starting the calculations.

To solve the governing equations using the finite volume method, see section 1.3.5 and 1.3.6, some things must be set. When solving the partial differential equations boundary conditions are required since not all solutions to the equations are of interest and the solution is dependent on the surroundings. To make the solution as close to the real case as possible data from 1-d simulations are used for some boundary conditions. The 1-d simulations are performed using the Optimum Power software. Optimum Power is a CFD software used typically for engine simulations. It performs calculations in 1-D and it models the combustion process with combustion models. From these simulations the pressure and the temperature at the inlets and outlets as well as in the combustion chamber are obtained. The pressure and the temperature is used as boundary conditions for the simulations. How these and the other boundary conditions are set is further described in section 2.6. The engine speed is constant and is the same as for the 1-d simulations, namely 9600 [rpm] or 160 [rps].

Once the boundary conditions are determined the methodology of how to solve the governing equations is set. This means that the iteration method for each equation and the corresponding convergence criteria are set. The algorithm for solving the equations with a pressure and velocity coupling is chosen. A turbulence model is chosen. When a setup that is stable for the fictional geometry is found the real geometry is to be used.

The real geometry, i.e. the H576 engine geometry, is now introduced. The calculations

are done for the real engine. The passive scalar is used to calculate the trapping efficiency. The simulations are validated by comparing the massflow with the 1-d simulations. A similar comparison is made with data from simulations using the 3-dimensional commercial software Fluent. The choice of turbulence model is also validated by a comparison with another turbulence model.

1.5 Limitations

To solve the problem and calculate the trapping efficiency some limitations are required. The domain consists of the cylinder, the exhaust channel, the scavenging and stratifying channels and the piston pockets. This means that the crankcase is not a part of the domain. Neither the carburettor nor the muffler is modelled.

The fluids, fresh air, air fuel mixture, exhaust gases etcetera are represented by one fluid. The fluid used is air and the air fuel mixture is modelled by a passive scalar. No ignition will occur and thus no combustion. Since only one specie is used and no combustion takes place there will be no chemical reactions.

The calculations are performed at a constant engine speed. The engine geometry is modified to be able to achieve a stable solution. More about the engine geometry modifications are found in section 2.1.1.

2 Method

The methodology and process of this work is described in this section. First the generation of the mesh, in other words the computational domain, and its properties is discussed. The handling of mesh topology changes is also described. The tools, like discretisation schemes etcetera, and boundary conditions that are used to solve the governing equations are presented.

2.1 Engine geometry

The H576 engine, see section 1.2, is now introduced as it is the object of evaluation. Figure 2.1 shows the entire H576 engine geometry at 0 CAD. This position, 0 CAD, is defined as the starting position.

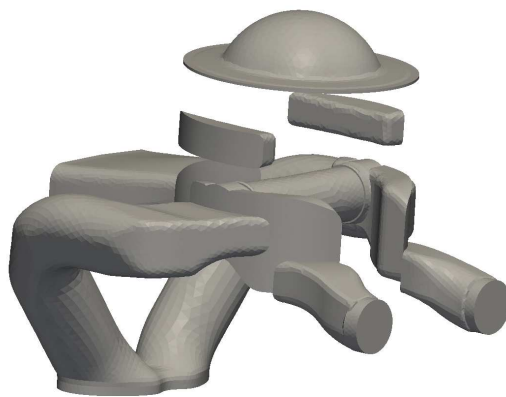


Figure 2.1: The entire geometry at 0 CAD.

The engine geometry consists of nine regions. These parts and their functions are described in section 1.3.2 and 1.3.3. The scavenging channels are shown in Figure 2.2 and the exhaust channel is shown in Figure 2.3. The cylinder (the combustion chamber) is shown in Figure 2.4. Figure 2.5 shows the stratifying channels. The piston pocket in Figure 2.6 is the pocket which creates the path between the scavenging port and stratifying channel. Whereas the function of the piston pockets in Figure 2.7 are to reduce the weight of the piston. Figure 2.8 further describes the part of the H576 engine.

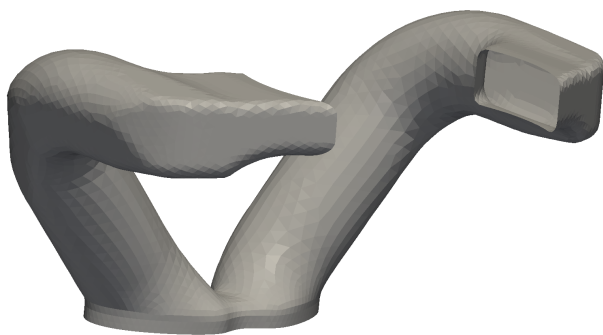


Figure 2.2: The intake channels.

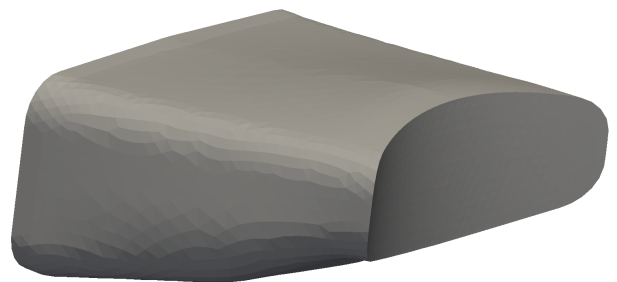


Figure 2.3: The exhaust channel.

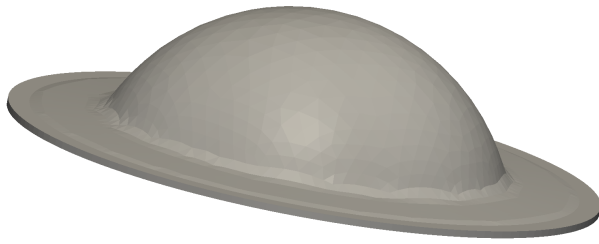


Figure 2.4: The combustion chamber.

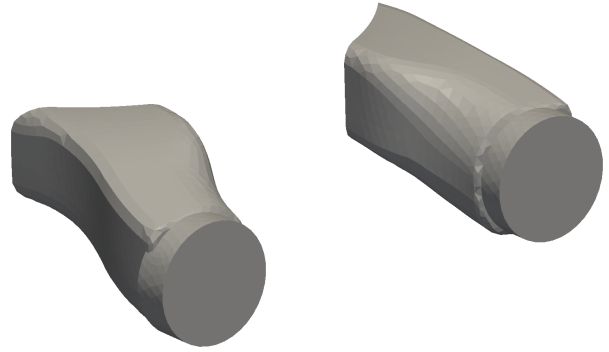


Figure 2.5: The stratifying channels.

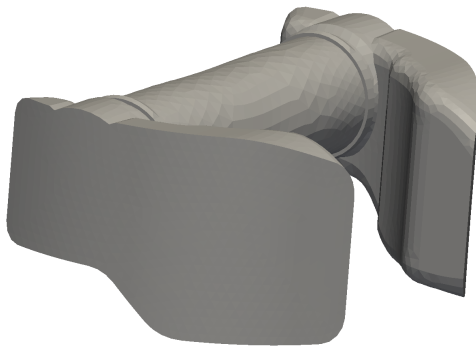


Figure 2.6: The piston pocket allowing the stratified charging.

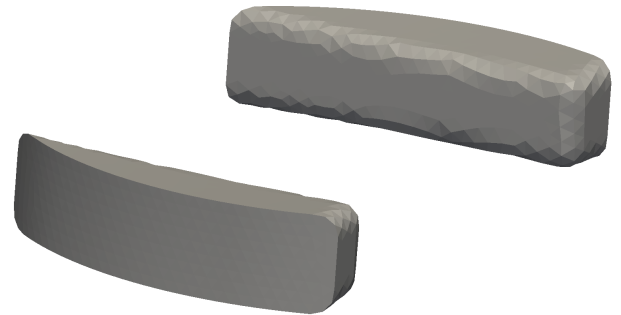


Figure 2.7: Small piston pockets used to reduce the piston weight.

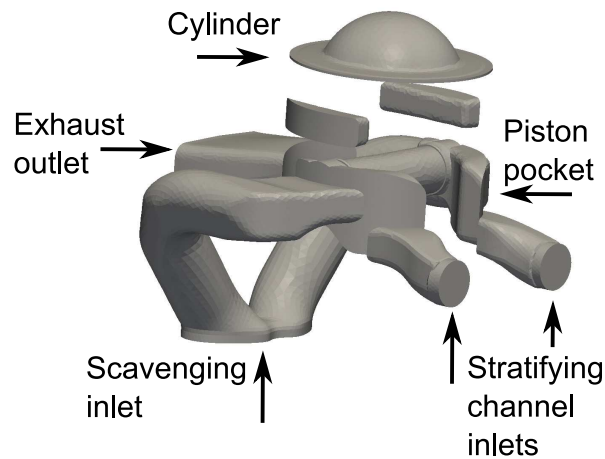


Figure 2.8: A further description of the engine geometry.

2.1.1 Engine geometry modifications

Some modifications are done to the H576 engine geometry. The lower region of the scavenging channel is a source of instability. The sharp edges in the near of the bottom, see Figure 2.2, is a source of divergence due to the turbulence that is immediately introduced. The problem is helped by a modification of the geometry at this region. A comparison of the original and the modified scavenging channel is shown in Figure 2.9.

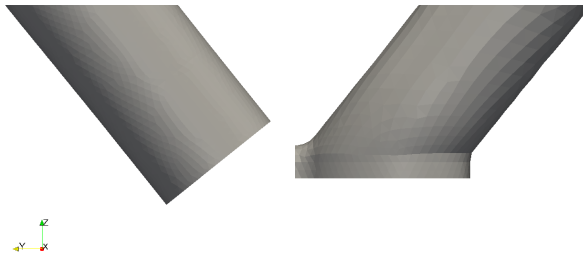


Figure 2.9: The left part shows the modified channel and the right part shows the original channel.

The combustion chamber is also modified. This modification was introduced to reduce the cell aspect ratio and eventually skewness. These properties will be further discussed in section 2.3.5. These issues occurred due to flat control volumes in the clearance, see section 1.3.1. This was solved by increasing the clearance by lowering the position of the piston at TDC. The length of the stroke is adjusted according to the increased clearance. This adjustment ensures that the position of BDC still corresponds to the specifications of the H576 engine.

The combustion chamber has also changed regarding the bowl in the top. A compatibility problem with GGI, General Grid Interface, and the `twoStrokeEngine`-library forced a reduction of the bowl since the bowl and the cylinder were merged with a GGI. The General Grid Interface is used to merge two volumes without changing the structure or topology of the meshes. The problem with the GGI is described in section 2.8.1. The entire geometry, with the modified scavenging channel and combustion chamber is shown in Figure 2.10.

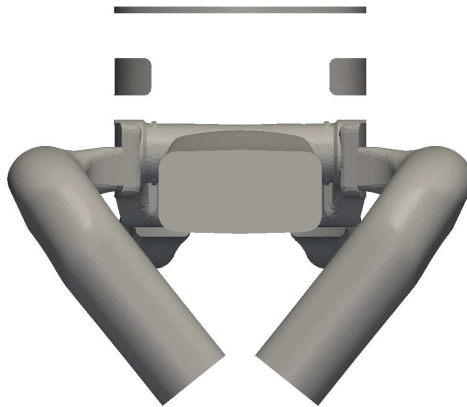


Figure 2.10: The modified geometry at 0 CAD.

2.2 Mesh generation

Two meshes are used in this work. One mesh of the modified real engine, the H576 engine, geometry and one simplified engine geometry mesh. The simplified mesh is used initially to get familiar with the required OpenFOAM features. The simplified geometry consists

of a number of blocks and is from now on referred to as the block geometry. This approach is discussed in section 1.4.

2.2.1 The H576 engine geometry

The solid geometry is created in the 3-dimensional computer aided design software CATIA. The fluid volumes must be extracted in order to get a geometry to work with. By extracting and converting the non-solid volumes to solid volumes the fluid geometry is created in CATIA. These volumes must now be divided into control volumes, i.e. creating a mesh, to be able to solve the governing equations using the finite volume method, see section 1.3.4 and 1.3.6.

The surfaces of the volumes are first meshed using the mesh software ANSA. The surface meshes are then used as bases when the volumes are meshed. The volume mesh is created in the mesh software T-GRID. The software extrudes the 2-dimensional surface mesh into a 3-dimensional mesh. The control volumes are extruded into the volume, so the fluid volume will be kept. This creates a layer of control volumes at the volume surface. The control volumes in this layer are distributed according to the surface mesh. From now on are the control volumes in the mesh referred to as cells. The cell vertices is from now on referred to as points. The cell faces in the outer boundary for a certain mesh belongs to patches. A boundary can consist of a number of different patches. A boundary face can only belong to one patch, hence the patches cannot overlap each other.

The mesh is converted to the OpenFOAM mesh format. The OpenFOAM `fluent3D-MeshToFoam`-utility is used to convert the mesh from Fluent to OpenFOAM format. The `fluent3DMeshToFoam`-utility keeps track of zone and patch names as well as the boundary types. The number and types of cells at the starting point are found in Table 2.1. The mesh is shown Figure 2.11.

Table 2.1: Cells at 0 CAD for the H576 engine geometry.

Cell type	Number of cells
Hexahedral.	8996
Tetrahedral.	355177
Pyramids.	525
Total	364698



Figure 2.11: An overview of the H576 mesh.

2.2.2 The block geometry

As the name reveals is this geometry created by blocks. It has the same functional parts as the H576 engine geometry. The parts are however represented by blocks. These blocks are constructed to be as similar to the real parts as possible. This means that measures and volumes are supposed to be as close to the original as possible. This mesh only consists of block shaped hexahedral control volumes. The block geometry is shown in Figure 2.12. The block geometry is constructed by the OpenFOAM `blockMesh` mesh tool, see [5].

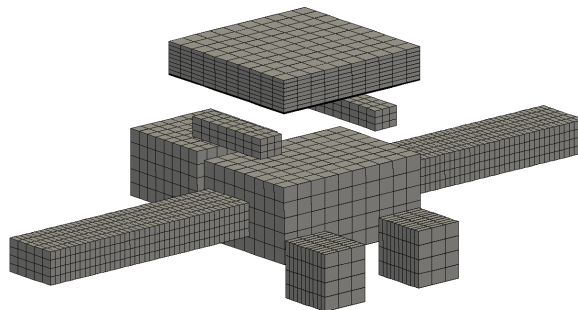


Figure 2.12: An overview of the block geometry mesh.

2.3 Dynamic mesh with topology changes

The two-stroke engine consists of moving parts. The parts will move when the engine is running. As a consequence the fluid volumes will move as well as deform. The mesh must adopt these changes and this requires motion of parts of the mesh as well as topology changes. A topology change implies a change in number of mesh control volumes. Topology changes are also changes in the number of control volume faces or a rearrangement of the control volumes. It is therefore required that the solver can perform the mesh operations needed for this kind of simulations. The required functions are described in this section. How OpenFOAM handles this kind of mesh operations is also described. To merge the

required mesh features and the solver for the governing equations, see section 1.3.5, the OpenFOAM-1.5-dev version is supplied with the `twoStrokeEngine`-classes.

2.3.1 Liner and ports

To accomplish the cylinder and port intersection some mesh definitions are required. The liner, and the piston pocket interfaces facing the ports are defined as a common patch named `liner`. The ports, i.e. exhaust port, intake port and stratifying ports, are defined as a common patch named `scavengingPort`.

2.3.2 Deforming mesh

The combustion chamber mesh is allowed to change in two manners as the piston is moving. The combustion chamber and ports interaction is not considered yet. The two modes in which the mesh is changing due to the piston motion are the `Mesh deformation mode` and `Piston Layering mode`. The latter is a topology change and will be described in section 2.3.3.

When the mesh is in deformation mode the topology is not changing. The topology remains constant since no cells are added and no cells are removed. The cell structure and distribution is not changing neither. The cells are either stretched or compressed. This mesh operation goes by the name morphing.

The piston is defined by some parameters required by the `twoStrokeEngine`-classes. These definitions are used to define which cells that will move as the piston moves. The points and the corresponding cells closest to the piston are defined, and in this work named `pistonPoints` and `pistonCells`. A set of internal faces named `pistonLayerFaces` is also required. Figure 2.13 and Figure 2.14 shows the `pistonPoints`, `pistonCells` and `pistonLayerFaces`.

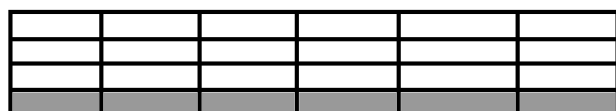


Figure 2.13: Grey cells and the corresponding points indicates piston cells and points.

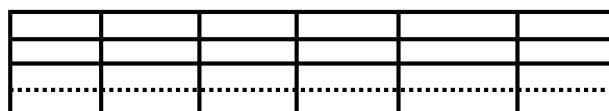


Figure 2.14: Dashed line indicates the set of faces named piston layer faces.

The `pistonPoints` and `pistonCells` are also required for the morphing of the cells together with two additional sets, the `headPoints` and `headCells`. The latter two sets defines points and cells which will remain fixed as the piston is moving. Figure 2.15 shows the `headCells` and `headPoints`. The `pistonCells` move with the piston speed and are not morphed. The cells between the piston and head cells are morphed as the piston moves.

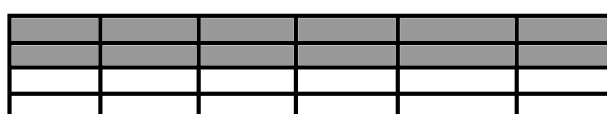


Figure 2.15: Grey cells and the corresponding points indicates head cells and points.

2.3.3 Layer addition and removal

The `Piston layering mode` is now introduced. This mode induces topology changes since layers of cells are added or removed.

The layer addition and removal is carried out according to the following algorithm. Two cell layers are defined, one static and one dynamic. These must have the same structure. The neighbouring faces of two layers must be normal to each other. The dashed line in Figure 2.14 shows such a set of faces. This requires a hexahedral mesh in the layering region.

The dynamic cell layer is named `pistonCells` and will move with the piston speed while the static cell layer is named `headCells`. The faces in the dynamic layer facing the static layer, upon which the layering will take place, is defined as `pistonLayerFaces`. See also section 2.3.2.

The thickness of the cell layer upon the `pistonLayerFaces` is approximated in the following manner. The volume of the cells in the active layer is divided by the area of the `pistonLayerFaces`. This ratio is defined as the cell layer thickness. The `pistonCells` will now move as the piston moves. The `pistonCells` will remain its topology and not morph. The active cells, the layer above the `pistonLayerFaces`, will start to morph in the direction of the piston velocity. For each morphing sequence, which is defined by the timestep, the cell layer thickness is calculated. During the expansion stroke the cells are growing. When the cell layer thickness reaches a certain limit, named `maxLayerThickness` in this work, the cell layer addition is triggered. When the cell layer addition is triggered, a new point will be added for every point in the `pistonLayerFaces`. A cell for every face in the `pistonLayerFaces` set is added, see figure 2.16 and 2.17. The cell faces in the new layer will be defined as internal or boundary faces depending on their location. Boundary faces in the new layer will belong to the patch in which they are inserted.

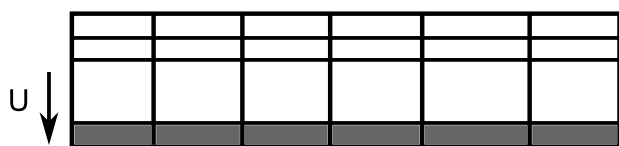


Figure 2.16: The piston cells are moving and the cell layer above is expanding.

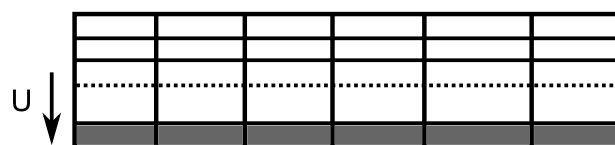


Figure 2.17: A new cell layer is introduced, dashed line.

During the compression stroke the cells in the active layer are compressed. The cell layer thickness is still determined as in the cell layer addition case. When the cells in the active layer reach a certain thickness, named the `minLayerThickness` in this work, a layer of cells will be removed, see figure 2.18 and 2.19. All the faces of the cells next to the `pistonCells` in the active layer will be removed, except those in the `pistonLayerFaces` set. The points that will be removed are replaced with the points in the `pistonLayerFaces`.

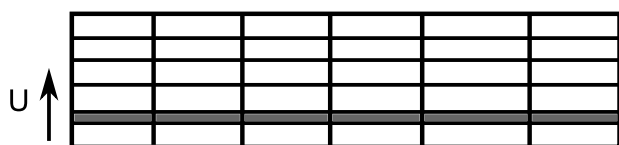


Figure 2.18: The grey cell layer is compressed.

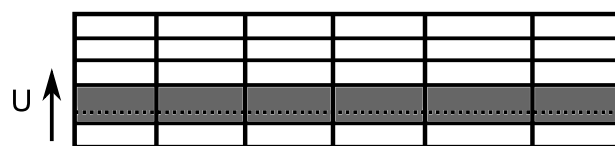


Figure 2.19: One cell layer removed, dashed line.

2.3.4 Sliding interfaces

It is obvious that at some instance of the stratified charged two-stroke engine cycle, described in section 1.3.3, some of the fluid volumes which are separate from each other will be connected. Since the engine works by repeating a cycle, volumes will consequently be disconnected from each other. The solver must be able to handle these volume interactions. In OpenFOAM a feature named `slidingInterface` copes with this kind of problem. The `slidingInterface` is used in the `twoStrokeEngine`-library to handle the interaction between the ports and the cylinder. The piston pockets are also concerned with this matter since they are also connecting to the ports.

The following algorithm is adapted for the sliding interfaces in OpenFOAM. One patch will act as master patch and the other as a slave patch. In this application the port interfaces will be the master patches named `scavengingPort`. The cylinder liner and piston pocket interfaces are the slave patches, named `liner`. These patches are shown in Figure 2.20 and Figure 2.21

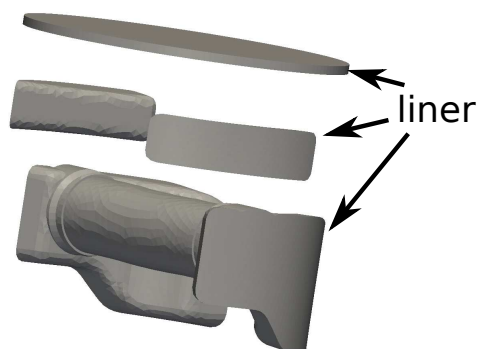


Figure 2.20: The liner patch for sliding interface is indicated.

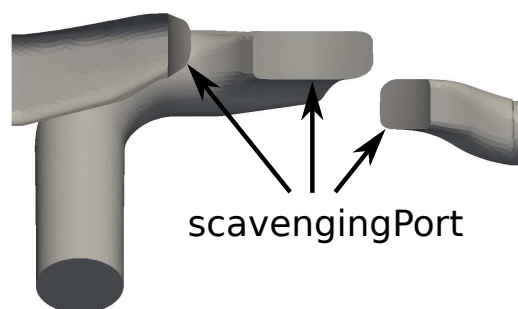


Figure 2.21: The scavenging patch for sliding interface is indicated.

As the two patches starts to overlap each other the sliding interface algorithm starts. First the slave points are projected onto the master patch. The projection is in the normal direction to the patches. For every point in the master and slave patch a tolerance is set. The tolerance is based on the fraction of a constant and the length of the shortest edge connected to the point. When a slave point is projected onto a master face the distance between the slave point and points on the master face is evaluated. If some distance lies within the tolerances for the actual points on both the master and slave face these points are merged. The master point will remain its position while the slave point will move. For slave points which these conditions are not fulfilled, hence are not yet merged, another condition is evaluated. If the distance to the nearest master edge is smaller than the tolerance the point will move to this edge. The points and faces which are merged due to sliding interface will from now on not belong to any patch. These will be internal faces. Figure 2.22 and Figure 2.23 visualises two sliding interfaces, a scavenging port and the exhaust port respectively.

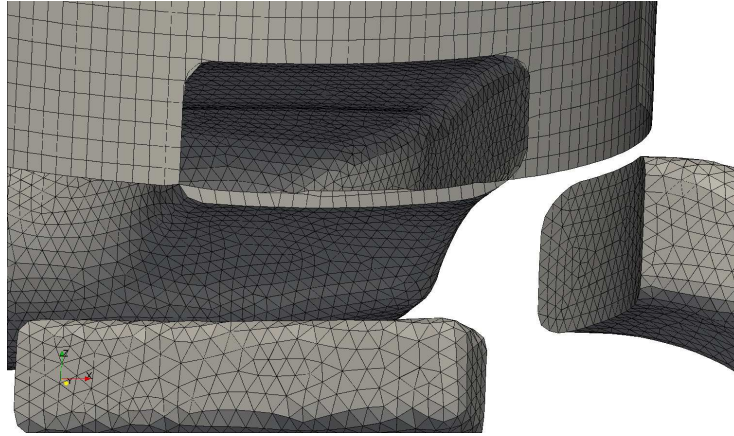


Figure 2.22: The scavenging port sliding interface at BDC.

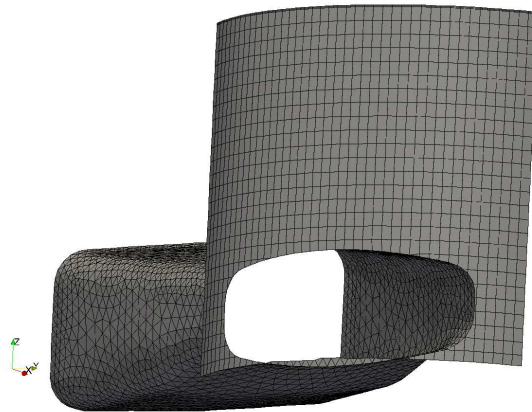


Figure 2.23: The exhaust port sliding interface at BDC.

2.3.5 Mesh quality

The mesh can be judged by some quality measures. The aspect ratio, skewness and non-orthogonality are used to measure the mesh quality. Since it is a dynamic mesh with a changing topology the quality of the mesh will vary.

The aspect ratio is determined by how square a cell is. An aspect ratio of 1 represents a square cell. The maximum aspect ratio is found in figure 2.24. Recommended threshold value is 1000 according to [5].

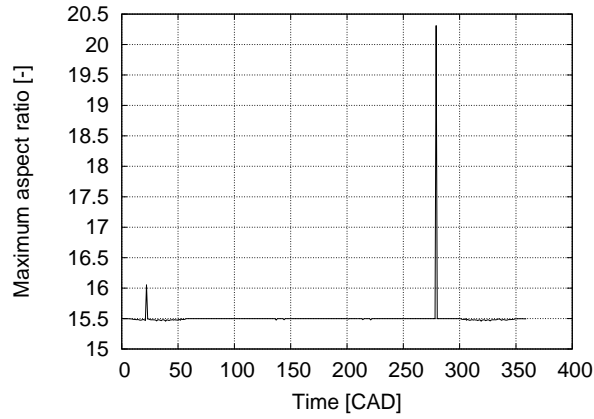


Figure 2.24: The maximum cell aspect ratio for the entire mesh for one cycle.

The maximum skewness is found in figure 2.26. The skewness, σ is defined in equation (2.1). See figure 2.25. The recommended threshold value is 4 according to [5]. Skewness might be a source to instability especially in combination with high non-orthogonality, see [4].

$$\sigma = \frac{|\mathbf{d}|^2}{\mathbf{c} \bullet \mathbf{d}} \quad (2.1)$$

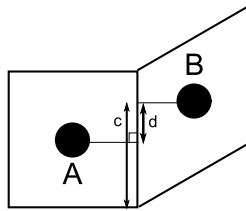


Figure 2.25: Defintion of cell skewness.

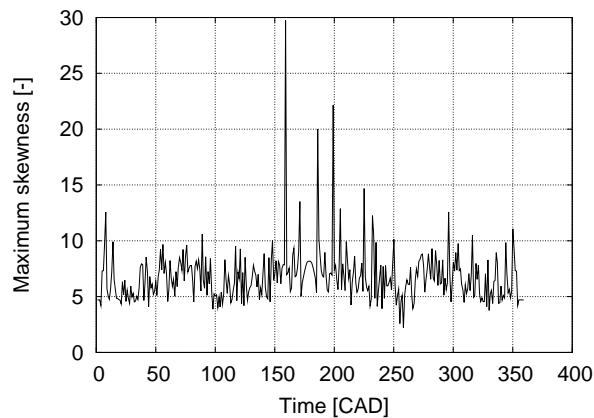


Figure 2.26: The maximum cell skewness for the entire mesh for one cycle.

The maximum non-orthogonality is found in figure 2.28. It is defined as the angle, α , between two vectors, \overrightarrow{AC} and \overrightarrow{AB} . Where \overrightarrow{AC} is normal to the face of the cell. While \overrightarrow{AB} is

the vector from cell center A to the neighbouring cell center B . See figure 2.27. Since the mesh shows quite high non-orthogonality, non-orthogonal correction steps is recommended in the PISO algorithm, see section 1.3.5 and [5]. Care must also be taken when choosing discretisation scheme for laplacian terms in the governing equations, see section 2.5.4. Non-orthogonality above the threshold value will have significant effect on the stability, see [4]. The recommended threshold value according to [5] is 65 degrees.

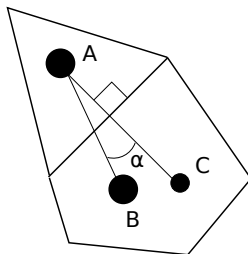


Figure 2.27: Definition of non-orthogonality.

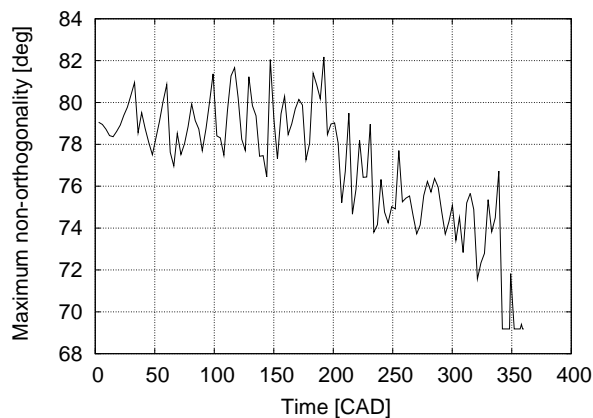


Figure 2.28: Maximum non-orthogonality for the entire mesh for one cycle.

2.4 Indata from 1-D software

To set proper boundary conditions at the inlets and outlets relevant information about these are required. From 1-D software simulations indata regarding pressure, temperature and massflow is obtained. Data for these properties yields at the inlets and outlets. Pressure and temperature curves for the stratifying inlets are shown in figure 2.29 and 2.30. Corresponding data for the intake inlet and exhaust outlet are found in 2.31, 2.32, 2.33 and 2.34. The cylinder pressure is shown in figure 2.41.

Volumes of channels for the geometry used in 1-D simulations are found in table 2.2. Corresponding volumes from current engine geometry are also found in this table as well as the trapped compression ratio. The following setup was used for the 1-D simulation. The Single Wiebe combustion model with Wiebe coefficients $a=5$ and $b=1.2$ is used and a complete combustion is assumed. Ignition starts 28 CAD BTDC and the combustion duration is 65 CAD. The air fuel ratio is 0.85.

Table 2.2: Volumes of 1-D and 3-D geometry.

	1-D geometry	3-D geometry
0.5cm Scavenging channel volume, [cm ³].	11.687	18.670
Exhaust channel volume, [cm ³].	16.604	13.397
Stratifying channel volume, [cm ³].	2.7489	2.640

2.5 Discretisation schemes

The governing equations are discretised over each control volume in the domain, see section 1.3.4, 1.3.5 and 1.3.6. Depending on the type, the equation terms are discretised in different manners. The discretisation schemes used are found in this section. A brief description is given for each scheme. Further information about the discretisation schemes is found in [7] and [4].

2.5.1 Time derivatives

An example of the time derivative of a variable, ϕ , is shown in equation (2.2). Equation (2.2) is discretised using the implicit Euler scheme, which is a first-order accurate and bounded scheme. Since the scheme is bounded, the value of ϕ will not be larger than the values of the same variable used to calculate it. It will not be smaller than the used values either. Equation (2.3) shows equation (2.2) discretised by the implicit Euler scheme. The scheme is implicit since $\phi(t + \delta t)$ is ϕ at the current time step.

$$\frac{\partial \phi}{\partial t} \quad (2.2)$$

$$\frac{\partial \phi}{\partial t} = \frac{\phi(t + \delta t) - \phi(t)}{\delta t} \quad (2.3)$$

2.5.2 Gradient terms

Gradients, see equation (2.4), are discretised using the central-differencing scheme. It is a second-order accurate interpolation scheme. The diffusion term of the momentum equation in one dimension is discretised in equation (2.5) as an example. In equation (2.5) is U the velocity and μ the viscosity. The central-differencing scheme is also used in section 1.3.6.

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) \quad (2.4)$$

$$\int_w^e \frac{\partial}{\partial x} \left(\mu \frac{\partial U}{\partial x} \right) dx = \left(\mu \frac{\partial U}{\partial x} \right)_e - \left(\mu \frac{\partial U}{\partial x} \right)_w = \mu_e \frac{U_E - U_P}{\delta x_e} - \mu_w \frac{U_P - U_W}{\delta x_w} \quad (2.5)$$

2.5.3 Divergence terms

Divergence terms, equation (2.6), are discretised using the first-order upwind scheme. Since the finite volume method implies an integration over the control volume the Gauss divergence theorem, shown in equation (2.7), is applied. The first-order upwind scheme determines the value of an variable, ϕ , at the cell face depending on the flow direction. For example, ϕ at the east face is set to the node value ϕ_P if the flow across the face is zero or positive in the x-direction. Otherwise the east neighbour node value, ϕ_E , is used. The

scheme is defined as transportive since it is based on the direction of the flow. See equation (2.8) which shows the integration in the x-direction of x-part for ϕ . Figure 1.11 in section 1.3.6 is also recommended.

$$\operatorname{div}(\mathbf{F}) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \quad (2.6)$$

$$\int_V \operatorname{div}(\mathbf{F}) dV = \int_A \mathbf{F} \bullet \mathbf{n} dA \quad (2.7)$$

$$\int_w^e \frac{\partial \phi}{\partial x} dx = \phi_e - \phi_w$$

$$\phi_e = \begin{cases} \phi_P, & \text{if } U \geq 0 \\ \phi_E, & \text{if } U < 0 \end{cases}$$

$$\phi_w = \begin{cases} \phi_P, & \text{if } U \leq 0 \\ \phi_W, & \text{if } U > 0 \end{cases} \quad (2.8)$$

2.5.4 Laplacian terms

Laplacian terms,(2.9), are discretised using a blended scheme. The scheme is a blend of the central-differencing scheme and the first order upwind scheme. The central differencing and the first order upwind schemes are described in section 2.5.2 and section 2.5.3 respectively. The blend of each part is defined by a number, γ . This is shown in equation (2.10) for a variable ϕ where CD denotes the part discretised by the central differencing scheme and FU denotes the part discretised by the first order upwind scheme. This scheme is to prefer when a mesh is non-orthogonal. Since the mesh in this work is non-orthogonal, see section 2.3.5, the variable γ is set to 0.333. The result of this is an increase in the transportiveness of the scheme, at the cost of a lower accuracy.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (2.9)$$

$$\phi = \phi_{CD}\gamma + \phi_{FU}(1 - \gamma) \quad (2.10)$$

2.6 Boundary conditions

In this work the finite volume method is used to solve the governing equations, see section 1.3.4-1.3.6. This method requires so called boundary conditions. A boundary condition for a variable is either a fixed value or a derivative which is known at the mesh boundary. In this section the boundary conditions used in this work are presented. The sections is structured according to the boundary types.

2.6.1 Walls

The boundaries that are defined as walls are the following. The liner, the scavenging channel walls, the stratifying channel walls, the exhaust channel walls, the cylinder head, and the piston, all these boundaries are walls.

The walls are assumed to be thermally isolated. Physically this means that no heat is transferred from the fluid to the wall. This is accomplished by setting the homogenous

Neumann boundary condition at the walls for temperature, T [K]. The homogenous Neumann is from now on denoted by the acronym HN. The HN boundary condition implies zero temperature gradient across the wall. The HN boundary condition for temperature is shown in equation (2.11), where n is the wall normal direction. This is met by setting the boundary node temperature equal to the the temperature in the neighbouring inner node in the equation system of the governing equations.

$$\frac{\partial T}{\partial x_n} = 0 \quad (2.11)$$

Regarding the pressure, the boundary nodes in the walls are not subjected to any pressure. Since the wall is fixed and will not deform there will be no pressure gradient across the wall. Consequently the pressure at the walls is set as a HN boundary condition.

For the velocity the no slip condition is assumed. This means that the velocity is zero in the three directions at the wall. The velocity vector at the walls is shown in equation (2.12). The boundary condition, where the value of the current variable is set, is called the Dirichlet boundary condition. The acronym D will now be used as an acronym for Dirichlet.

$$\mathbf{U} = (0, 0, 0) \quad (2.12)$$

Since the piston and piston pockets are not fixed the piston velocity must be taken into account on these walls. There should still be no mass flux across these walls.

The turbulence modelling properties, k and ε , are defined by the wall functions at the walls. How these are defined is described in section 1.3.5.

2.6.2 Inlets and outlets

This section will present the boundary conditions at the inlets and outlets. The scavenging channel inlet, the stratifying channel inlet and the exhaust outlet boundary conditions will all be described in this section. The reason for this common heading is the behavior of the flow at these boundaries. All of these boundaries experiences both flow into the domain and flow out of the domain in the simulations. This requires some extra attention.

If the inlets where just inlets, i.e. the flow direction would only be into the domain, the D boundary condition would be set at the inlets for the temperature. The temperature would then be set to a defined value on the flow going into the domain. The D boundary condition must not be set if the flow is going of the domain. The physical intpretation of this is that the defined temperature is forced on to the flow going out of the domain.

If the outlet was a pure outlet, meaning that the flow was only going out of the domain, the HN boundary condition is set for the temperature. This would result in the same temperature in the outlet as in the cells immediately inside of the outlet. The exhaust outlet in this case is however experiencing flow into the domain at some instances. If then the HN boundary condition is set the temperature behavior will not be transportively represented. This might result in a heat flux out of the domain despite a convection dominated flux into the domain. In the case of flow into the domain the D boundary condition should be set, as mentioned previously in this section.

To handle the issue of the flows going in and out of the same boundary an adaptive boundary condition is used. If the flow is going into the domain the D-condition is set. If the flow is going out of the domain the HN-condition is set. This adaptive boundary condition is set for the temperature at the inlets and outlets. The temperature that is set when the flow is going into domain is the mean value of the temperature found in the indata from 1-dimensional simulations for the corresponding boundary. The temperatures

from the 1-D simulations for the inlets and outlets are found in Figure 2.30, 2.32 and 2.34.

The pressure at the inlets is set as a D boundary condition. At the inlets and outlets the static pressure is set. The static pressure, p_{stat} is found in equation 2.13. The total pressure p_{tot} is found from the 1-D simulation in data. The total pressures for the inlets and outlet are found in Figure 2.29, 2.31 and 2.33.

$$p_{stat} = p_{tot} - \frac{\rho U^2}{2} \quad (2.13)$$

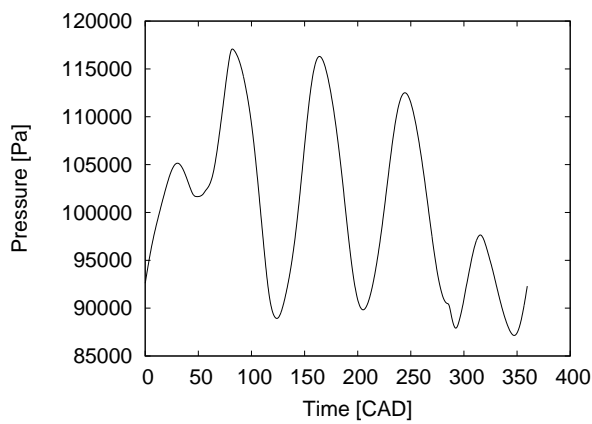


Figure 2.29: Pressure in the stratifying channel inlet versus crank angle degrees.

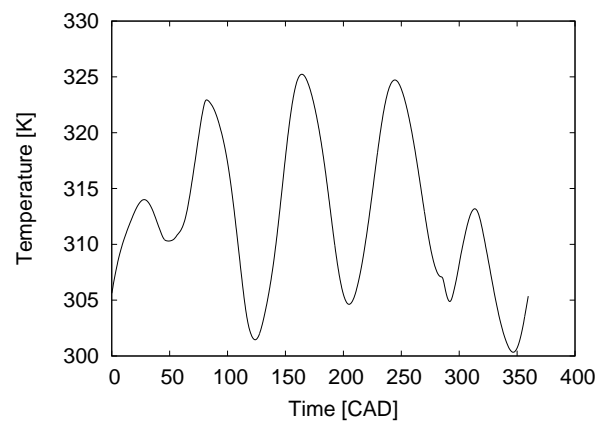


Figure 2.30: Temperature in the stratifying channel inlet versus crank angle degrees.

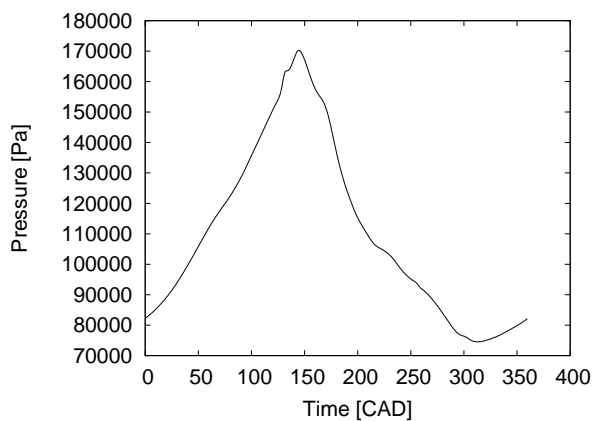


Figure 2.31: Pressure in the scavenging channel inlet versus crank angle degrees.

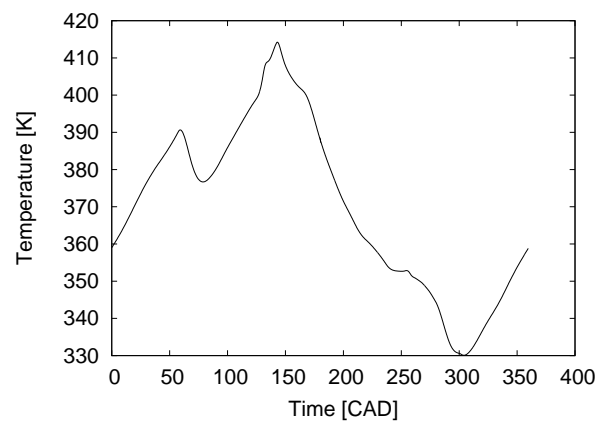


Figure 2.32: Temperature in the scavenging channel inlet versus crank angle degrees.

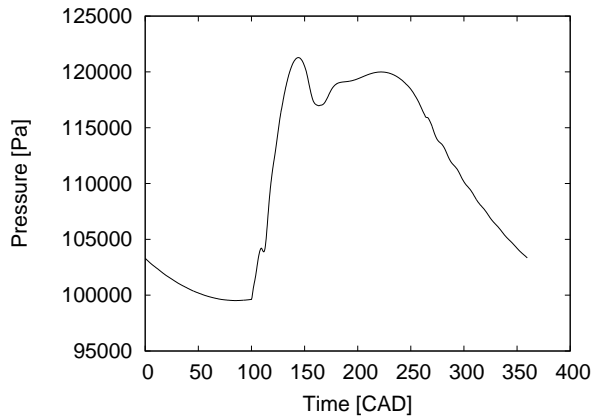


Figure 2.33: Pressure in the exhaust outlet versus crank angle degrees.

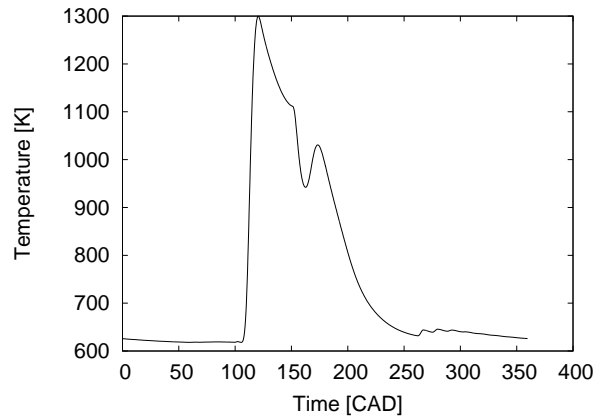


Figure 2.34: Temperature in the exhaust outlet versus crank angle degrees.

At the inlets and outlets the velocity is set in a similar manner as the temperature with the adaptive boundary condition. This means that when the flow is directed out of the domain the HN condition is set and when the flow is directed into the domain the D condition is set. The velocity which is set when the D condition is active depends on the pressure. The velocity is set in the inlet, (or outlet), normal direction.

The adaptive boundary condition used for the temperature is also used for the turbulence modelling properties, namely k and ε , at the inlets and outlets. The values for the D condition are small, yet finite, values. This is based on the assumption that the flow into the domain is not turbulent. The k and ε values are set to 10^{-4} for the flow into the domain.

According to section 1.3.7 a passive scalar is used to represent the air fuel mixture. The air fuel mixture enters the domain through the scavenging channel inlet. This is resolved by setting the adaptive boundary condition, which uses the D and HN condition, at the scavenging channel inlet. The value of the scalar is set to 1 which implies a 100 percent concentration of air fuel mixture in the flow going into the domain. When backflow occurs in the scavenging channel inlet the HN condition is set, this is similar as for the temperature. The stratifying channel inlet and the exhaust outlet uses the same boundary condition. The difference is that the concentration is set to zero percent when the flow is directed into the domain.

2.7 Time step and convergence criterion

When the simulation starts the solver starts to solve the governing equations. The solution will either converge or diverge. If the solution diverge the simulation stops and the source of divergence must be found. If the solution converge according to the convergence criterion, (the convergence criterion will be described later in this section), the solver will take a step forward in time. How large the time step, δt , will be is determined by calculating the Courant number, Co [-]. The definition of the Courant number is found in equation, (2.14), where δx is the length of the cell in a certain direction, and U the fluid velocity in the same direction. The physical interpretation of this is the fraction of the cell length a fluid particle will travel during one time step, δt . When determining a time step a Courant number is set while the cell lengths are known from the mesh and the velocities are known from the recently converged solution. Now δt is calculated for every cell in the domain. The minimum δt found is set as the new time step. It is recommended to have a Courant

number smaller than 1 for stability reasons, see equation (2.15).

$$Co = \frac{\delta t |\mathbf{U}|}{\delta x} \quad (2.14)$$

$$\delta t < \frac{\delta x}{|\mathbf{U}|} \quad (2.15)$$

The solution is considered converged when the convergence criteria for each variable is fulfilled. The Gauss-Seidel iteration method, described in section 1.3.6, is used in this work with one exception. The passive scalar requires a iteration method which handles asymmetric matrices. The convergence criterion for a variable is based on the residuals from each iteration. How the residuals are calculated is described in section 1.3.6. The convergence criteria are found in Table 2.3. More information about the time step and how the residuals are calculated is found in [7] and [4].

Table 2.3: The residual levels.

	Residual level
Continuity equation, ρ .	10^{-9}
Momentum equations, U .	10^{-9}
Energy equation, h .	10^{-11}
k -equation	10^{-9}
ε -equation	10^{-9}
Pressure correction equation, p .	10^{-8}
Passive scalar transport equation, θ	10^{-5}

2.8 Setup for calculations in parallel

In order reduce the time of the calculations the simulation is run using multiple computer processors. An acronym for a computer processor is CPU, which means central processing units. A simulation run using one CPU is called a serial run. When multiple CPU:s are used the a simulation is run in parallel.

To be able to run a simulation in parallel the software, more specifically the solver, must be prepared for this feature. The computational domain must be decomposed in a number of parts corresponding to the number of CPU:s that will be used. When the domain is decomposed the simulations can be run. Each CPU now has a case for which the simulation will be run. For the parallel simulation to be useful the results must be conform with the results of the same case run in serial. Hence the CPU:s cannot work independtly of each other. The computational domain must still be continous as in the serial case. The information from each CPU be transferred across the boundaries of the parts. This means that one CPU cannot solve for one time step while another CPU solves for another time step since the over all domain is mutual. Consequently the CPU with the highest load will set the computational speed of the solution. Therefore it is preferred to decompose the computational domain in such manner that the weight on each CPU is similar. The `twoStrokeEngine`-libraries have some requirements on the domain decomposition to work in parallel.

2.8.1 Layer addition and removal in parallel

As mentioned in the previous section, see section 2.8, care must be taken when doing the decomposition of the domain. The zone where the layer addition and removal, see section

2.3.3, must be decomposed in certain manner. A part of this zone must be present and contain some cells, points and faces on all of the CPU:s. If a part of the zone is not present the solver will not be able to perform the mesh operations on the that current CPU and the simulation will not run. In a physical point of view this means that the combustion chamber must be decomposed in the x- and y-direction. It must not be decomposed in the z-direction.

Due to these requirements a GGI, General Grid Interface), is not possible on top of the head cells when running in parallel. The piston layering zone must be present on all processors while the patches that should be connected with a GGI must be on the same processor, which is an obvious conflict.

2.8.2 Sliding interfaces in parallel

The sliding interfaces feature has some similar demands on the decomposition as the layer addition and removal feature, see section 2.8.1. For the solver to handle the mesh operations regarding the interaction between the cylinder and the ports, see section 2.3.4, some precautions must be taken. Every CPU must have a share of the `liner-` and `scavengingPort-` patches. This is similar as for the layer addition and removal zone. An additional requirement must also be fulfilled for the solvers mesh operation to work. An interface where the sliding mesh operation takes place must not be decomposed and put on several CPU:s. Physically this means that the exhaust port and its corresponding area on the liner must be kept intact on one CPU. The same requirement yields for the scavenging ports, the stratifying ports and the piston pockets.

2.8.3 Domain decomposition

To run the case in parallel the domain must be decomposed. The decomposition must be done according to some demands for the mesh operations, and thus the solver, to work. These demands are described in section 2.8.1 and 2.8.2. According to these demands and the engine geometry there is basically one way to do the decomposition. The domain is decomposed in three parts and the simulation can thus be run on three CPU:s.

The current domain is decomposed in three sections. Section one, I, is found in figure 2.35 and 2.36. Section two, II, is found in figure 2.37 and 2.38. Section three, III, is found in figure 2.39 and 2.40. By having a look at these parts it makes clear that the demands for the decomposition are fulfilled. It is also obvious that this is more or less the only way to do the decomposition and still put a rather equal load on the processors.



Figure 2.35: Top view of section I.

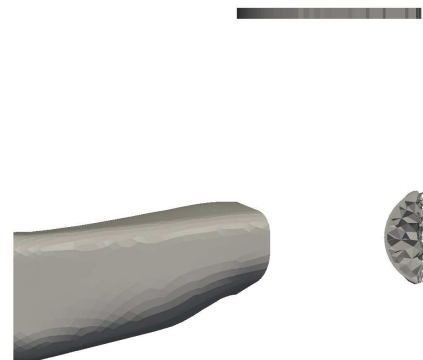


Figure 2.36: Side view of section I.

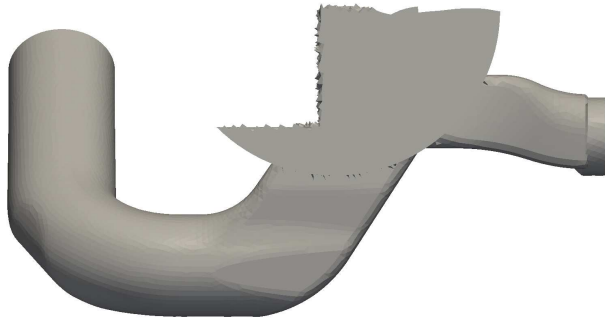


Figure 2.37: Top view of section II.

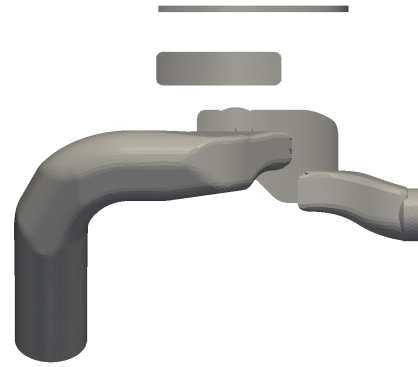


Figure 2.38: Side view of section II.

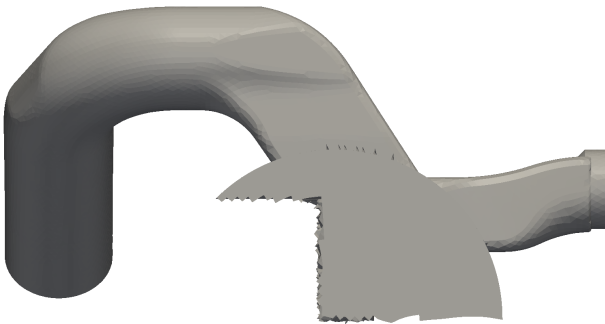


Figure 2.39: Top view of section III.

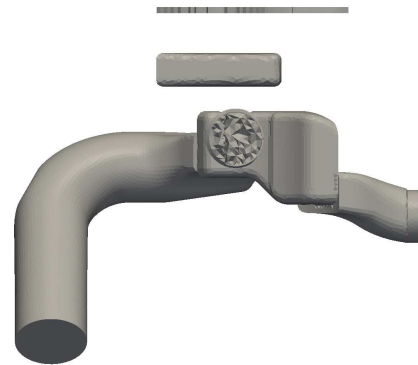


Figure 2.40: Side view of section III.

To determine the time gained by running the simulation in parallel a test was carried out. This test was also done to get an idea of how much of the simulation time is spent on the mesh operations. The test is carried out in the following manner. The mesh operations were first tested. The mesh motion was carried out without solving any of the governing equations, see section 1.3.5. A fixed time step of 0.05 CAD was used, the start time was 0 CAD and the end time 1 CAD. The test was performed with the OpenFOAM `checkMesh`-option switch on as well as switch off. The `checkMesh` utility performs a quality and overall mesh survey at every time step. No data is written to disk during the run. This was run both in serial and in parallel, (with the previously mentioned decomposition).

A new test was run with the same start time, stop time and time step as the mesh operations test. In this case the solver did not only perform the mesh operations but also solved the governing equations. These tests were done with the `checkMesh`-option switched off and the results are shown in Table 2.4 and Table 2.5.

Table 2.4: Parallel and serial run comparison.

Test setup	Time for 3 CPU:s, [s]	Time for 1 CPU, [s]
Mesh operations only.	82	175
Mesh operations + checkMesh	122	253
Mesh operations and CFD.	213	387

Table 2.5: Parallel and serial run comparison.

Test setup	Time fraction for 1 and 3 CPU:s, [-]
Mesh operations only.	2.1
Mesh operations + checkMesh	2.1
Mesh operations and CFD.	1.8

2.9 Assumptions and approximations

The assumptions and approximations used in this work are summarised in this section. This section also presents some approximations which have not yet been presented.

2.9.1 Fluids and combustion

In this work the two-stroke engine scavenging is simulated. The flow is defined as cold which means that no combustion will take place. Regarding the fluid, only one fluid is used namely air. As a consequence no chemical reactions will occur. Despite the lack of combustion and chemical reactions the flow is supposed to act as if it has occurred. This is solved by an approximation of the combustion. The combustion is approximated by setting a pressure and temperature in the cylinder just before the exhaust port is about to open. The cylinder pressure is obtained from the 1-D simulation indata. See Figure 2.41 where the calculated cylinder pressure from the 1-d simulations is shown. The pressure found at the CAD where the exhaust port is about to open is set in the cylinder at the same CAD. In a similar manner is the temperature set. However is the temperature in the cylinder not available from the 1-D indata. The temperature in the exhaust outlet is given in the 1-d indata. The maximum temperature in the exhaust outlet from the 1-D indata, see Figure 2.34 in section 2.6.2, simulations is set in the cylinder just before the exhaust port is about to open.

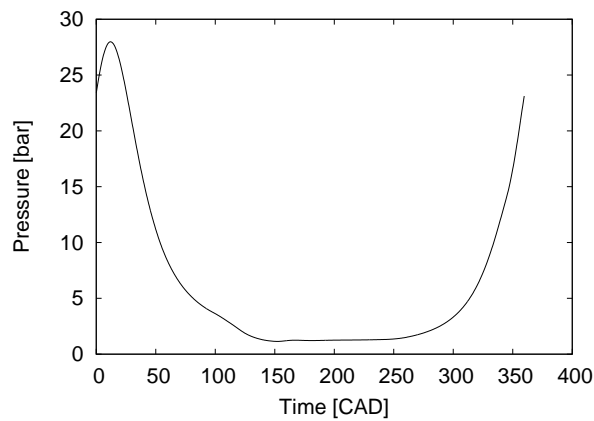


Figure 2.41: Pressure cylinder versus crank angles.

2.9.2 Initial field values

A good initial guess of the internal field values can reduce the computational time for the first step since a lower number of iterations is required. If the initial field values are set in a major discrepancy according to the boundary conditions it might also be hard to reach convergence for the first time step. The initial pressure and temperature fields are set

separately for each volume, i.e. the exhaust channel has one temperature and pressure while the scavenging channel fields are set to other values. The inlet and outlet temperatures and pressures are known from the 1-D indata. The initial pressure and temperature fields in a certain volume are thus set equal to the values at the volume inlets (or outlets) at the start position CAD. E.g. for the scavenging channel inlet the 1-D indata temperature is A [K]. The initial temperature field of the scavenging channel is then set to A [K].

3 Results

The results of the work is presented in this section. First the results regarding the scavenging performance of the H576 engine are presented. The simulations and the OpenFOAM software is then validated. The 1-d simulations and simulations from a commercial 3-d software Fluent are used for validation.

3.1 Scavenging and trapping efficiency

The engine geometry is axisymmetric over the x-axis. It is therefore of interest to study the behavior at the scavenging channel inlets and stratifying channel inlets. It is found that the differences in massflow is negligible at these inlets. A comparison of the massflow through the left and right scavenging channel inlets are shown in Figure 3.1. A similar comparison for the stratifying channel inlets is shown in Figure 3.2. As can be seen in the difference is negligible. From now on will only one of each inlet type be used when results are presented. The right scavenging channel inlet and the right stratifying channel inlet will be used.

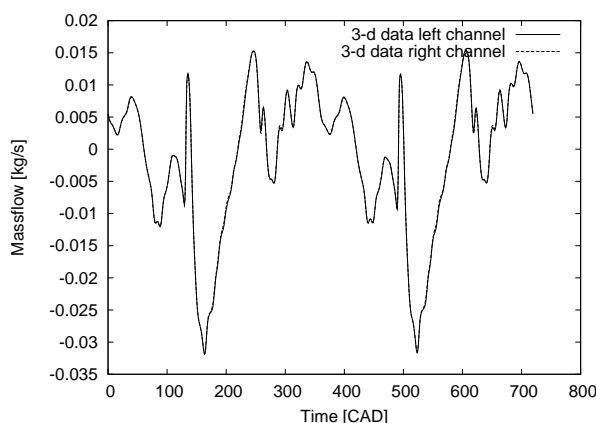


Figure 3.1: Comparison of massflow in the left and right scavenging channels.

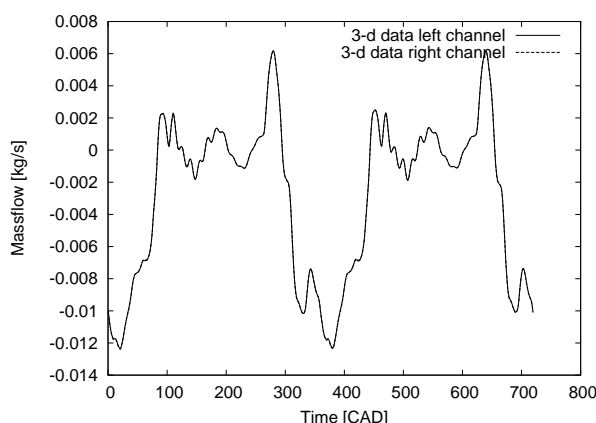


Figure 3.2: Comparison of massflow in the left and right stratifying channels.

The trapping efficiency is calculated as described in section 1.3.3. Using this method and according to the assumptions approximations the trapping could be determined. When

the solution is considered converged, the trapping efficiency was found to be approximately 93 %. To visualise the flow of the air fuel mixture passive scalar is used. The following series of figures shows the passive scalar concentration throughout a cycle, i.e. one revolution of the crankshaft. The angle of view is shown in Figure 3.3.

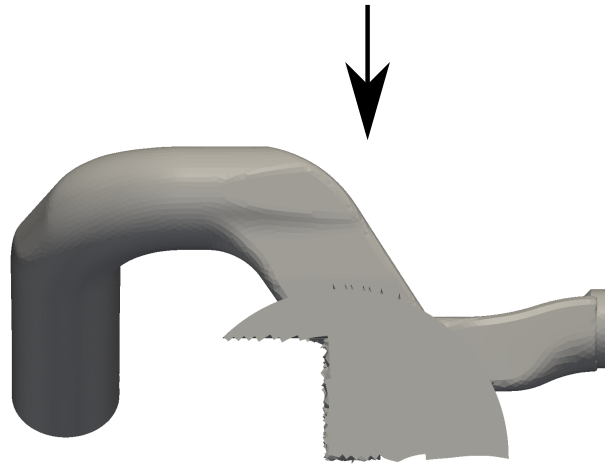


Figure 3.3: The angle of view for the visualisation of the passive scalar

In Figure 3.4 the piston is at TDC and the expansion stroke is about to begin. The passive scalar concentration is highly visible in the scavenging channel. The pass between the stratifying port and scavenging port is open.

A couple of crank angle degrees later, namely at 50 CAD, the expansion stroke has begun, see Figure 3.5. The pass is still open for the fresh air to enter the scavenging channel through the piston. It is obvious that the passive scalar has been suppressed in the scavenging channel. The passive scalar front is now close to the crankcase, compare with Figure 3.4.



Figure 3.4: The passive scalar is visualised at 0 CAD.

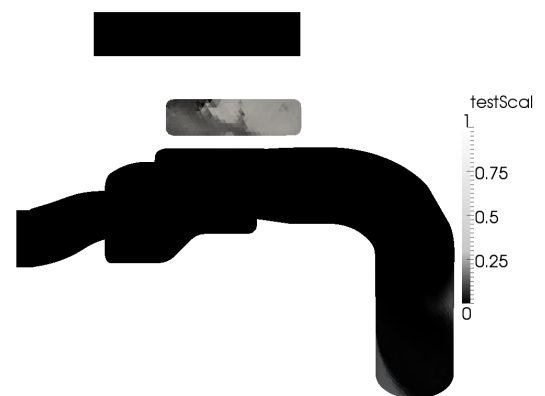


Figure 3.5: The passive scalar is visualised at 50 CAD.

In Figure 3.6 the connection between the scavenging channel and the stratifying channel has just been broken. The air fuel mixture from the crankcase is thus no longer suppressed by the fresh air from the stratifying channel. A comparison between Figure 3.5 and Figure 3.6 shows that the air fuel mixture (the passive scalar) front has begun to travel up the

scavenging channel.

Figure 3.7 shows how the passive scalar front has travelled further up the scavenging channel towards the scavenging port. At this moment, 101 CAD, the exhaust port is just about to open.



Figure 3.6: The passive scalar is visualised at 75 CAD.



Figure 3.7: The passive scalar is visualised at 101 CAD.

The scavenging port has just started to open in Figure 3.8. Initially the pressure and temperature in the combustion chamber results in a flow from the cylinder into the scavenging channel.

However a couple of CAD after the scavenging port has opened the flow has changed direction and the fresh air and the air fuel mixture is accelerated in the direction of the scavenging port. This is obvious when comparing the location of the passive scalar front in Figure 3.8 and 3.9.

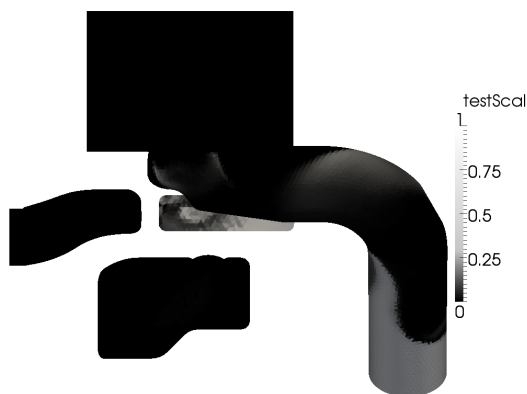


Figure 3.8: The passive scalar is visualised at 120 CAD.



Figure 3.9: The passive scalar is visualised at 160 CAD.

The piston has now reached BDC, see Figure 3.10. The air fuel mixture, the passive scalar, has now begun to enter the cylinder. At this stage the piston changes direction of motion and the compression stroke begins.

The compression stroke has now begun, see Figure 3.11. The air fuel mixture has now really begun to flow into the cylinder. Almost the entire scavenging channel is now filled with air fuel mixture. The eddy that can be seen in the cylinder in Figure 3.10 has evolved in Figure 3.11.



Figure 3.10: The passive scalar is visualised at 180 CAD.

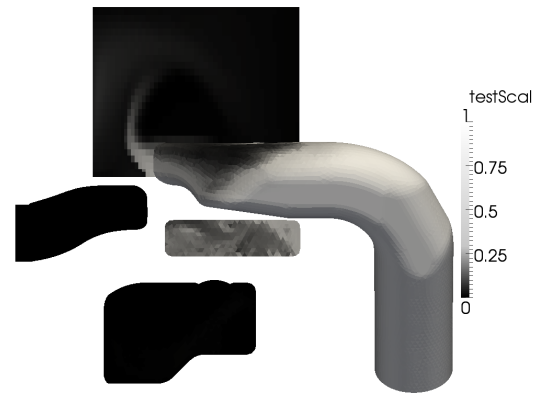


Figure 3.11: The passive scalar is visualised at 200 CAD.

In Figure 3.12 the scavenging port is just about to close, about 240 CAD. The eddy shown in the cylinder has now evolved further. A comparison can be made to Figure 3.11. At 259 CAD the exhaust port has also closed, see Figure 3.13.

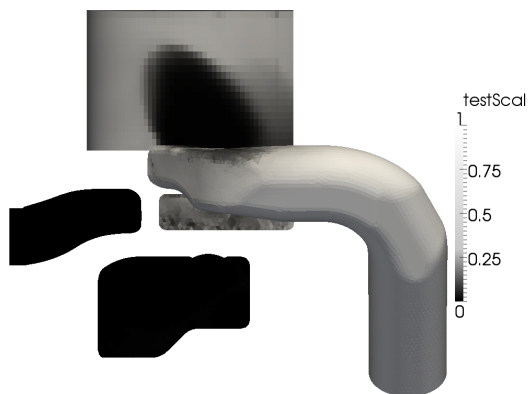


Figure 3.12: The passive scalar is visualised at 240 CAD.

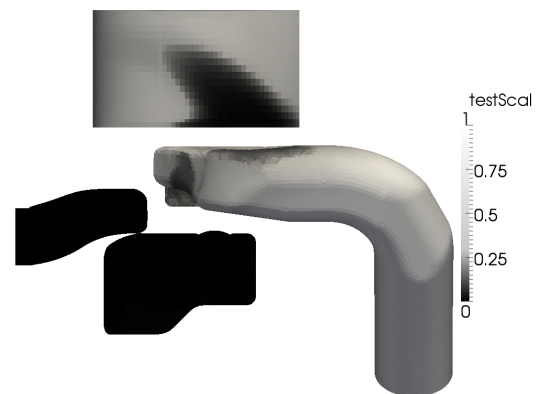


Figure 3.13: The passive scalar is visualised at 259 CAD.

The compression stroke continues and none of the ports are now open. The pass between the scavenging port and the stratifying port is not yet open. A comparison of the scavenging channel in Figure 3.13 and 3.14 shows that not much has happened regarding the passive scalar.

However in Figure 3.15 the pass has opened between the scavenging channel and the stratifying channel. A front of fresh air has begun to suppress the air fuel mixture, (passive scalar).



Figure 3.14: The passive scalar is visualised at 285 CAD.

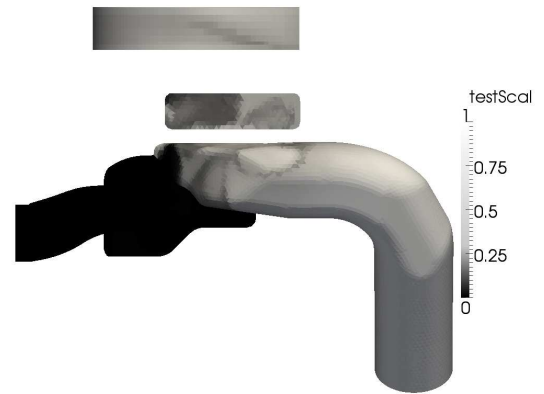


Figure 3.15: The passive scalar is visualised at 310 CAD.

The piston has now reached TDC. The pass through the piston has been open for a while and the passive scalar has been further suppressed in the scavenging channel which is shown in Figure 3.16. The crankshaft has now turned one revolution and the cycle is completed. The piston is now at TDC which is the same position as in Figure 3.4. A comparison of Figure 3.4 and Figure 3.16 shows that the passive scalar has the same starting point as endpoint.



Figure 3.16: The passive scalar is visualised at 360 CAD.

The flow of air fuel mixture, the passive scalar, through the exhaust port and exhaust channel will now be visualised. The following figures shows a plane in center of the exhaust channel together with a part of the cylinder. In Figure 3.17, the exhaust port is just about to open. It can be seen that there still are residuals of the passive scalar in the exhaust channel from the previous iteration. The passive scalar has not yet begun to flow into the cylinder since the scavenging port is not open.

The exhaust port has no begun to open, see Figure 3.18. Due to the pressure gradient between the cylinder and the exhaust channel the residuals of passive scalar in the exhaust channel is flushed out over the exhaust outlet.

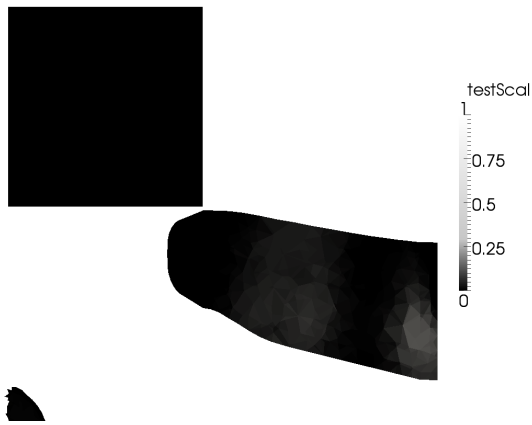


Figure 3.17: The passive scalar visualised in the exhaust channel at 101 CAD.



Figure 3.18: The passive scalar visualised in the exhaust channel at 120 CAD.

The scavenging port has been open for a while but the passive scalar has not really begun to flow into the cylinder, see Figure 3.19. Compare this with Figure 3.10 which shows the scavenging channel and port at the same CAD.

The piston has now reached BDC and passive scalar has begun to enter the cylinder, see Figure 3.20. No substantial amounts of passive scalar has yet begun to flow out into the exhaust channel.



Figure 3.19: The passive scalar visualised in the exhaust channel at 160 CAD.



Figure 3.20: The passive scalar visualised in the exhaust channel at 180 CAD.

In Figure 3.21 the compression stroke has begun, i.e. the piston has changed direction of motion and is on its way up again. The air fuel mixture has now started to flow into the cylinder in a significant amount. This results in a flow of air fuel mixture into the exhaust channel as well. This reveals that the trapping efficiency is below 100 %.

A couple of CAD later the scavenging port is about to close. The amount of passive scalar in the cylinder as well as in the exhaust channel has increased.

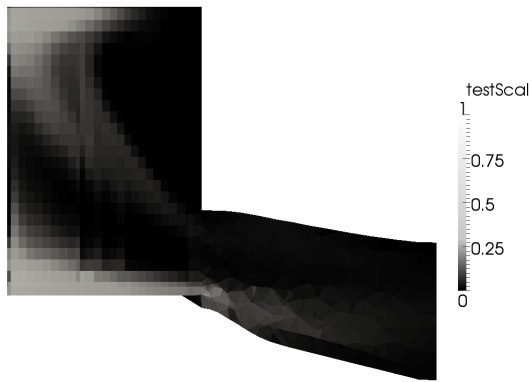


Figure 3.21: The passive scalar visualised in the exhaust channel at 200 CAD.

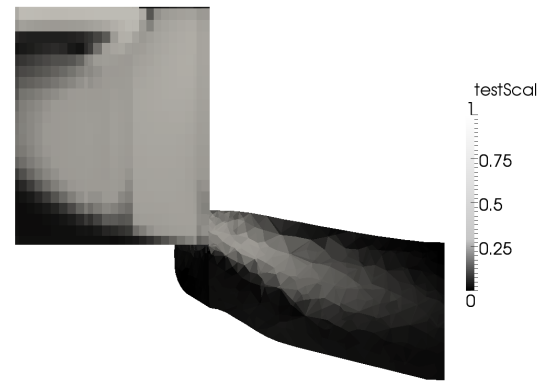


Figure 3.22: The passive scalar visualised in the exhaust channel at 240 CAD.

Figure 3.23 shows that the exhaust port just has closed. The cylinder is quite rich on air fuel mixture. The exhaust channel also contains a fair amount of air fuel mixture, (passive scalar).

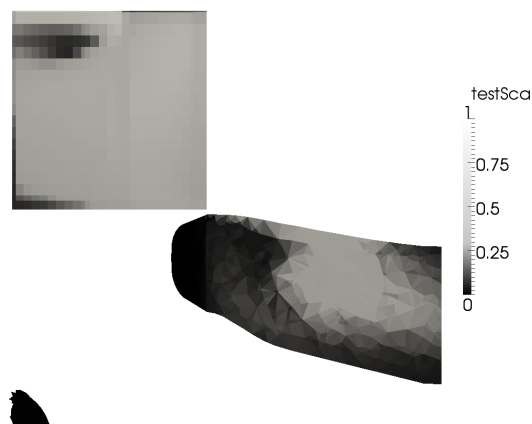


Figure 3.23: The passive scalar visualised in the exhaust channel at 259 CAD.

To measure how well the H576 engine scavenging works the trapping efficiency for the air fuel mixture is calculated, see equation (1.6) in section 1.3.3. The trapping efficiency is found by calculating the mass of the air fuel mixture in the cylinder after the exhaust port has closed, which is the mass of delivered air fuel mixture retained. The total mass of air fuel mixture delivered is found by adding the mass of the air fuel mixture left in the exhaust channel. The mass left in the exhaust channel from the previous cycle must of course be subtracted from this mass. The mass which has left the exhaust outlet from the opening of the exhaust port until the close of the exhaust port must also be added. For every new cycle the amount of air fuel mixture is set to zero in the cylinder. The trapping efficiency of the air fuel mixture is now found. The trapping efficiency for the first four revolutions is shown in Table 3.1. After four revolutions, or 1440 CAD, the trapping efficiency is considered to have a periodic behavior over each cycle and it is considered as converged.

Table 3.1: Trapping efficiency for four revolutions of the crankshaft.

	360 CAD	720 CAD	1080 CAD	1440 CAD
Trapping efficiency	99.34 %	93.14 %	93.61 %	93.77 %

3.2 Validation

The accuracy of the simulations performed by the OpenFOAM the results must be validated. This is important to find out if OpenFOAM is a useful tool for this kind of simulations. The results from OpenFOAM have been compared with results from the 1-d simulations as well with results from a commercial CFD software.

3.2.1 Commercial 1-D software

To determine the correlation between the 1-d simulations and the OpenFOAM simulations some kind of comparison is required. The mass flow over the inlets and the outlet is used as the variable for comparison. Figure 3.24 shows the results for the scavenging port inlets. In Figure 3.25 is the stratifying inlet results presented and in Figure 3.26 the exhaust outlet comparison.

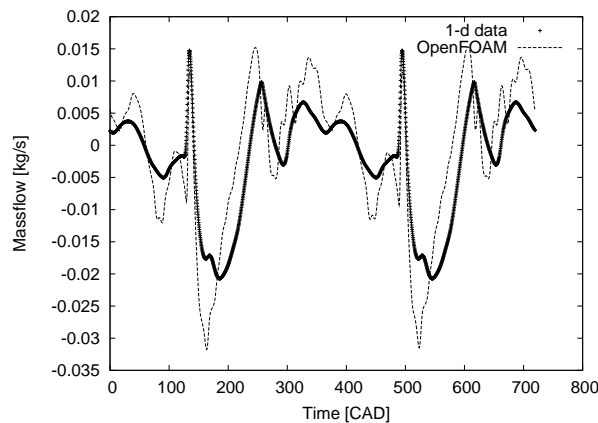


Figure 3.24: Intake inlet massflow from 1-d simulations and OpenFOAM.

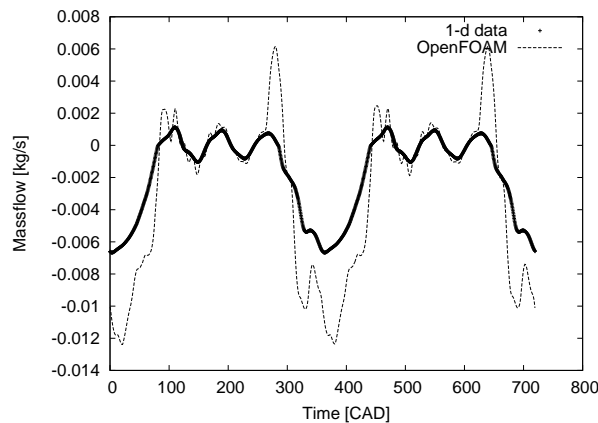


Figure 3.25: Stratifying inlet massflow from 1-d simulations and OpenFOAM.

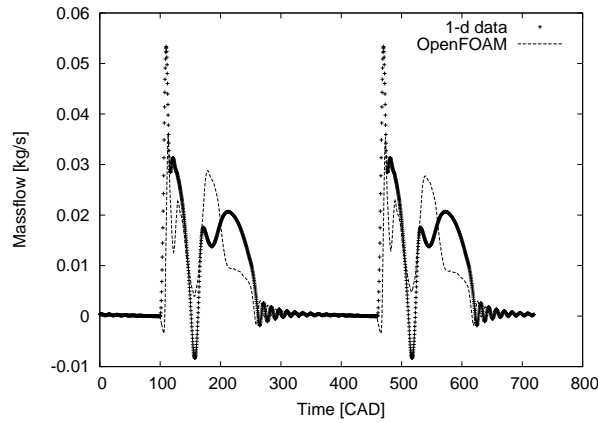


Figure 3.26: Exhaust outlet massflow from 1-d simulations and OpenFOAM.

There are obvious differences in the massflow for the different simulations. The shape of the curves are however in a quite good correlation. There are a number of possible reasons for the differences in magnitudes. The OpenFOAM simulations gives a better representation of the dynamics in the system since the domain has a higher resolution. The geometries are quite different also. The 1-D simulations are based on a geometry consisting of cylinders while the OpenFOAM geometry is close to the real geometry, except for the modifications.

3.2.2 Turbulence modelling

The $k-\varepsilon$ turbulence model is used for turbulence modelling. To determine the how the flow depends on the choice of turbulence model simulations where done using the $k-\omega-SST$ turbulence model. Figure 3.27 shows a comparison of the massflow through the scavenging channel inlet for the two turbulence models, $k-\varepsilon$ and $k-\omega-SST$. A similar comparison for the stratifying channel inlet and the exhaust outlet are found in Figure 3.28 and Figure 3.29.

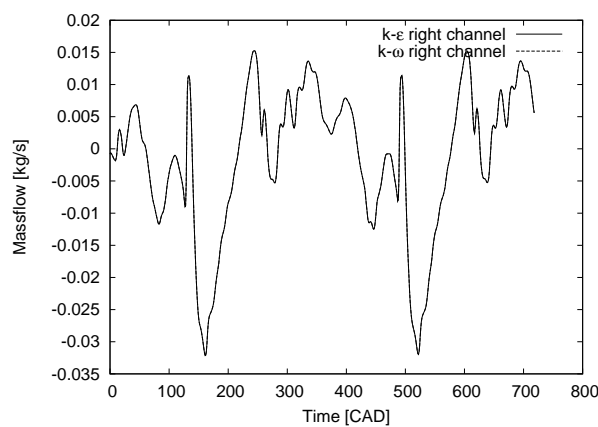


Figure 3.27: Comparison of massflow in intake channels for $k-\varepsilon$ and $k-\omega$ turbulence models.

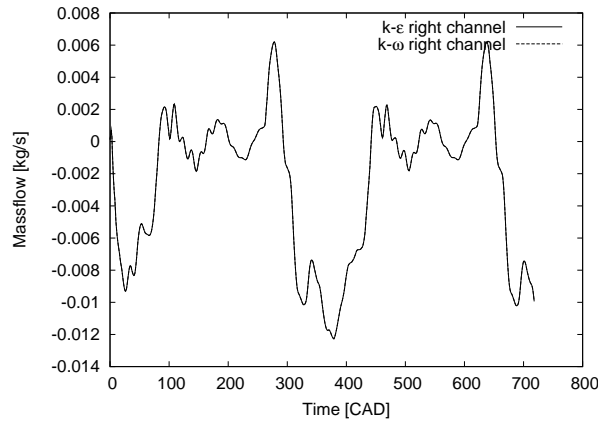


Figure 3.28: Comparison of massflow in stratifying channels for $k - \varepsilon$ and $k - \omega$ turbulence models.

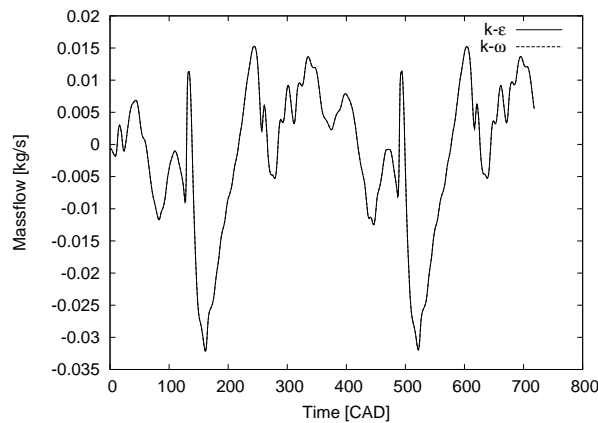


Figure 3.29: Comparison of massflow in exhaust channel for $k - \varepsilon$ and $k - \omega$ turbulence models.

To further determine how much impact the choice of turbulence model has on the results the actual flow was studied. The turbulent kinetic energy, i.e. the turbulence, was compared for the two turbulence models where minor differences are to be seen in the turbulent kinetic energy. However the overall flow behavior is kept when changing from the $k - \varepsilon$ to the $k - \omega - SST$ turbulence model as well as the trapping efficiency. See Figure 3.30 and Figure 3.31 for a comparison of the turbulent kinetic energy k . Similar comparisons are found in Figure 3.32 and 3.33 and in Figure 3.34 and 3.35.



Figure 3.30: The turbulent kinetic energy visualised at 0 CAD using. The $k-\varepsilon$ turbulence model is used.



Figure 3.31: The turbulent kinetic energy visualised at 0 CAD using. The $k-\omega$ turbulence model is used.

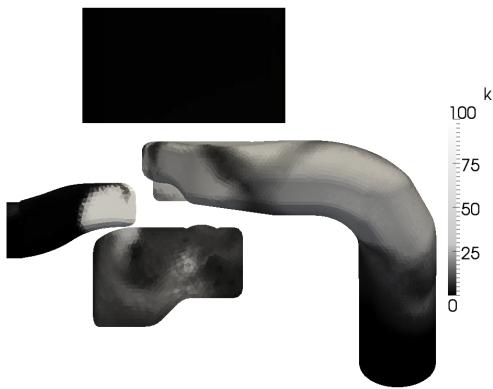


Figure 3.32: The turbulent kinetic energy visualised at 101 CAD using. The $k-\varepsilon$ turbulence model is used.



Figure 3.33: The turbulent kinetic energy visualised at 101 CAD using. The $k-\omega$ turbulence model is used.

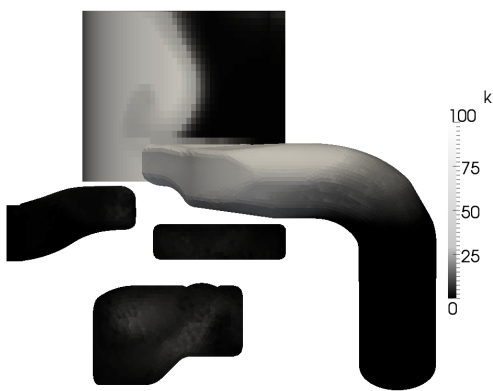


Figure 3.34: The turbulent kinetic energy visualised at 180 CAD using. The $k-\varepsilon$ turbulence model is used.

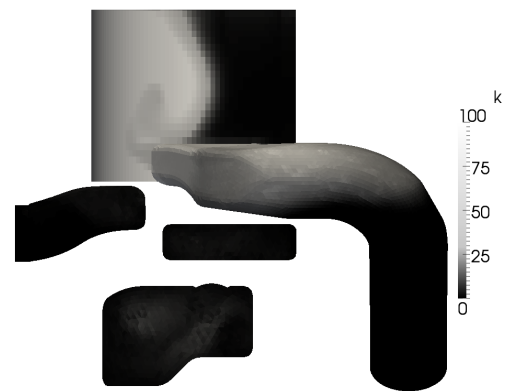


Figure 3.35: The turbulent kinetic energy visualised at 180 CAD using. The $k-\omega$ turbulence model is used.

3.2.3 Commercial 3-D software

The results from the OpenFOAM simulations are also compared with results from a simulation using the commercial CFD software Fluent. The case which was run in Fluent used the same geometry and boundary conditions. Similar as for the OpenFOAM case the pressure and temperature in the cylinder are set just before the exhaust port is about to open. The following figures shows a comparison in massflow for Fluent, the OpenFOAM and 1-D simulations. See Figure 3.36, Figure 3.37 and Figure 3.38 for this comparison. It is found that the the massflow from the two 3-D simulations correlates well. Still there are smaller differences in magnitude and the reason for this can be the different type of solution algorithm used. The OpenFOAM simulation uses the PISO-algorithm while the Fluent solution is obtained by the use of the SIMPLE-algorithm, see [7] for a description of the algorithms. The time step in OpenFOAM is adaptive and based on the Courant number while the time step in Fluent is set to a fixed value.

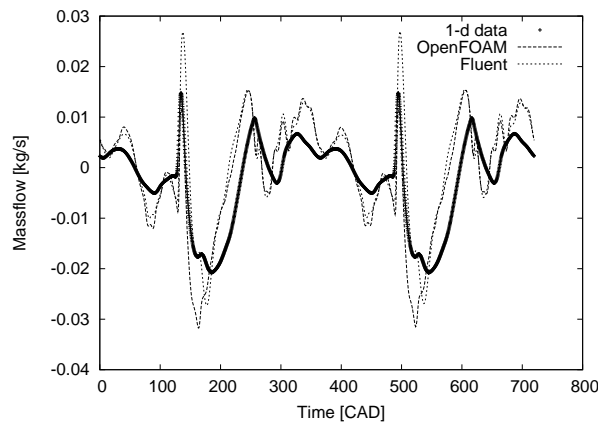


Figure 3.36: Intake inlet massflow from 1-d simulations, OpenFOAM and Fluent.

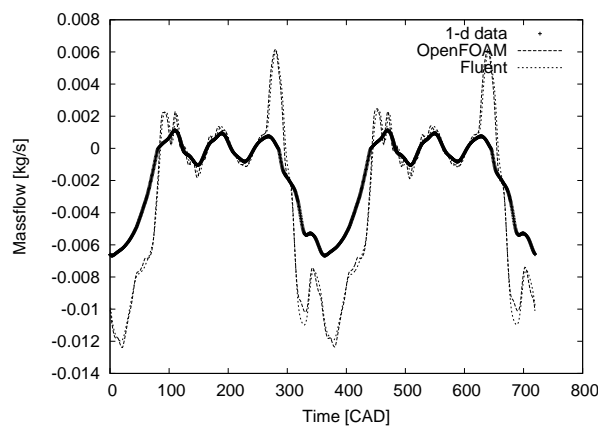


Figure 3.37: Stratifying inlet massflow from 1-d simulations, OpenFOAM and Fluent.

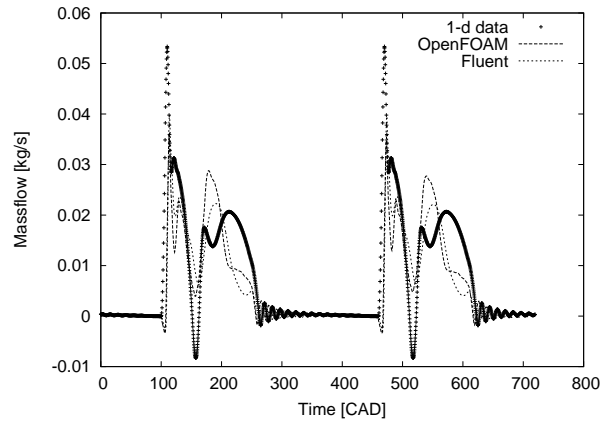


Figure 3.38: Exhaust outlet massflow from 1-d simulations, OpenFOAM and Fluent.

4 Conclusions

In this section the results are concluded. How OpenFOAM works for this kind of applications is also discussed. This section is closely associated with the section containing the results, namely section 3.

From the simulations it is quite clear that the flow in the scavenging channels and stratifying channels are symmetric regarding the left and right side. This might not be a surprise since the same boundary conditions are used and the geometry is axisymmetric over the x-axis. However this is not an obvious behavior. Since the flow is turbulent, especially in the cylinder which is the connection between the right and left scavenging channels, the flow cannot be regarded as axisymmetric. When the pass is open between the scavenging channel and the stratifying channel the left and right stratifying channels are connected through the piston pocket. From 3.30 it is obvious that also turbulence is present in the stratifying channels. Despite the turbulence it appears that the flow is rather axisymmetric.

Regarding Figure 3.30 in section 3.2.2 it is obvious that the sharp edge in the beginning of the stratifying channels is a source of turbulence. This behavior, turbulence due to sharp edge, is previously mentioned in section 2.1.1. In this case the sharp edge in the beginning of the scavenging channel resulted in such a turbulent behavior that the solver did not manage to get the solution to converge. The geometry was thus changed. These are two areas of the engine geometry which gives rise to high turbulence.

The trapping efficiency was calculated by using a passive scalar to trace the part of the fluid which is supposed to be the air fuel mixture, see section 1.3.7. The results of these calculations is a trapping efficiency of approximately 93 %, see section 3.1. From experiments on the real H576 engine the trapping efficiency was found to be 92 %. This indicates that the method has resulted in a solution quite close to the real numbers. One must remind that the case is limited and based on approximations.

What is also interesting is the performance of OpenFOAM. Both visually and in numbers, (the trapping efficiency and the massflow), the solution has converged after four revolutions of the crankshaft. When running the simulations in parallel on three CPU:s the simulation of one revolution takes approximately 24 hours. It is worth to mention that this is with the `checkMesh`-option turned on, see section 2.8.3. With the `checkMesh`-option turned off the time could be reduced by approximately 19 %, see Table 2.4 in section 2.8.3. The simulations were also run with a fixed time step set to the smallest value obtained by the Courant-number based time step. The same results in the trapping efficiency were obtained as for the Courant number based time step. Simulations with a Courant-number based time step is thereby recommended to save computational time.

This means that if a change is made to the geometry or some boundary condition has changed a new solution is obtained within four days, despite the fact that only three CPU:s are used. This makes it a rather powerful tool for this kind of simulations.

5 Discussion and recommendations

It appears that OpenFOAM-1.5-dev and the `twoStrokeEngine`-library is useful tool for this kind of simulations. Even though the case was based on some rough approximations and assumptions the results, (the trapping efficiency), was quite close to the real case. A better model would be quite easily achieved by using the engine geometry without any modifications. To further make the model more accurate the geometry might be extended by crankcase, muffler, carburettor, to mention a few examples. The next step would be to use different fluids and eventually model the combustion to further develop the model. Tools for this kind of simulations are already present in OpenFOAM-1.5-dev. To even further enhance the model heat transfer between the walls and the fluids could be incorporated. The fantasy, (or probably the resources), sets the limits for how accurate the model can be.

There are many things to do to further validate the model and the method. Running the same simulation on another engine, testing different solving algorithms will give an indication of the accuracy in this method. To find out the effect of the turbulence, simulations can be run with just a laminar flow. It is also interesting to simulate a completely inviscous flow. The dependency on the mesh should also be evaluated to find out how the solution depends on the mesh resolution.

The OpenFOAM software can be recommended for this kind of applications. In relatively short time hints about weak spots in the geometry can be achieved. Once the geometry is converted to OpenFOAM format it is quite a short process to have a running case.

References

- [1] John B. Heywood. *Internal Combustion Engine Fundamentals*. McGraw-Hill, Inc, 1988.
- [2] R.U.K. Gustafsson M. Bergman and B.I.R Jonsson. Scavenging system layout of a 25 cc two-stroke engine intended for stratified scavenging. *SAE paper no. 2002-32-1840*, 2002.
- [3] R.U.K. Gustafsson M. Bergman and B.I.R Jonsson. Emission and performance evaluation of a 25 cc stratified scavenging two-stroke engine. *SAE paper no. 2003-32-0047*, 2003.
- [4] OpenCFD Limited. *OpenFOAM The Open Source CFD Toolbox Programmer's Guide Version 1.5*, 2008.
- [5] OpenCFD Limited. *OpenFOAM The Open Source CFD Toolbox User Guide Version 1.5*, 2008.
- [6] Masanori Noguchi Toshiharu Sawada, Minrou Wada and Buhei Kobayashi. Development of a low emission two-stroke cycle engine. *SAE paper no. 980761*, 1998.
- [7] H.K. Versteeg and W Malalasekera. *An Introduction to COMPUTATIONAL FLUID DYNAMICS The Finite Volume Method second edition*. Pearson Education Limited, 2007.
- [8] Frank M. White. *FLUID MECHANICS fifth edition*. McGraw-Hill, 2003.