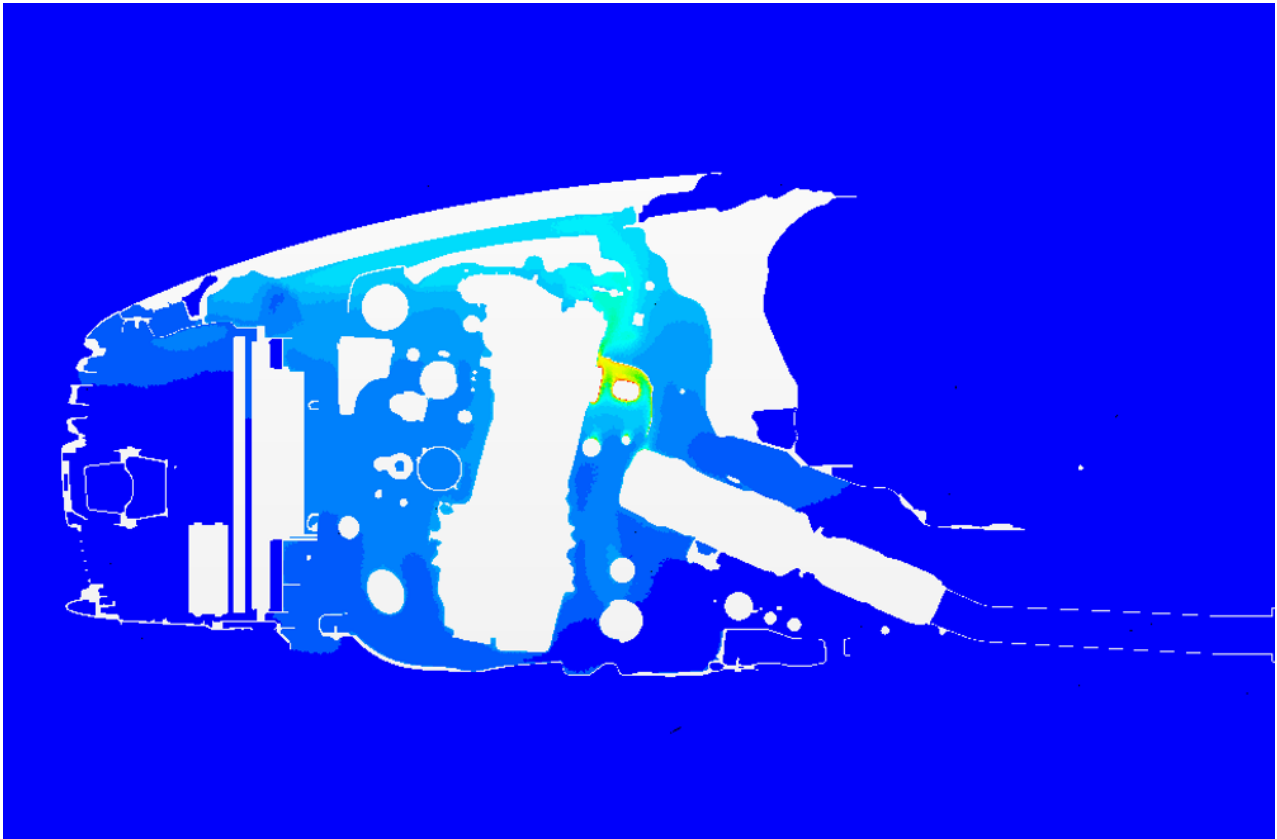




CHALMERS



Unsteady Thermal Management Simulations for a Passenger Vehicle using 1D and 3D Tools

Master's thesis in Automotive Engineering

SAKET KUMAR
DANIEL PASCUAL

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Unsteady Thermal Management Simulations for a Passenger Vehicle using
1D and 3D Tools

SAKET KUMAR
DANIEL PASCUAL

Department of Applied Mechanics
Division of VEAS and Combustion
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Unsteady Thermal Management Simulations for a Passenger Vehicle using 1D and 3D Tools
SAKET KUMAR
DANIEL PASCUAL

© SAKET KUMAR, DANIEL PASCUAL, 2016

Master's thesis 2016:06
ISSN 1652-8557
Department of Applied Mechanics
Division of VEAS and Combustion
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:

Temperature profile across the longitudinal-cross section of the underhood region of a Volvo S80 sedan, parked in a quiescent environment, obtained from steady state simulation in STAR-CCM+.

Chalmers Reproservice
Gothenburg, Sweden 2016

Unsteady Thermal Management Simulations for a Passenger Vehicle using 1D and 3D Tools
Master's thesis in Automotive Engineering
SAKET KUMAR
DANIEL PASCUAL
Department of Applied Mechanics
Division of VEAS and Combustion
Chalmers University of Technology

ABSTRACT

This work develops a coupled 1D-3D computational tool for analyzing conjugate heat transfer in the engine bay of a passenger vehicle. 1D thermal models of powertrain in conjunction with detailed 3D models of flow structures can provide a powerful means to correctly capture the physics of the system while keeping the computational cost reasonably low. To the authors' best knowledge, simultaneous coupling of 1D and 3D computational tools has not been applied to modelling thermal interaction in the vehicle powertrain before. Thus, this work constitutes a first step towards performing such coupled simulations and analyzing their outcome.

The present study builds on a 1D model of a powertrain cooling system, built in GT-SUITE software, by adding a comprehensive lubrication circuit and mean value engine model to monitor the time dependent temperatures of oil and engine solids. A 3D model of natural convection & radiation in the engine bay was built in STAR-CCM+ where the primary boundaries for heat transfer between the surrounding air in the underhood region and engine solids were defined. The two models were run simultaneously and data were exchanged for the surface temperatures of the primary engine components as well as the heat transfer rate between these components and the engine bay surroundings.

The developed tool was used to study the variation of mass averaged oil temperature over a customized driving cycle. This cycle consists of the recently introduced Worldwide harmonized Light duty driving Test Cycle (WLTC) with additional shut-down periods during which the car was assumed to be parked in a quiescent environment. The delivered numerical procedure proved to be an effective approach for such drive cycles. The present work can thus serve as a solid basis for implementing more advanced thermal management models in the future.

Keywords: coupled 1D-3D computational tool, conjugate heat transfer, GT-SUITE, mean value engine model, natural convection, radiation, STAR-CCM+, mass-averaged oil temperature, WLTC

PREFACE

This master thesis has been completed in partial fulfilment of the requirements for a Master's degree in Automotive Engineering. It is a cross-divisional project between the Vehicle Engineering & Autonomous Systems (VEAS) and Combustion divisions of the Applied Mechanics department at Chalmers University of Technology, Gothenburg, Sweden. The primary work has been performed from January to June 2016 by two students in the division of VEAS while utilizing resources from both the participating divisions.

ACKNOWLEDGEMENTS

We would like to express our utmost sincere gratitude to all those who have helped us through the different stages of this project. First and foremost, we would like to thank our project supervisors, Blago Minovski and Jelena Andric. Without their efficient guidance, critical inputs and untiring support, this project could never have been completed. We would also like to thank them for providing us with all the required resources and setting-up a comfortable working environment which helped in the smooth operation of our project. It was highly motivating and a great pleasure for us to work under their supervision.

Furthermore, we would like to thank the engineers of Volvo Cars Corporation of Gothenburg for all the data provided by them, especially Mirko Bovo from their Engine CAE department, for his valuable insights and feedback on our work.

A special thanks goes to our examiners at Chalmers, Prof. Lennart Löfdahl and Prof. Sven Andersson for their active interest and belief in our efforts as well as the spontaneous humour whenever required.

Last but not the least, we would like to thank all the support staff at Gamma Technologies and CD-adapco for their advice and technical inputs throughout our project.

Saket Kumar and Daniel Pascual

Gothenburg, September 12, 2016

CONTENTS

| | |
|---|-------------|
| Abstract | i |
| Preface | iii |
| Acknowledgements | iii |
| Contents | v |
| Nomenclature | viii |
| Abbreviations | viii |
| 1 Introduction & Motivation | 1 |
| 1.1 Project Background and Scientific Relevance | 1 |
| 1.2 Thesis Scope | 1 |
| 1.3 Limitations | 3 |
| 2 Theoretical and Modelling Background | 4 |
| 2.1 Heat Transfer | 4 |
| 2.1.1 Conductive Heat Transfer | 4 |
| 2.1.2 Convective Heat Transfer | 4 |
| 2.1.3 Radiation Heat Transfer | 4 |
| 2.2 Governing Equations of Fluid Flow | 5 |
| 2.3 Turbulence Modelling | 6 |
| 2.4 Grashof Number (Gr) | 6 |
| 2.5 Cooling Systems | 7 |
| 2.5.1 Coolant Pump | 7 |
| 2.5.2 Thermostat | 7 |
| 2.5.3 Charge Air Cooler(CAC) | 8 |
| 2.5.4 Radiator | 8 |
| 2.5.5 Fan | 9 |
| 2.5.6 Expansion Tank | 9 |
| 2.5.7 Lubrication System | 10 |
| 2.5.8 Oil Cooler | 10 |
| 2.5.9 Driving Cycle | 11 |
| 2.6 Computer modeling | 11 |
| 2.6.1 GT-SUITE | 11 |
| 2.6.2 Artificial Neural Networks (ANN) | 11 |
| 2.7 Macro | 13 |
| 2.8 Comma Separated Values File | 13 |
| 3 Methods | 14 |
| 3.1 Modelling Domains | 14 |
| 3.2 3D CFD Model of Underhood Region | 15 |
| 3.2.1 Generated Mesh | 16 |
| 3.2.2 Boundary Conditions | 17 |
| 3.2.3 Solver Settings | 19 |
| 3.2.4 Results from Standalone Simulations | 20 |
| 3.3 1D Model of powertrain cooling system | 20 |
| 3.3.1 Upgrading | 20 |
| 3.3.2 Oil Circuit | 21 |
| 3.3.3 Mean Value Engine Modeling | 25 |
| 3.4 Final Coupled Model | 29 |
| 3.4.1 Coupling Mechanism | 29 |

| | | |
|----------|--|-----------|
| 3.4.2 | Quasi-Stationary Approximation | 31 |
| 3.4.3 | Coupling Strategy | 31 |
| 3.4.4 | Customized Drive Cycle | 33 |
| 3.4.5 | Ambient Conditions | 33 |
| 4 | Results and Discussion | 34 |
| 4.1 | Predicted Oil Temperatures | 34 |
| 4.2 | Sensitivity Analysis | 34 |
| 4.3 | Thermal Soak Phenomenon | 36 |
| 5 | Concluding Remarks | 37 |
| 6 | Future Work | 38 |
| | Bibliography | 38 |
| A | Appendix | II |
| A.1 | Java Macro Code | II |
| A.2 | Distribution of Content | IX |

LIST OF FIGURES

| | | |
|------|---|----|
| 1.1 | Subject vehicle: Volvo S80 sedan; source: google.com | 2 |
| 2.1 | Pump of the cooling system | 7 |
| 2.2 | Thermostat | 7 |
| 2.3 | Charge air cooler | 8 |
| 2.4 | Radiator | 8 |
| 2.5 | Vehicle fan | 9 |
| 2.6 | Expansion tank | 9 |
| 2.7 | Schematic of Primary Heat Exchanges | 10 |
| 2.8 | WLTC Class 3 | 11 |
| 2.9 | GT user interface | 12 |
| 2.10 | Artificial Neural Network of two inputs, three neurons and one output | 12 |
| 3.1 | Schematic representation of the all the interactions taking place between different elements of the entire modelling domain | 14 |
| 3.2 | Division of modelling domains | 15 |
| 3.3 | Computational domain for the 3D-CFD model defined in STAR-CCM+ | 15 |
| 3.4 | Longitudinal cross-sectional view of the generated computational mesh grid | 17 |
| 3.5 | Boundaries defined for the exchange of information between the two software | 18 |
| 3.6 | Temperature profile across the longitudinal-cross section obtained from steady state simulation with nominal boundary conditions. | 20 |
| 3.7 | Detailed Lubrication Model | 21 |
| 3.8 | Reynolds Vs Loss coefficient | 22 |
| 3.9 | Areas related to the piston lubrication | 23 |
| 3.10 | Bottom area of the engine block | 24 |
| 3.11 | Cylinder head surface measured | 24 |
| 3.12 | Simplified Oil Circuit | 25 |
| 3.13 | Complete Thermal Engine | 25 |
| 3.14 | Comparison between experimental and computed data | 26 |
| 3.15 | Neural Network | 27 |
| 3.16 | Comparison of the brake power computed by the mean value model and the detailed gas exchange model | 28 |
| 3.17 | Comparison of the exhaust manifold temperature computed by the mean value model and the detailed gas exchange model | 28 |
| 3.18 | Comparison of the exhaust manifold temperature computed by the mean value model and the detailed gas exchange model | 29 |
| 3.19 | CosimFileBased template in GT-SUITE | 30 |
| 3.20 | Schematic of the coupling strategy employed by the Java macro for the exchange of data between the two software | 32 |
| 3.21 | Customized drive cycle constructed for analysis of the coupled simulation tool | 33 |
| 4.1 | Variation of temperature of oil in the sump for simulation number discussed in section | 34 |
| 4.2 | Mass-averaged oil sump temperatures from coupled simulations | 35 |
| 4.3 | Mass-averaged exhaust manifold temperatures from coupled simulations | 36 |
| 4.4 | Thermal soak phenomenon observed for the cylinder head | 36 |

NOMENCLATURE

| Parameter | Symbol | Value | Unit |
|--------------------------------------|------------|---------------|------------------|
| Area | A | | m^2 |
| Convection heat transfer coefficient | α_0 | | $Wm^{-2}K^{-1}$ |
| Engine speed | N | | RPM |
| Internal energy | U | | J |
| Lower heating value | Q_{hv} | | $kJkg^{-1}$ |
| Mass | m | | kg |
| Mass flow of air | m_a | | $kg s^{-1}$ |
| Thermal conductivity | k | | $Wm^{-1}K^{-1}$ |
| Time | t | | s |
| Specific heat capacity | c_p | | $Jkg^{-1}K^{-1}$ |
| Stefan-Boltzmann constant | σ | $5.67 * 10^8$ | Wm^{-2} |
| Surface area | A_s | | m^2 |
| Surface temperature | T_s | | $^{\circ}K$ |
| Rate of heat transfer | Q | | W |
| Rate of heat transfer per area | \dot{q} | | Wm^{-2} |

ABBREVIATIONS

| | |
|------|--|
| CAC | Charge Air cooler |
| HTC | Heat transfer coefficient |
| WLTC | Worldwide harmonized light vehicles test cycle |
| PCS | Powertrain cooling systems |

Introduction & Motivation

1.1 Project Background and Scientific Relevance

In the current scenario of constantly rising fuel efficiency targets, a lot of research work being done in the automotive industry is devoted to developing vehicle thermal management solutions [1][2] that can contribute in this regard. Employing thermal encapsulations for the engine is a technique for reducing the heat it dissipates to the environment, after it has been switched off. This approach is beneficial for the reduction of fuel consumption and gaseous emissions. By encapsulating the engine, the oil temperatures are maintained at much higher levels till the next instance of running of the vehicle. In this way, it is possible to avoid the cold start-up periods and hence minimize the friction losses & exhaust emissions. The efforts of a major automobile manufacturer in this direction were recently recognized by the European Commission recently [3]. In order to implement solutions with similar capabilities, it is essential to understand and model the conjugate heat transfer phenomenon in the engine bay of parked vehicles. Experiments have been carried out to improve the understanding of buoyancy driven flows in the engine bay of vehicles that are parked in a quiescent environment after having been subjected to a typical drive cycle [4].

At the same time, there is a rising demand for developing simulation models to minimize or even eliminate physical testing. These help significantly in reducing the cost and development life-cycle time of the vehicles. The solutions for vehicle thermal management have a significant effect on the overall architecture of a vehicle. Therefore the ability to perform reliable thermal management simulations at concept stage is critical for accelerated development times. For a vehicle parked in a quiescent environment, buoyancy driven convection and radiation are the dominating modes of heat transfer in the underhood region. A well-documented numerical method described by Minovski et al.[5] offers a reliable procedure for modelling this phenomenon on a commercial 3D CFD platform. On the other hand, 1D simulation tools are typically favoured for transient simulations of powertrain cooling systems. This work investigates the possibility of coupling these 1D and 3D simulation platforms to correctly predict the conjugate heat transfer in the engine bay while reducing computational costs. It constitutes the first step towards performing simultaneously coupled simulations of this nature.

1.2 Thesis Scope

This thesis work comprises the following major steps :

- 1) Further development of an integrated 1D GT-POWER model of the powertrain cooling system of Volvo S80. The model is upgraded by adding the oil circuit elements to the engine thermal model. A mean value engine mode is implemented to more accurately capture the gas-exchange process.
- 2) Creation of a 3D CFD model of the vehicle underhood flow using a commercial CFD software STAR CCM+. This model predicts natural convection heat transfer in the engine bay based on the method previously proposed by Minovski et al.[5].
- 3) Coupling between the 1D and 3D simulation tools in order to capture the dependence between the buoyancy driven flow in the underhood region during the thermal soak period and the heat transfer in the engine coolant and lubrication circuits.
- 4) Performing coupled 1D-3D simulations using a customized driving cycle to investigate the feasibility of the coupled modeling framework.



Figure 1.1: Subject vehicle: Volvo S80 sedan; source: google.com

1.3 Limitations

The study has been carried out under the following limitations:

- 1) The simulation results will be examined for consistency from a theoretical perspective. However, a comparison with experimental data is beyond the scope of the present work.
- 2) The heat exchange due to forced convection during the periods when the engine is turned on are neglected. However, the primary effect of this flow is taken into account by assuming appropriate values for the heat transfer rate available in the literature.
- 3) Due to limited time frame, the 3D CFD simulations consider the heat dissipation in the engine bay to be a quasi-stationary process, refer to Sect. 3.4.2 for more details.
- 4) A uniform temperature distribution has been assumed across the engine surfaces.

Theoretical and Modelling Background

This chapter introduces the readers to the theoretical background of this project. To begin with, it highlights the primary concepts on which the models are based. A brief description of the primary components of powertrain cooling systems follows and the topics of computer aided modelling relevant to this work are introduced here as well.

2.1 Heat Transfer

Matter is made up of molecules and atoms that are in constant motion and possess heat or thermal energy. The greater the particle motion and energy levels, the higher will be the temperature of the system. Heat transfer always occurs from a higher-temperature system to a lower-temperature system. The different modes of heat transfer are conduction, convection and radiation. The respective characteristics and concepts behinds these are discussed below.

2.1.1 Conductive Heat Transfer

Conduction is the phenomenon that consists of heat propagation between two bodies or different parts from the same body that are at different temperatures due to the agitation of thermal molecules.

The Fourier's Law for heat conduction in one-dimensional form is given as:

$$\dot{Q}_{cond} = -kA \frac{dT}{dx} \quad (2.1)$$

where \dot{Q}_{cond} heat flow through a cross-sectional area A and k is the thermal conductivity. [6].

2.1.2 Convective Heat Transfer

Convection is defined as the transmission of heat resulting from the bulk motion of the molecules. This phenomenon only occurs in fluids as their particles can exhibit bulk current flows and significant diffusion. These can occur due to density differences (natural convection) or by external means such as pumps and fans(forced convection).Convective heat transfer is mathematically described by the equation 2.2 [6]:

$$\dot{Q}_{conv} = hA(T_{inf} - T_s) \quad (2.2)$$

It can be deduced from the above equation 2.2 establishes that the heat flux \dot{Q}_{conv} between a fluid at temperature T_{inf} measured far away from the wall that is at temperature T_s is proportional to the area in contact h ,heat transfer coefficient, and the temperature difference between both.

2.1.3 Radiation Heat Transfer

This mode of heat transfer exists between any two bodies that are at different temperatures even without any physical connection or material medium between them. Radiation is a form of electromagnetic emission that is emitted by any body which is at a temperature higher than absolute zero. The emission of an idealized black body can be calculated with equation 2.3 [6]:

$$\dot{Q}_{rad} = \sigma A_s T_s^4 \quad (2.3)$$

Where A_s and T_s are the area and the temperature of the black body respectively and $\sigma = 5.67 * 10^8 Wm^{-2}$ is the Stefan-Boltzmann-constant. For a non black body, this phenomenon is described by the equation 2.4 [6]:

$$\dot{Q}_{rad} = \epsilon \sigma A_s T_s^4, \quad (2.4)$$

where ϵ is the emissivity of the surface that also depends on the temperature of the body and varies within the range $0 < \epsilon < 1$ [6].

2.2 Governing Equations of Fluid Flow

The governing equations for incompressible flow comprise of three main equations. These are the equations of mass continuity, conservation of momentum and the conservation of energy respectively. They are given as in equations 2.5, 2.6 and 2.7 below:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (2.5)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \Delta u_i \quad (2.6)$$

$$\frac{\partial e}{\partial t} + u_j \frac{\partial e}{\partial x_j} = \phi + \frac{1}{\rho} \frac{\partial (K \frac{\partial T}{\partial x_j})}{\partial x_j} \quad (2.7)$$

where u_i is the velocity component in the i -th direction, p is the pressure, ρ and ν denote the density and the kinematic viscosity of the fluid respectively. Also Δ is the Laplacian operator. e is the internal energy per unit mass and ϕ is the viscous dissipation per unit mass given as:

$$\phi = \frac{\nu}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.8)$$

The governing equations are formulated for the instantaneous flow field.

Reynolds decomposition technique is used to account for the equation terms for turbulent flows. The instantaneous variables are represented as a sum of their time-averaged and fluctuating components. For the instantaneous velocity and pressure fields, this decomposition reads as:

$$u_i = U_i + u'_i \quad (2.9)$$

$$p = P + p' \quad (2.10)$$

$\overline{U_i}$ and P are the time averaged components, and u'_i and p' are the fluctuating components. The Reynolds averaged Navier Stokes (RANS) equations are given as:

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (2.11)$$

$$U_j \frac{\partial U_i}{\partial x_j} = \frac{\partial \left[-\frac{P}{\rho} \delta_{ij} + 2\nu \overline{S_{ij}} - \overline{u'_i u'_j} \right]}{\partial x_j} \quad (2.12)$$

Here δ_{ij} is the symbol for Kronecker delta function;

$$S_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad \text{is the mean strain tensor} \quad (2.13)$$

The term $-\overline{u'_i u'_j}$ in equation 2.12 is referred to as the Reynolds stress. This non-linear term needs to be modelled in order to solve the RANS equations.

2.3 Turbulence Modelling

The presented work employs the Realizable k- ε turbulence model. In this model, the approximation for the Reynolds stress is based on the Boussinesq assumption, i.e.

$$-\rho \overline{u_i u_j} = 2\mu_t S_{ij} - \frac{2}{3}K \delta_{ij} \quad (2.14)$$

here μ_t is given as: $\mu_t = \rho C_\mu \frac{K^2}{\varepsilon}$

where K is the turbulent kinetic energy and ε is the turbulent dissipation. Contrary to the standard k- ε turbulence model, C_μ is considered a variable in this approach, see [7] for more details. Realizable k- ε model has better ability to correctly capture the physics of buoyancy driven turbulent flows compared to the other options such as the standard k- ω , SST k- ω etc.

2.4 Grashof Number (Gr)

The Grashof's number (Gr), named after the German engineer Franz Grashof, is a dimensionless quantity used for analyzing the flow characteristics and the velocity distribution in scenarios where buoyancy driven flow exists. The turbulent nature of the buoyancy driven flow in the engine bay of a vehicle can be identified with the help of this quantity. Physically, it is the ratio of the buoyant forces to the viscous forces and is analogous to the Reynolds number for cases of forced convection. Mathematically,

$$Gr = \frac{Buoyant\ forces}{Viscous\ forces} \quad (2.15)$$

Also,

$$Gr_L = \frac{g\beta(T_s - T_\infty)L^3}{\nu^2}; \text{ for vertical flat plates} \quad (2.16)$$

and

$$Gr_D = \frac{g\beta(T_s - T_\infty)D^3}{\nu^2}; \text{ for bluff bodies} \quad (2.17)$$

where:

g is acceleration due to Earth's gravity

β is the coefficient of thermal expansion (equal to approximately $1/T$, for ideal gases)

T_s is the surface temperature

T_∞ is the bulk temperature

L is the vertical length

D is the diameter

ν is the kinematic viscosity.

while The L and D subscripts indicate the length scale basis for the Grashof Number. In the cases of buoyancy driven flow from flat vertical plates, the transition to turbulent flow is characterized by the occurrence of Grashof number in the range specified as $10^8 < Gr_L < 10^9$. For higher values of this number the boundary layer is turbulent while at lower Grashof numbers, it is laminar.

The product of the Grashof number and the Prandtl number gives the Rayleigh number, which is another dimensionless number that characterizes convection problems in heat transfer. The theory governing buoyancy driven flows is available in more detail in the dedicated literature works in this field.[7].

The following section introduces the readers to the primary components of a vehicle powertrain cooling system that have been modelled in this project.

2.5 Cooling Systems

The main aim of all cooling system is to maintain the operational temperatures of the powertrain within desired limits and prevent it from reaching any critical scenarios where the components may get damaged. For internal combustion engines, initial warming up of the engine during cold starts, cooling down of pistons, engine block and cylinder head are the main functions of the cooling system. In order to carry out the cool down there are two ways to do it, either using air directly as cold medium or using a coolant liquid as intermediate medium to cool down the engine and then cooling down the coolant liquid using a radiator.

2.5.1 Coolant Pump

The coolant pump is the device in charge of supplying energy to coolant fluid in order to circulate it through the coolant circuit. Pump usually is connected to the crankshaft using a belt with a speed ratio, from this connection extracts the necessary energy to rotate. A sketch of centrifugal pump [8] can be found on Figure 2.1.

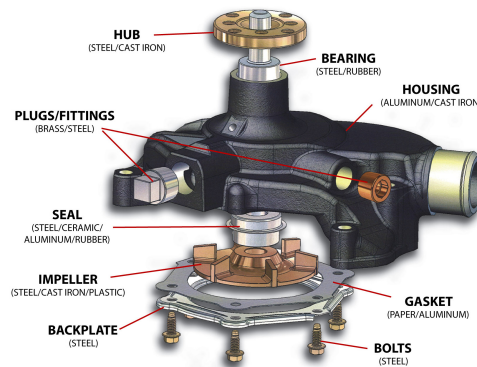


Figure 2.1: Pump of the cooling system

2.5.2 Thermostat

A thermostat [9] is a valve which controls coolant flow going from engine to the radiator. There are two different types of the thermostats, first type is based on wax tensibility. Second type of thermostats is based on the fluid mixture tensibility with a mixture of water and alcohol. Nowadays, the most common thermostat is that of the wax type. When the thermostat is opening, the coolant starts to flow to the radiator circuit. The thermostat opens depending on the coolant temperature and when the temperature of the coolant reach 70–80°C starts to open. The thermostat is fully opened after reaching the temperature 85–95°C. These values can vary depending on the car and the strategy followed. [10]

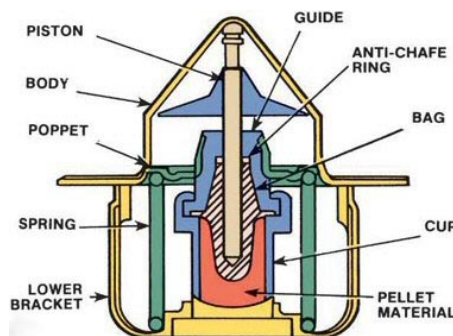


Figure 2.2: Thermostat

2.5.3 Charge Air Cooler(CAC)

Forced induction is an effective method for improving the power output of internal combustion engines while maintaining their displacement or size. Charging of the engines is provided by the turbocharger or supercharger. However, increasing the temperature of the compressed air is not useful to rise the engine power. In order to enhance the efficiency of the combustion this air coming from the compressor is cooled down in the Charge air cooler [11] before going to the combustion chamber to increase its density and obtain a powerful combustion [10].

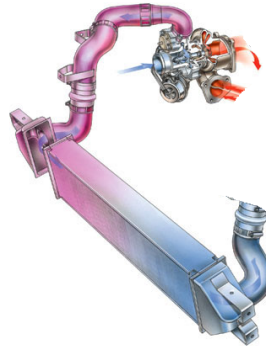


Figure 2.3: Charge air cooler

2.5.4 Radiator

A radiator is an liquid to air, cross-flow type heat exchanger that usually is placed after charge air cooler. It is the principal effective interface for heat exchange between coolant and surroundings environment. The most commonly used design for modern vehicle radiators is the finned aluminium type that uses flat tubes held by header plates. An example of a passenger car radiator [12] is shown on Figure 2.4

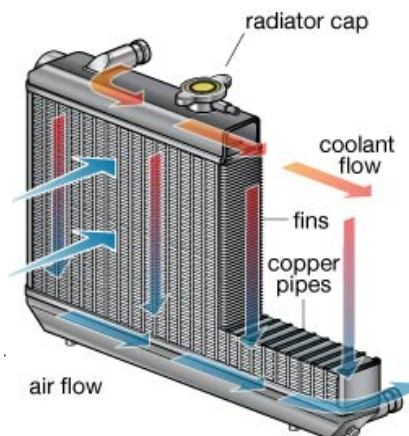


Figure 2.4: Radiator

The thickness and face areas sizes of the radiator cores can vary. Pressure drop across the coolant side of the radiator is mainly affected by the number and dimensions of the tubes where the liquid flows. At the air side is fin geometry what has high influence on the drop pressure and cooling efficiency of the radiator. Heat is transferred from the hot coolant to the wall by convection, by conduction through the wall and to the cold fluid by convection again.

Proper selection of the radiator is a very important task in automotive engineering that requires large

amounts of computer simulations to reach the requirements and design targets. Nevertheless it is necessary to have previous performance knowledge of the radiator such as pressure loss and cooling rate that usually are provided by the supplier [13].

2.5.5 Fan

In order to increase air flow through the radiator at low speeds a fan is installed. Depending on the size of the engine combination of two and more fans can be used. Fans [14] can be driven by the engine, however nowadays fan speed is controlled by electric motor, making it independent to engine speed.

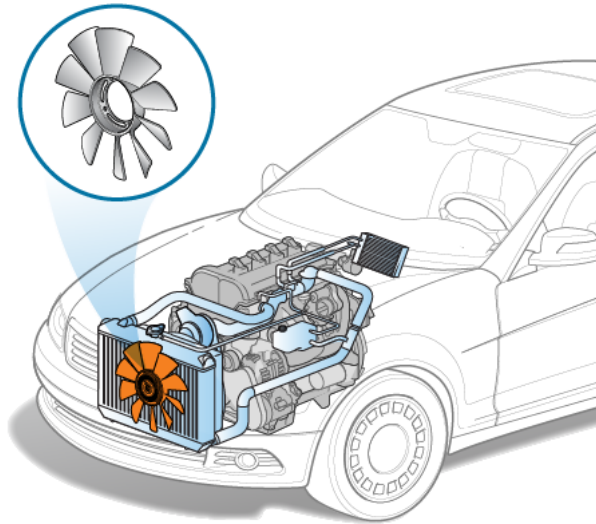


Figure 2.5: Vehicle fan

2.5.6 Expansion Tank

Expansion tank is a solid or deformable tank mounted at the top of the cooling system. The function of this tank is to permit coolant expansion at high temperatures since if the coolant is not allowed to expand, the pressure reached in the coolant circuit would be too high.(see Fig. 2.6) [15].

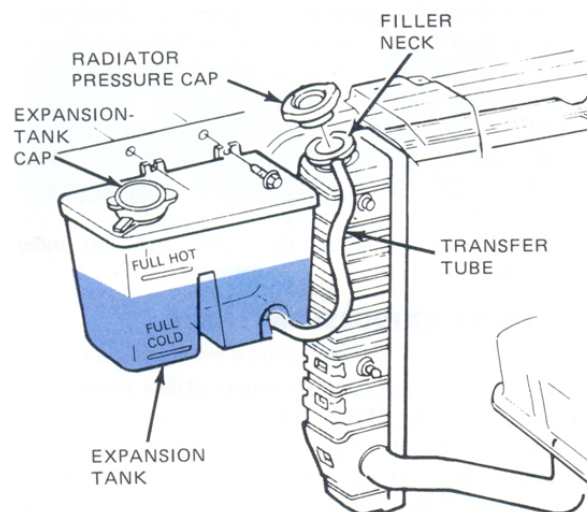


Figure 2.6: Expansion tank

2.5.7 Lubrication System

The lubrication system is responsible for friction reduction between moving parts inside the engine as well as protect them from wear and tear. If the lubrication layer breaks down, components in motion can reach temperatures high enough to lead to localized welding and hence failure of the engine. The lubricant is always in contact with the most critical components of the engine since its objective is to reduce friction, so at the same time that reduces friction levels, it extracts heat generated because of the inevitable friction and from the combustion process. A clear example of this double purpose is the piston lubrication throughout cylinder movement. Before being sent back to the sump, the hot lubricant is cooled down through the oil cooler. A schematic of primary heat exchanges in an engine is depicted in the Figure 2.7

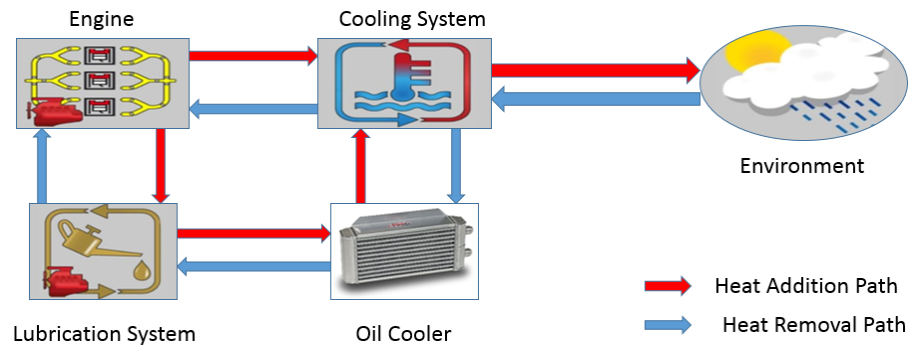


Figure 2.7: Schematic of Primary Heat Exchanges

2.5.8 Oil Cooler

Lubrication of the different moving components of an engine is essential at every moment. It is used for reducing friction losses at the engine. But the fluid used to this purpose, oil, gets warm up after lubricating and it is important to maintain the oil at a temperature lower than certain value and it can keep its properties. The purpose of a oil cooler is to maintain the temperature of the oil under certain value since if it reaches high temperature, the properties of the oil would change and the lubrication of the engine would not be proper.

Nevertheless the lubricant can be also warmed up in the oil cooler, for instance, at a cold start, the coolant warms up faster than the oil leading to a temporary phase where the oil is warmed up by the hotter coolant in the oil cooler. This helps in decreasing oil warm-up period and therefore reducing the friction losses due to the greater viscosity of the oil at low temperatures.[10]

2.5.9 Driving Cycle

A driving cycle may be defined as a series of vehicle speed values, given as a function of time for a specified duration. Driving cycles are used for quantifying the performance of vehicles on the basis various parameters such as fuel consumption and polluting emissions in a normalized way, so that different vehicles can be compared. They are developed by automotive experts across the globe to represent naturalistic driving patterns. Examples of driving cycles include the New European Driving Cycle (NEDC) and the more recently introduced Worldwide harmonized Light vehicles Test Cycle (WLTC), which is illustrated in figure 2.8 below:

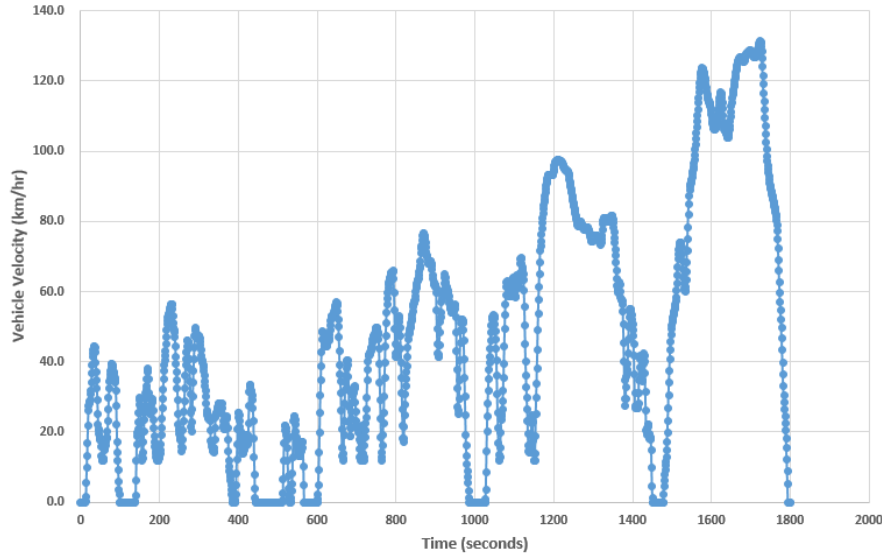


Figure 2.8: WLTC Class 3

2.6 Computer modeling

This section introduces the readers to the modelling platforms, techniques and other computing details relevant to this project.

2.6.1 GT-SUITE

GT-SUITE is a multiphysics platform for simulating different vehicle technical areas. This platform allows integration of different tools under a common interface that can be simulated simultaneously. For detailed information about GT see reference [16].

The modeling in GT-Suite is template based and the user interface used is called GT-ISE. In the Figure 2.9 the templates can be found in the left sided library and after picking the template and define it becomes a object. This object can be dragged into the model map. One of the advantages of GT-Suite is that a single object can be used to create other objects with similar attributes. This can be done by using different names, but all these objects will be under a "parent" object. Thus, anything changed in the "parent" object will affect the attributes of these objects. Once these objects are created and dragged to the model map they can be connected with other different objects to create a flow circuit for example the coolant flow circuit shown in the Figure 2.9. Some of the options in the object can be parametrized. This can be done by assigning a name for this parameter in brackets. [10]

2.6.2 Artificial Neural Networks (ANN)

Artificial neural networks are automated processes and a learning paradigm and inspired from the human biological nervous system. They consist of an interconnected system of neurons that share information among them to produce an output. Each neuron receives a certain number of inputs and sends an output depending on two functions, namely: input function 2.18 and activation function 2.19.

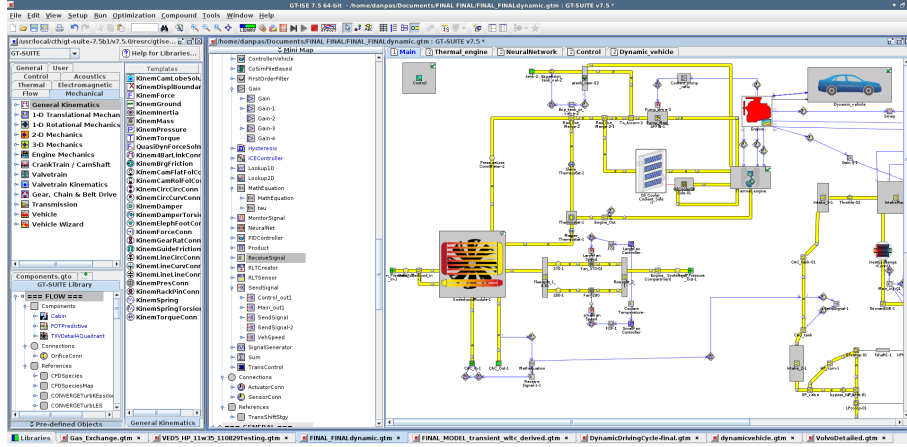


Figure 2.9: GT user interface

$$s = \sum_{j=1}^N w_j x_j + b, \quad (2.18)$$

where N is the number of inputs of each neuron, x the input value, w is the weight of each input and b is the bias, which is a constant value used in case all inputs are zero.

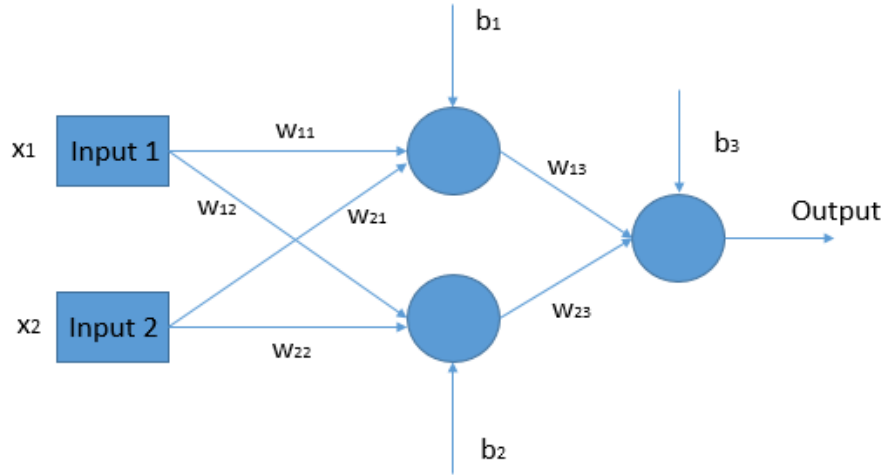


Figure 2.10: Artificial Neural Network of two inputs, three neurons and one output

$$\sigma(s) = \frac{1}{1 + e^{gs}}, \quad (2.19)$$

where g is a constant.

The ANN reads the input values and output values to train itself using an iterative method to compute the indicated weights of each input until a certain minimum error value is achieved between the predicted and the target output values. The weights are optimized by backward propagation of the error during training. An example of a basic ANN is shown in the Figure 2.10 [17]

2.7 Macro

A macro or a "macroinstruction" is essentially a fragment of programming code that contains a sequence of instructions to be executed in order to complete certain routines. The macros for STAR-CCM+ are written in standard Java programming language which enables the user to access to all the programming constructs of that language, such as loops and conditional constructs. They are compiled and executed within the STAR-CCM+ workspace which enables the user to establish two-way communication with any external software for exchanging the values of different parameters. It is highly recommended to use an integrated development environment or an "IDE" for compiling, writing and debugging such macros as they help the users to identify the potential errors in the code. The NetBeans IDE 8.1 has been utilized for the purposes of this project since it is fully compatible with all the instruction libraries of STAR-CCM+ and is recommended by CD-adapco.

2.8 Comma Separated Values File

A Comma Separated Values (CSV) file is a type of delimited text file that stores tabular data (numbers and text) in plain text. Each of the successive fields in a CSV file are separated from the former with a comma (.). CSV formats are best used to represent sets or sequences of records in which each record has an identical list of fields. This corresponds to a single relation in a relational database, or to data (though not calculations) in a typical spreadsheet. The format is very popular in the field of computing and is widely used to pass data with different internal word sizes, for data formatting needs, and so forth. For this reason, CSV files are common on all computer platforms.

Methods

This chapter introduces the readers to the different methods and procedures utilized in this project. It begins with giving an overall view of the various interactions taking place within the entire domain that this project aims to capture. This is followed by the description of the methods adopted for modelling of the 3D and 1D models respectively.

3.1 Modelling Domains

As a first step, the different interactions involving the transfer of mass and thermal energy taking place in the physical environment of our interest, which is the entire vehicle underhood region, are identified and studied. These interactions are depicted in the form of a schematic representation in figure 3.1 below. This study highlighted that the phenomena and systems whose modelling is of primary interest for this project are:

- Buoyancy driven flow in the underhood region of the subject vehicle when it is parked in a quiescent environment after being subjected to a driving routine.
- Its powertrain cooling systems and the various thermal interactions among them.
- Conjugate heat transfer between the external surfaces of the engine solids and their surrounding space in the underhood region.

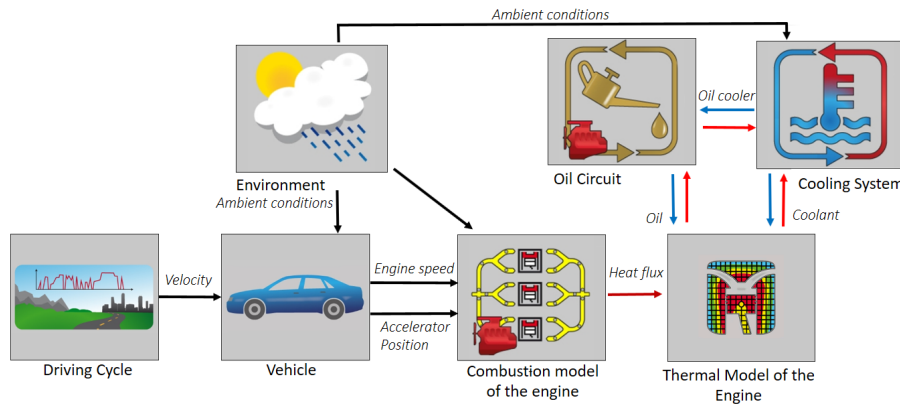


Figure 3.1: Schematic representation of the all the interactions taking place between different elements of the entire modelling domain

Based on their respective nature and complexity, the modelling of these phenomena and system is separated into two major domains where 1D and 3D CFD modelling are required respectively. The buoyancy driven flow in the underhood region is unsteady and turbulent in nature. The properties of the resulting plumes and flow structures are inherently 3D in nature. High spatial and temporal resolution are required for their analysis[?]. Hence a 3D CFD model, built in a commercially popular CFD package called STAR-CCM+ (Version 10.06.009) is employed for capturing the physics of this phenomenon.

Powertrain cooling systems (PCS) consist of a multitude of components whose functioning and mutual interactions are governed by a number of parameters. Transient analysis of these systems essentially requires 1D modelling in order to keep the computational demands within realistic limits. A previously built 1D model for the subject vehicle PCS is upgraded and utilized for their analysis. The commercial software GT-SUITE (Version 7.5) is used for the development of this model. The main functionality of this model consists of running transient simulations with the VED5 engine of the subject vehicle and its powertrain cooling systems (PCS) thereby capturing all the relevant thermal interactions taking place among them.

Finally to capture the conjugate heat transfer (CHT) between the systems represented in these two models, mutual coupling is required to link the data processed by them individually. This aspect of the project will be further elaborated ahead in section 3.4. Hence the division of the physical systems modelled in this project can be summarized in the form of a block diagram in figure 3.2.

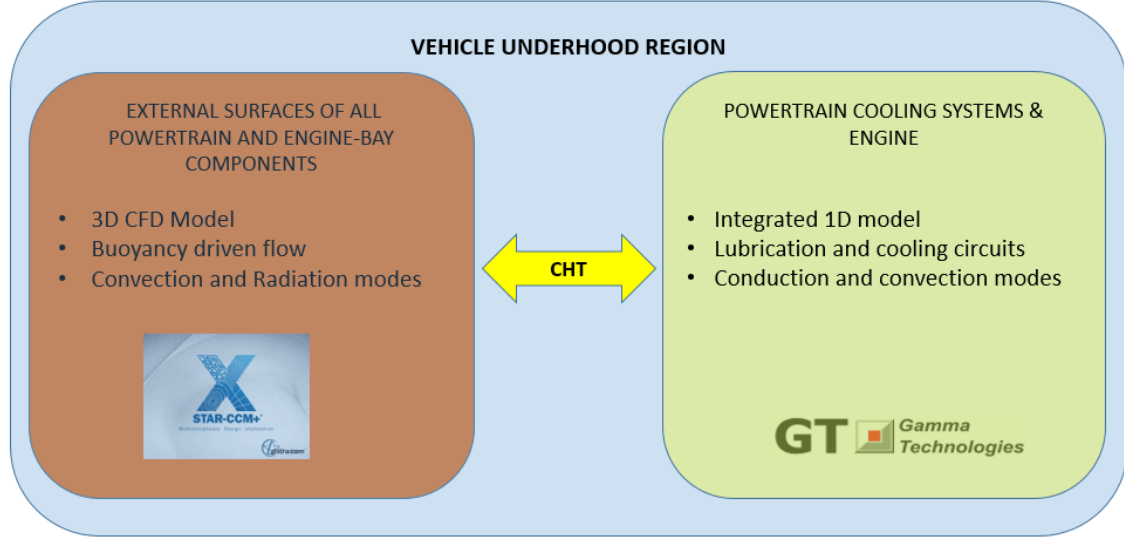


Figure 3.2: Division of modelling domains

3.2 3D CFD Model of Underhood Region

A detailed surface mesh of the complete vehicle provided by Volvo cars is available as an input to this project. Primary preprocessing of this mesh has been done in ANSA to define a computational volume illustrated in figure 3.3. This volume comprises of a cuboidal enclosure of dimensions measuring approximately 2 m(length) X 2 m(breadth) X 1.6 m (height).

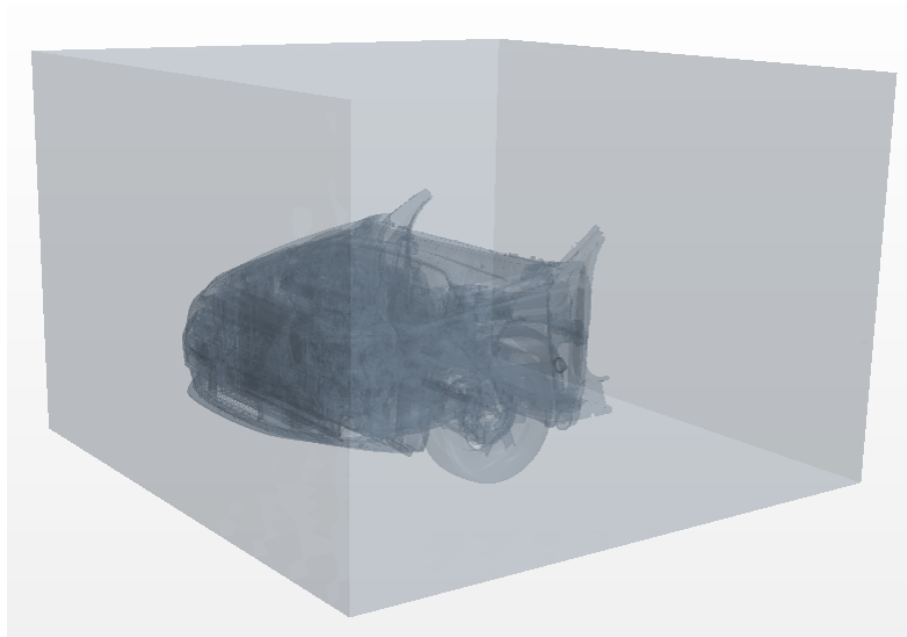


Figure 3.3: Computational domain for the 3D-CFD model defined in STAR-CCM+

It captures the frontal half of the vehicle extending from the grille to the A-pillars, making sure that the entire underhood region and its immediate surroundings are included. This initial representation of the computational volume is then exported to STAR-CCM+. Since the heat transfer within the engine solids and among the powertrain cooling system components is modelled in the 1D model in GT-SUITE, the fluid domain in the underhood region was of primary interest for this model to capture the convective and radiative modes of heat transfer.

3.2.1 Generated Mesh

The region based meshing approach is employed for this model since it is a tried and tested method for such applications [?]. To mesh the fluid domain of the computational volume, the following meshing models in STAR-CCM+ are employed:

- **Surface Wrapper:** Given the presence of multiple intersecting surfaces and gaps in the provided raw mesh, the generation of a well-resolved, water tight surface mesh is required as a first step. The surface wrapper model is used for this purpose as it generated the desired mesh much faster and easily compared to the wrapper available in ANSA, and has been recommended in several published works [?]. It also provides the option of including refinements that are based on curvature, proximity, and individual surfaces out of which the curvature refinement option is utilized in this case. However the resulting surface quality from this model is not optimal. Therefore, it is used with the surface remesher model, as elaborated ahead, to provide a high quality starting surface for the volume mesher.
- **Surface Remesher:** To improve the overall quality of the surface mesh generated using the wrapper model and to existing surface and optimize it for the volume mesh models, the surface remesher is be used to re-triangulate the surface for improving its overall quality and optimizing it for the generation of the volume mesh. The target edge length is specified as 5 mm along with both curvature and proximity refinement for capturing the complex geometries of the underhood region. Localized relaxation in the mesh size is employed for the fluid region surrounding the vehicle components to reduce the unnecessary expenditure of computational resources. The surface growth rate is specified as 1.3. A list of the values specified for the surface mesh is presented in table 3.2 below. The option of aligned meshing is selected in the remesher model to optimize the resulting surface triangulation further.

| Parameter | Value |
|--|--------|
| Target cell size for vehicle component boundary surfaces | 5 mm |
| Minimum cell size | 3 mm |
| Localized target cell size for fluid region | 300 mm |
| Localized minimum cell size for fluid region | 250 mm |
| Surface growth rate | 1.3 |

Table 3.1: Specification of surface mesh generated for the fluid domain

- Advancing Layer Mesher:** The Advancing Layer model, which has been introduced recently by STAR-CCM+, is utilized for generating the volume mesh. It converts the existent triangular surface mesh into a polygonal surface. This polygonal surface is then advanced into the domain volume by extruding it into the region volume to form prismatic cell layers. As a result the dependency between the subsurface and region topology is reduced which allows it to generate a thicker, more uniform layer than the prism layer mesher can achieve. 3 layers of prismatic cells are generated around the external surface mesh which capture the boundary layer, turbulence effects, and heat transfer near wall boundaries. The prism layer thickness is set to 1.3 mm. The remaining surrounding void of fluid region is filled with polyhedral cells. The advancing layer mesher also collapses and splits faces and edges, which helps to improve mesh quality. The final generated volume mesh comprised of approximately 50 million cells. The longitudinal cross sectional view of this mesh is depicted below in figure 3.4.

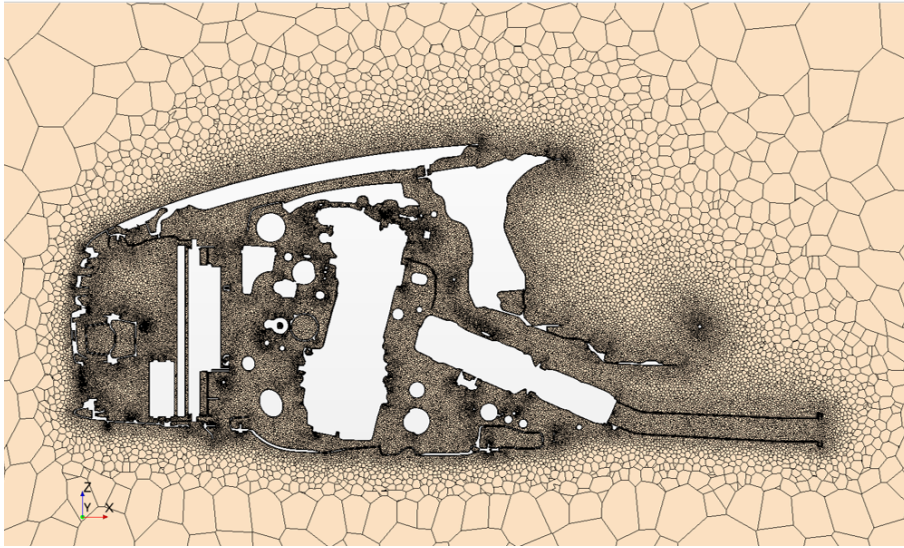


Figure 3.4: Longitudinal cross-sectional view of the generated computational mesh grid

3.2.2 Boundary Conditions

Since this model represents the vehicle standing in a quiescent environment, atmospheric pressure is assigned to the fluid domain. The ambient temperature is set to 20 degrees Celsius. The main interfaces through which the conjugate heat transfer takes place between the engine and the surrounding air are defined as illustrated in figure 3.5.

These include the external surfaces of the following engine solids:

- Engine cylinder head
- Engine block
- Exhaust manifold
- Oil Sump

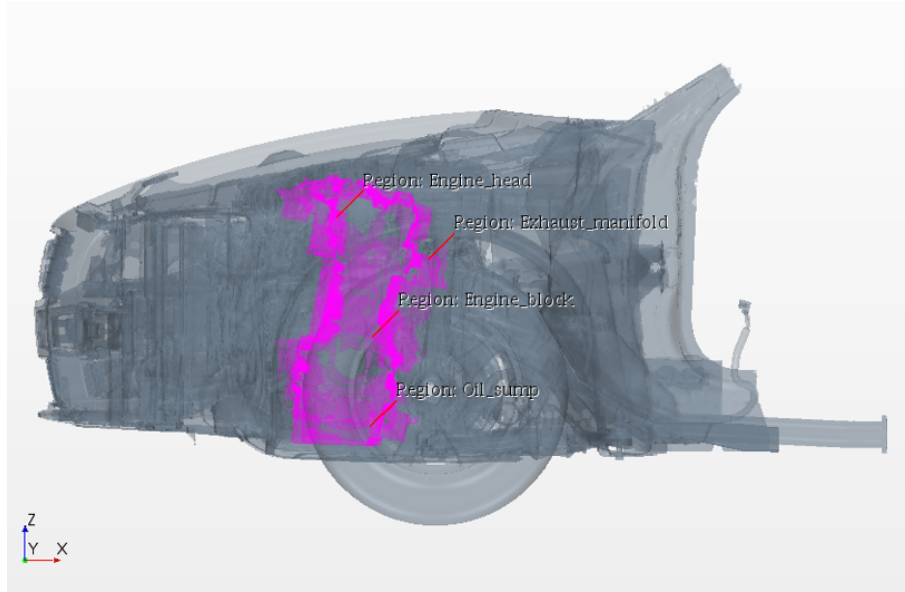


Figure 3.5: Boundaries defined for the exchange of information between the two software

In order to verify the functionality of the model before using it for running coupled simulations, certain verification simulations are performed with the 3D model running in isolation. Nominal values for the temperature are assigned to the boundary surfaces specified above for performing these simulations. These values are given as:

- Ambient : 20°
- Engine block : 100°
- Cylinder Head : 110°
- Oil Sump : 125°
- Manifold : 600°

At the same time, all other surfaces present in the model are modelled as adiabatic components so that while the influence of their geometrical detail on the flow structures is captured, their participation in the thermal interactions taking place in this model is neglected for simplicity. In the final coupled model, the temperature values for these boundaries will be assigned through the data obtained from the 1D model in GT-SUITE in real-time.

3.2.3 Solver Settings

The major setting employed in this model for the solver are enlisted below:

- **Steady state solver:** Owing to the complexity and the number of cells in the computational grid in this model, the steady state solver is employed. Transient simulations with models of such detail are computationally very expensive.
- **Fluid Selection:** All the physical and chemical properties of air are assigned to the fluid surrounding the powertrain components in this region. Ideal gas laws are used for modelling the governing physics for this fluid.
- **Gravity and Radiation:** To effectively capture the flow dynamics and the modes of heat transfer, gravity and gray-thermal radiation models are included. The size of the radiation patches are set to a face proportion of 30 percent, implying that the mesh cell size is approximately 30 percent of the patch size.
- **Realizable $k-\varepsilon$ Turbulence Model:** STAR-CCM+ offers three different approaches for the definition of turbulent eddy-viscosity, namely: Spalart-Allmaras, $k-\varepsilon$ and $k-\omega$ models. A linear relationship between the strain-rate and the Reynolds-stress tensor, better known as Boussinesq assumption, is assumed in all of them with the proportionality referred to as "turbulent eddy-viscosity". Each of these model are suitable for specific applications. The standard $k-\varepsilon$ model offers a good compromise between robustness and accuracy for a larger range of flow configurations. Compared to other options such as the standard $k-\omega$ and the SST $k-\omega$ model, it does not have the tendency to overpredict the turbulence levels in regions with large normal strain, like stagnation regions and regions with strong acceleration, which is existent in the boundary layer around the hot engine solids. The realizable $k-\varepsilon$ model is substantially better than even the standard $k-\varepsilon$ model and can generally be relied upon to give results that are at least as accurate.
- **Wall Treatment:** The $k-\varepsilon$ model is implemented in STAR-CCM+ with a Two-Layer All $y+$ wall treatment, which activates it to be used with fine meshes that resolve the viscous sub-layer. This wall treatment is a hybrid approach that attempts to emulate the high $y+$ wall treatment for coarse meshes and the low $y+$ wall treatment for fine meshes and is also formulated in order to produce reasonable solutions for meshes of intermediate resolution, i.e. when the wall-cell centroid falls within the buffer region of the boundary layer, where $5 < y+ < 30$.

3.2.4 Results from Standalone Simulations

The steady-state simulations with the 3D model achieved convergence after approximately 4000 iterations while utilizing a total of around 3000 core-hours each. The resultant temperature field around powertrain components showed high temperature distribution in the regions above the exhaust manifold and cylinder head as the plumes rising from these hot surfaces spread around in the engine bay. This distribution is illustrated in the figure 3.6.

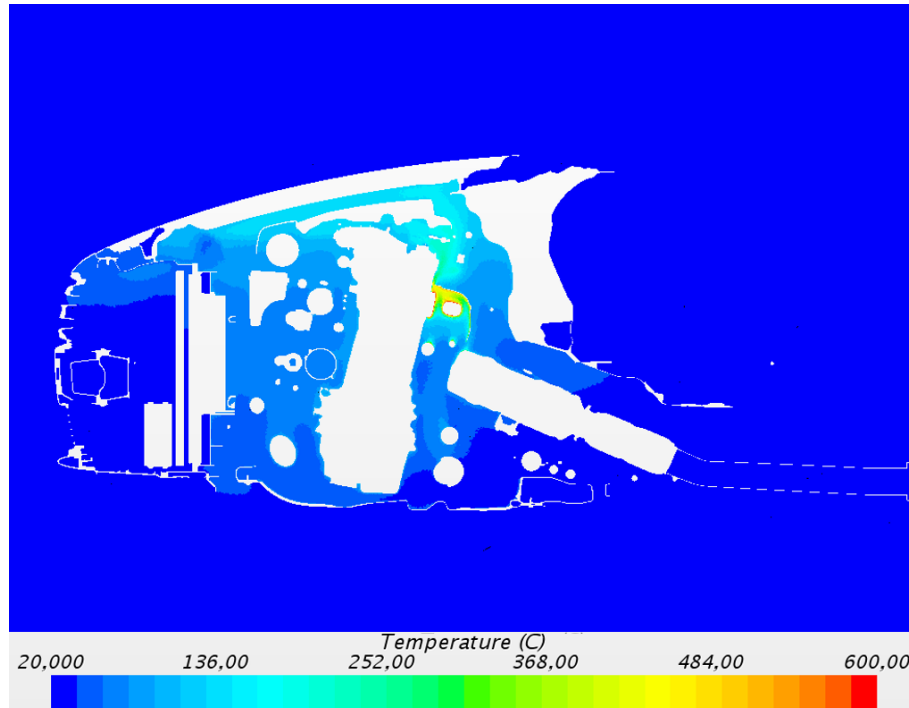


Figure 3.6: Temperature profile across the longitudinal-cross section obtained from steady state simulation with nominal boundary conditions.

3.3 1D Model of powertrain cooling system

The 1D model of powertrain cooling systems available as an input to this project has been built by a 6-member project group during the autumn semester of 2015 at Chalmers University of Technology. It consists of:

- Simplified thermal engine model
- Complete powertrain cooling system
- Mechanic vehicle model

3.3.1 Upgrading

To meet the targets for this thesis work, some necessary changes and upgrades have been made to the given model.

Convergence

The first upgrade carried out was the increase in number of time steps converged in the GT-Solver. When the first simulation was run for the given model using the WLTC (Worldwide harmonized Light vehicles Test Cycle) some convergence problems occurred. This is explained because the drive cycle used in this work, WLTC drive cycle has a higher level of fluctuations in the speed compared to the sample drive cycle provided for the purposes of that project. One of the most critical parts of the cooling system is the pump since at high engine

speed a great amount of energy is transmitted to the fluid. Because of this the relaxation factor of the pump was reduced ten times. This change helped to make almost 100 % of the time steps converge but with longer computational times as the relaxation factor is a parameter used to dampen the solution in iterative solvers and cut out oscillations.

Thermal Nodes

The thermal masses of the given thermal engine model are modelled using *ThermalMassOld* templates in GT-suite to represent the different parts of the engine such as the main block, cylinders and the cylinder head. This template is obsolete and has limited connectivity. All *ThermalMassOld* objects were replaced by their successor *ThermalNode* objects that are basically the same but allow a greater connectivity option for the user.

3.3.2 Oil Circuit

Lubrication Detailed Model

A detailed and complex lubrication model was given by Volvo. This model was designed to compute accurately the pressures over all bearings in the engine and its lubrication.

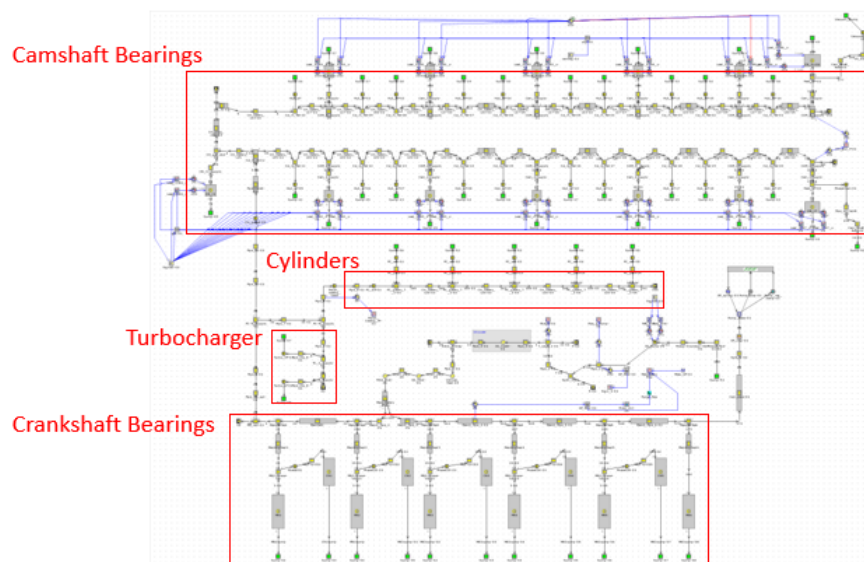


Figure 3.7: Detailed Lubrication Model

Simplification

Because of the complexity of the model, the required simulation time was excessive. As a first step to overcome this hurdle, the number of flow splits have been reduced all redundant objects were either eliminated or replaced. However it is important keep the mass flow of each branch of the oil circuit in order to know the amount of oil that will be in contact with the hot parts in the engine.

To achieve this aim, the *PressureLossConn* template is utilized to impose pressure drops and remove each of the bearing in the detailed model. This template provides the option to permit the creation of a function relating the Reynolds number for the flow through the circuit and the desired pressure drop across it.

The whole lubricated model has been simplified into seven branches that can be classified by:

- Main Bearing branch number 1
- Main Bearing branch number 2
- Main Bearing branch number 3
- Cylinders branch

- Turbocharger branch
- Exhaust Camshaft branch
- Intake Camshaft branch

As mentioned above, it is necessary to maintain the flow rates, and thus the behaviour of the circuit is captured at diverse working points by running several cases. Pump speed and the temperature of the wall of pipes are primary parameters that are modified in each of these cases.

All pressures, densities and mass flows at the beginning of each branch are stored to compute the pressure loss coefficient according to equation given below:

$$\Delta P = \frac{1}{2} K \rho \left(\frac{\dot{m}}{\rho A} \right)^2 \quad (3.1)$$

ΔP = difference in total pressure across the connection (total upstream - total downstream)

K = loss coefficient defined in this template

\dot{m} = mass flow rate through the 'PressureLossConn'

A = reference area

ρ = density of the fluid

Since all branches of the circuit end at the oil sump ,in our case the value of ΔP is the difference between the pressure at the beginning of the branch and the atmospheric pressure and the reference area is the section area of the pipe. At the same time the loss coefficient was computed, the Reynolds number of to the branch was determined in order to create a relation between both values. This procedure was carried out for each of the branches. For instance the scatter obtained in the main bearing branch number 1 is 3.8

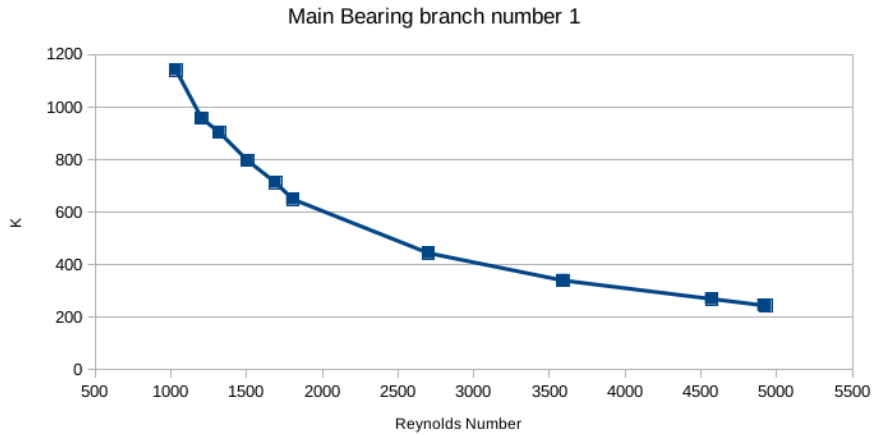


Figure 3.8: Reynolds Vs Loss coefficient

In order to make the model faster, some pipe objects were simplified to reduce the number of pipe objects while maintaining the total length of the circuit. It is important to mention why loss coefficient method was chosen among other different ways that were available to impose a pressure drop. The main advantage of this method is that the *PressureLossConn* template can take into account the temperature of the fluid crossing through it, being more specific the density and viscosity of the fluid. It is known that viscosity and density of fluids change with the temperature and oil is no exception to this. Since the oil flows at distinct temperatures through the lubrication circuit, the pressure drop due to the pipes, bearings and nozzles will be also distinct.

The way that *PressureLossConn* template reacts to these changes is by taking into account the Reynolds number, since by definition Reynolds number measure the dynamic viscosity μ , density ρ and velocity of the fluid.

$$Re = \frac{\rho v L}{\mu} \quad (3.2)$$

As a result, once all *PressureLossConn* were computed and the pipes were simplified, a fast and temperature dependant model was obtained. This model is able to keep the same mass flow rate as the detailed model for a transient simulation.

Identifying the main heat transfer interfaces

In order to capture the oil warm-up inside the engine three different transfer heat areas where the oil is in contact with the hot parts of the engine were considered. The most relevant of them are these where the oil is in contact with the cylinder liners to provide lubrication . When the piston moves throughout the cylinder liner, it generates a high friction levels that need to be reduced. Pressurized oil is used to overcome this by creating a thin layer between the piston and the liner. Due to the high temperature of the combustion chamber and friction produced the oil warms up. Two areas related to the piston lubrication are seen in Figure 3.9. The blue area represents the bottom part of the piston and the orange area represents the surface of the liner. Both areas were measured and introduced into model in order to compute the heat transferred to the oil.

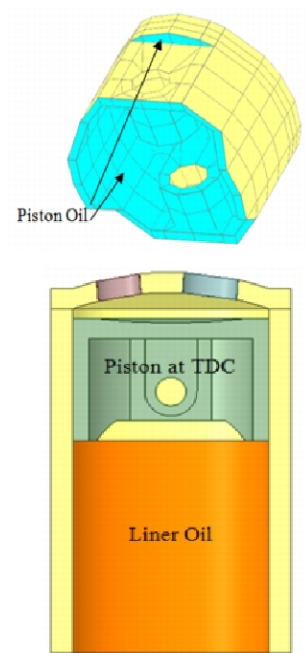


Figure 3.9: Areas related to the piston lubrication

Another relevant heat transfer area is the bottom part of the engine block. This surface was selected since the oil used to lubricate the main bearings is splashed due to the spinning motion of the crankshaft. Since the heat transfer area of this phenomenon is very difficult to select accurately, an estimated area was measured in its place. This area selected is shown highlighted in pink in the Figure3.10.

The third heat transfer area considered was the base of the cylinder head. This area was selected since the oil to lubricate the camshaft has to cross through this hot surface before returning to the oil sump and hence there is a heat transfer between them. The are measured in ANSA is shown in pink in the Figure3.11. The cylinder head is shown cut in order to see the surface clearer.

The final simplified oil circuit is shown in the Figure3.12. Two different *Flowsplit* objects were created to compute the heat transferred by the thermal engine. One represents the area of the Figure3.11 and it is the upper grey box in the Figure3.12 that is connected to the camshaft branches of the oil circuit, and the other is the grey box situated in the middle of the Figure3.12 which represents the heat transferred through the measured surfaces 3.9 and 3.10.

Calibration

The different experimental values of the temperatures achieved in the oil sump are available in the data provided by the Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren institute through the experiments carried



Figure 3.10: Bottom area of the engine block

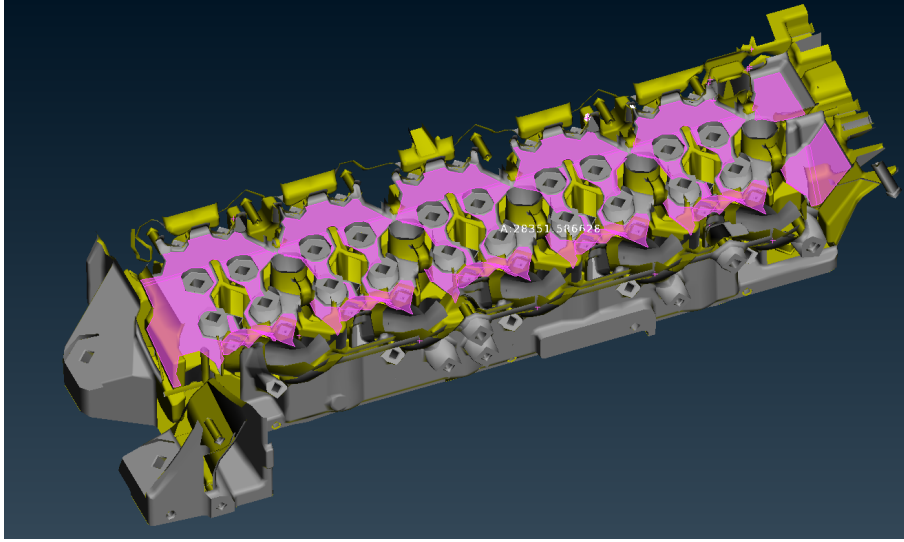


Figure 3.11: Cylinder head surface measured

out by them on Volvo's behalf. These values have been recorded on a test-rig using the same engine used for this project. In total 35 cases were recorded in the test-rig varying the velocity of the engine from 750 rpm to 400 rpm and the position of the accelerator from 20% to 100%. The main aim of the calibration of the predictive oil circuit was to match the same temperature in the oil sump of the model with the experimentally measured temperature. For the calibration of the model 35 steady cases with the same boundaries that in the experimental cases were run. In order to achieve the same temperature in the predictive oil circuit that in the experimental data, the value of the heat transfer coefficient 2.2 was modified at each steady case until the desired temperature was obtained.

The heat transfer coefficient is not a constant value, hence it was necessary to develop a relation between the value of the heat transfer coefficient and the mass flow rate of the oil that was flowing through the oil circuit since it is a parameter in which the heat transfer coefficient highly relies on. Comparing all the heat transfer coefficient computed for each case in an excel file, a function, namely a power Equation 3.3 was created to relate the heat transfer coefficient and the mass flow rate through the oil circuit as it was the function that better followed the trend of the values computed at the steady cases.

$$htc = 1.05712 \times 10^{-5} \dot{m}^{1.90745}, \quad (3.3)$$

where htc is the heat transfer coefficient and \dot{m} is the mass flow rate of the oil.

The complete oil circuit connected to the thermal engine masses is shown in the Figure 3.13

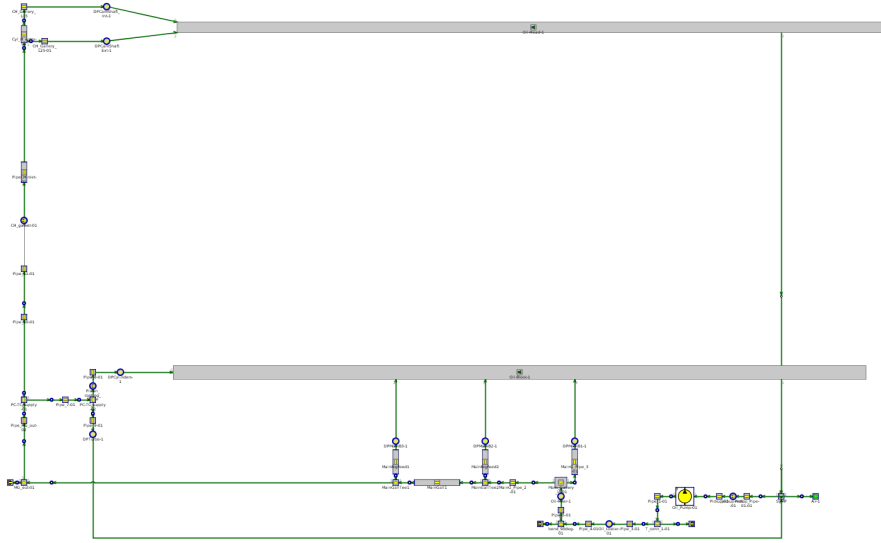


Figure 3.12: Simplified Oil Circuit

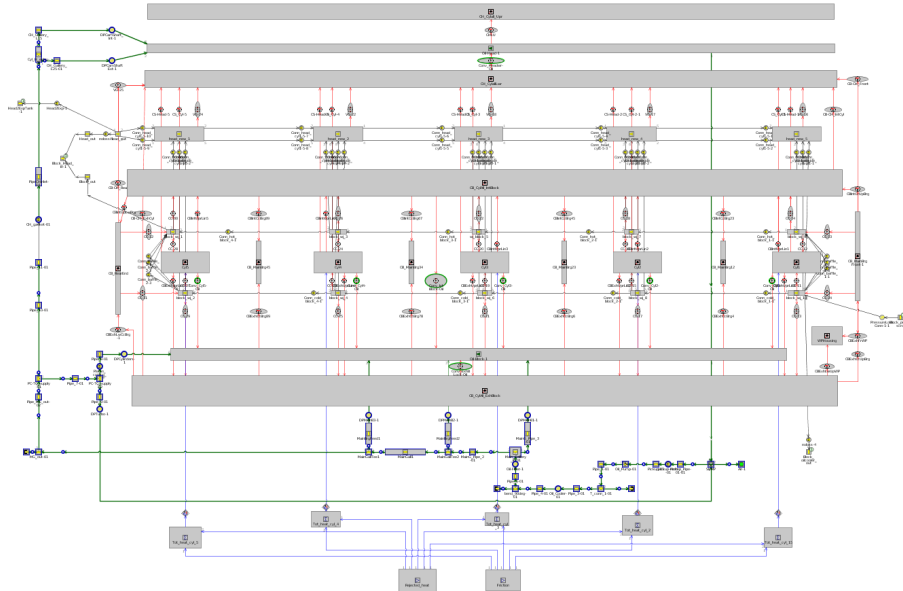


Figure 3.13: Complete Thermal Engine

Validation

The validation for transient simulations of the oil circuit has been done by comparing the temperature computed in the oil sump using the predictive oil circuit with experimental data provided by Volvo from a drive cycle where the temperature of the oil sump were recorded. As it is shown in Figure 3.14 the model overpredicts temperature in some periods. However, the conclusion obtained from this validation is that it was possible to ensure that the oil circuit predicts accurately enough the temperature achieved in the oil sump for transient simulations.

3.3.3 Mean Value Engine Modeling

The mean value engine is used to model an engine using a map-based cylinder. The mean-value cylinder model uses maps to characterize the distribution of fuel energy and cylinder air flow. This provides a model that executes more rapidly than a detailed gas exchange model, as the combustion process is not predicted. Mean engine value models are typically used in simulations where computation speed is important and detailed

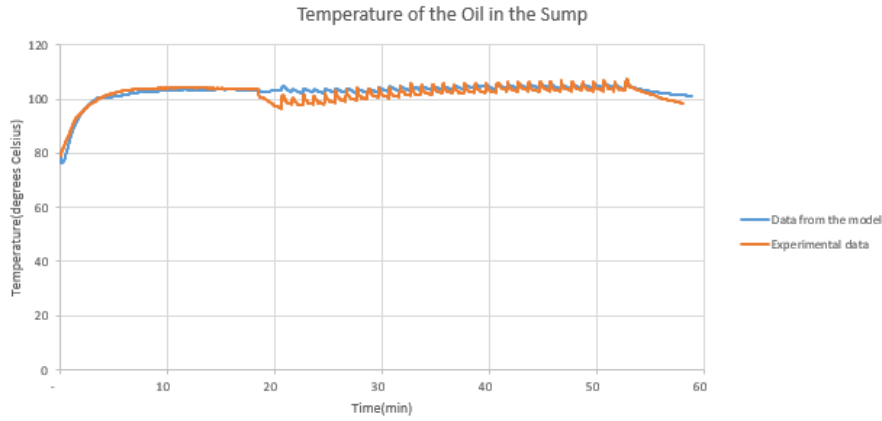


Figure 3.14: Comparison between experimental and computed data

characterization of engine processes is considered relatively unimportant.

There are three parameters entered into a mean value model that are used to characterize the engine cylinder performance. The first is the volumetric efficiency of the cylinder, that is used to compute the mass flow of air through the cylinder that will be imposed throughout the simulation. The second is the indicated efficiency, that is used to quantify how much of the fuel energy contained in the cylinder is turned into work done on the piston. The final parameter is the exhaust energy fraction, that is used to compute how much of the fuel energy is employed to warm up the exhaust gases. Then the remainder of the fuel energy that is not used for indicated work or exhaust energy is assumed to be lost through heat transfer [18].

Using a mean value engine instead of a *EngineState* template which was used to model the engine in the previous project, a greater engine performance accuracy is obtained at transient simulations. Also the mean value engine offers the possibility of calculating the exhaust manifold temperature, which is one of the hottest parts in the underhood and hence must be considered in natural convection cooling.

Design of Experiment Parameters

As mentioned above, the mean value engine is a map-based cylinder so it was necessary to create a cylinder performance map and a DOE (Design Of Experiment) was prepared. In order to characterize cylinder performance under numerous situations, a Latin hypercube ¹ DOE with 4500 different cases was prepared and the simulations were run using the detailed gas exchange model provided by Volvo using the following eight parameters with the mentioned ranges:

- Intake manifold pressure [1,2.6](bar)
- Intake manifold temperature [273,450](K)
- Start of injection timing [-26,1](deg)
- EGR valve aperture diameter [0,26](mm)
- Engine speed [750,4500](rpm)
- Exhaust manifold pressure [1,4.6](bar)
- Ambient temperature [273,311](K)
- Fuel injected per cylinder per cycle [0,70](mg)

¹Latin hypercube: "It is a method for ensuring that each probability distribution in the model is evenly sampled which at first glance seems very appealing" [19]

Artificial Neural Network

After carrying out all experiments and storing the results, it was necessary to create a look-up process that predicts basing on the above inputs each one of the four mean value engine required parameters. To accomplish this purpose, GT-SUITE provides the option of training a Neural Network, introducing stored data from DOE simulation as inputs.

Comparing results of three different methods, Self-Organizing Local Linear, Radial Basis Neural Network and 3-Layer Feed Forward Net and using lower RMS(Root Mean Square) error criteria a 21-neuron 3-Layer Feed Forward Net was trained for volumetric efficiency parameter, a 26-neuron 3-Layer Feed Forward Net for indicated efficiency parameter, a 26-neuron 3-Layer Feed Forward Net for exhaust energy fraction parameter and a 21-neuron 3-Layer Feed Forward Net was trained for the friction mean effective pressure. A diagram of the four neural networks is shown in the Figure 3.15

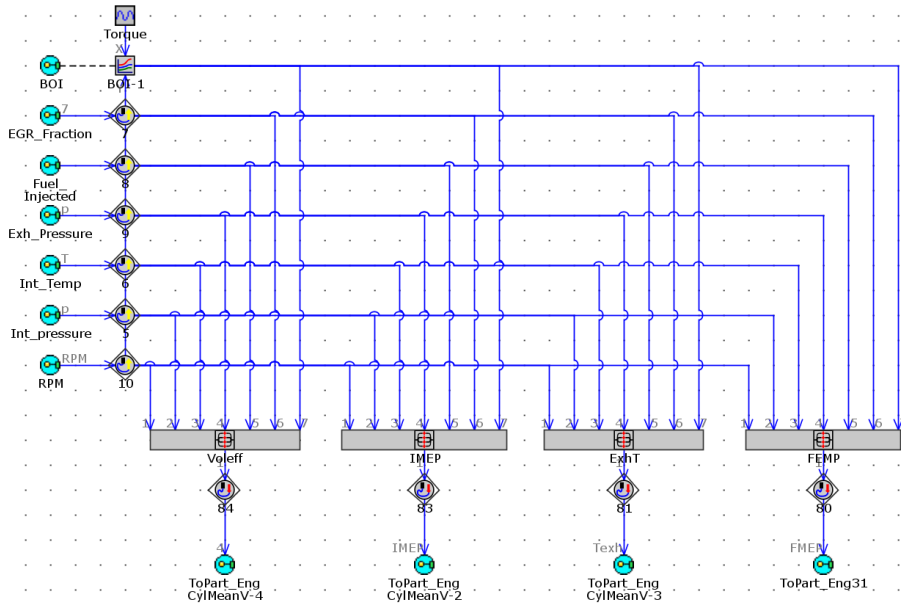


Figure 3.15: Neural Network

Simplifying the Flow System

Because of the mapped nature of the mean value engine model, the flow is steadier than a detailed engine model. Consequently the air handling system may be simplified to reduce the simulation run-time. The intake and exhaust systems may each be lumped into larger equivalent volumes. [18]

Taking these factors into account, all of the pipes before the compressor are simplified into one *Flowsplit* conserving the total volume and surface. The same procedure is followed for the intake manifold, exhaust manifold and the pipes before the turbine.

Calibration

The simplification of the air handling system has a couple of effects to be considered. First the pressure drop reached through the air circuit in the simplified model is lower because of the lack of pipes. This make necessary to impose a pressure loss to the system through a *OrificeConn* template with a calibrated diameter. The calibration of the diameter is carried out comparing the maximum pressure drop of the system, which is achieved at the maximum engine load case and matching both values. Second, the heat transferred between the gas and the simplified volumes are less than in the detailed model due to the reduced heat transfer area and the velocity reached by the flow in the simplified model. This heat transfer difference can be neglected in the intake manifold since the temperature of the air flowing through is not high enough to affect the value of the heat transferred to the wall of the pipes . However the heat transfer in the exhaust manifold was re-adjusted through a heat transfer multiplier until the same exhaust temperate was reached.

Validation

The validation of the mean value engine was done by comparing values obtained in steady simulations in the detailed gas exchange model and the values predicted by the mean value engine. The first value compared was the brake power given by the engine. The Figure 3.16 shows the values of the break power obtained using the detailed model and the predictive model. Comparing both results it concludes that the predicted brake power matches satisfactorily.

The second value compared was the temperature of the exhaust manifold. The Figure 3.17 shows as the previous figure the results obtained using both models. In this case it distinguishes a overpredicted temperature by the predictive model been the maximum error between both values 9 %.

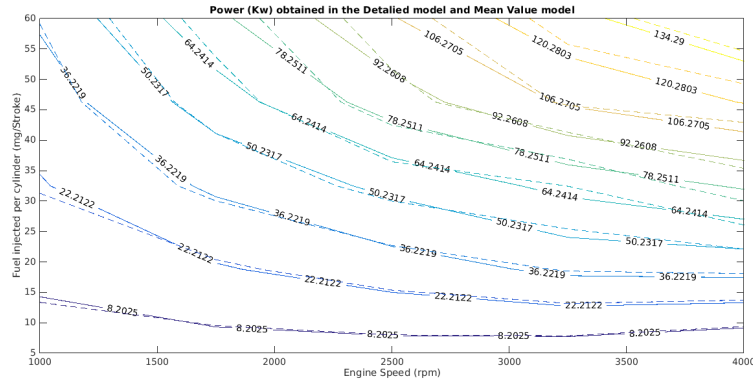


Figure 3.16: Comparison of the brake power computed by the mean value model and the detailed gas exchange model

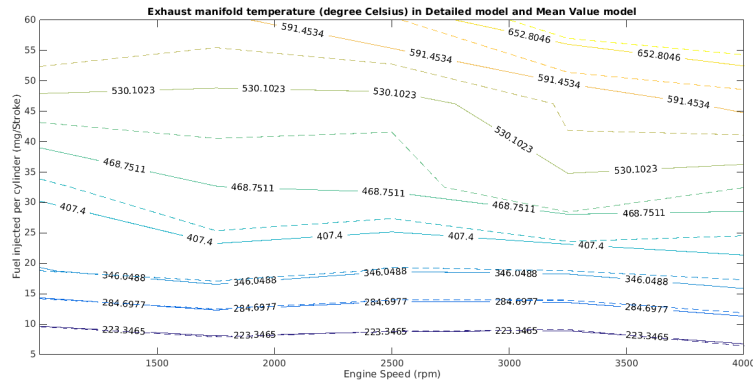


Figure 3.17: Comparison of the exhaust manifold temperature computed by the mean value model and the detailed gas exchange model

3.4 Final Coupled Model

Along with the individual development and verification of the 1D and 3D models, potential methods for coupling them are examined and evaluated, leading to the development of the final coupled tool. This section describes the final stages of development of the coupled simulation tool that is desired as the end-product of this project.

3.4.1 Coupling Mechanism

The two models developed on the 3D and 1D platforms capture the physics and phenomenon occurring in two different domains between which there is no mass transfer. The 3D model of the underhood region captures the various physical phenomena that occur for the air surrounding the powertrain components in the engine bay. On the other hand, the 1D model represents the interactions among the powertrain cooling systems themselves. In order to model the conjugate heat transfer in the engine bay, it is essential to capture the interaction between the hot surfaces of engine solids and the boundary layer of air surrounding them. This is achieved through the exchange of the values of temperatures and heat transfer rates between them in the following manner:

- First, the temperature values for the engine solids are calculated by the 1D model and imposed on the corresponding boundaries defined in the 3D model.
- The 3D model then computes the rate at which heat is lost from the surface of these engine solids through convection and radiation and feeds their values back to the 1D model.
- The 1D model processes utilizes the data fed from the 3D model and updates the temperatures of the engine solids subsequently. This cycle is subsequently repeated for the entire duration of the simulation. An illustration of the same is shown in figure 3.18.

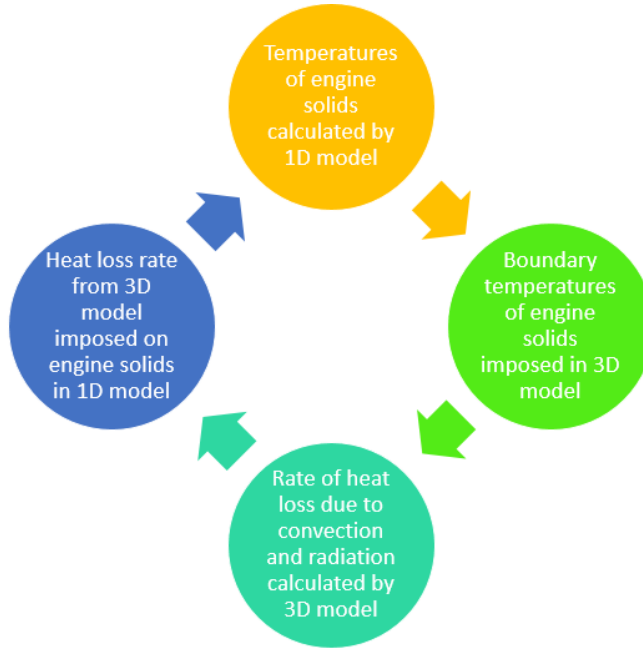


Figure 3.18: Comparison of the exhaust manifold temperature computed by the mean value model and the detailed gas exchange model

For running co-simulations of the kind required for this project, i.e., where scalar data, i.e., the temperature and heat transfer rates, have to be communicated between two different software, GT-SUITE offers two primary methods. These are:

- Using an object from the GT-SUITE library, known as **CoSimFileBased 3.19**. This template enables co-simulation of GT-SUITE with any third-party software through communication input/output data via intermediate text files in comma separated values (.csv) format. Two separate .csv files are used to transfer scalar values that are exported from GT-SUITE and imported by the external tool.
- **Using blocks in simulink** to act as the bridge between the two models for exchange of data.

The first option is adopted in this project over the second one. This is done to minimize the complexity of the final simulation tool by eliminating the need to have three different software running at the same time instead of two. This also reduces the risk of compatibility issues in the future. The first option offered a simpler and more straightforward method while reducing the number of interfaces that had to be set-up. A macro 2.7 developed in Java computing language by one of our supervisors, Blago Minovski, is utilized for enabling the communication from the 3D model's end. This macro can be used to read and print values of various parameters in the STAR-CCM+ workspace to any .csv file 2.8 available.

Also, since STAR-CCM+ has not yet adopted the Functional mockup interface standard, hence it is not possible to establish communication between the models in GT-SUITE and STAR-CCM+ through this method and hence it is not considered in this regard.

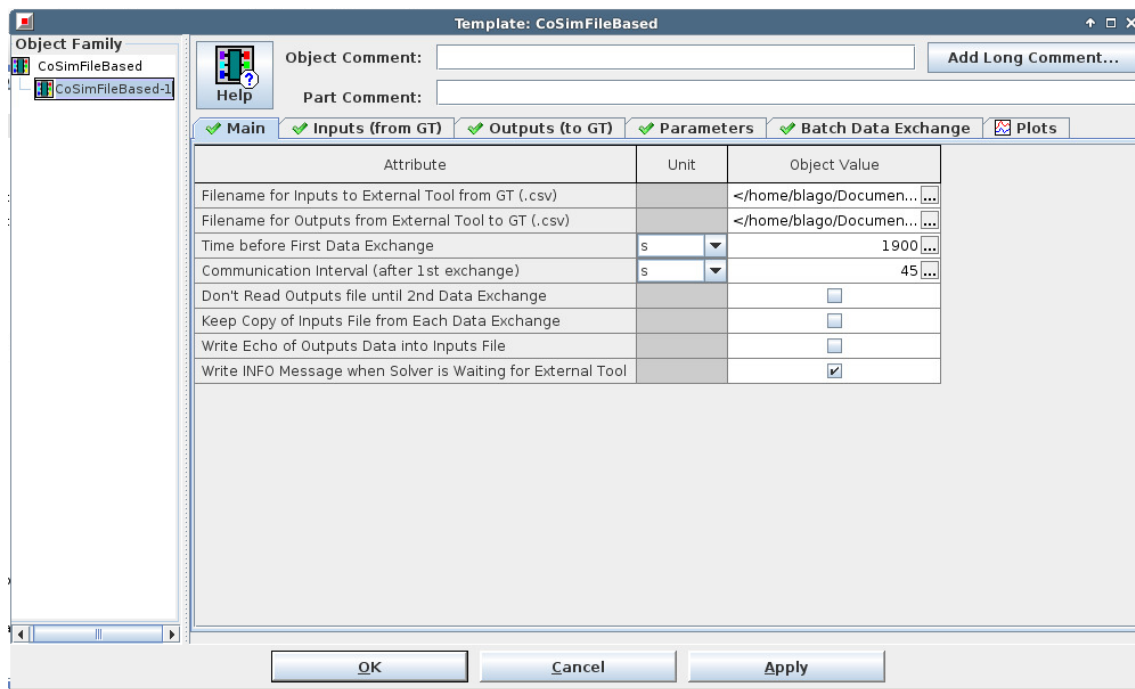


Figure 3.19: CosimFileBased template in GT-SUITE

The data transfer between the different platforms through the described method can be summarized as follows:

Data exported from GT-SUITE:

- Surface temperatures of engine head, engine block, oil sump and the exhaust manifold
- Engine state, i.e., whether the engine is running or is switched off.
- Elapsed physical time for the simulation.

Data exported from STAR-CCM+:

- Heat transfer rate from the surface of the engine head, engine block, oil sump and the exhaust manifold to their surrounding environment

Data imported by GT-SUITE:

- Heat transfer rate from the surface of the engine head, engine block, oil sump and the exhaust manifold to their surrounding environment

Data imported by STAR-CCM+:

- Surface temperatures of engine head, engine block, oil sump and the exhaust manifold

The data regarding the engine state is utilized by the Java macro to implement a computational strategy for the coupled model which will be further elaborated in the section strategysect ahead.

3.4.2 Quasi-Stationary Approximation

The process of heat dissipation from the surface of engine solids in the underhood region is essentially a transient phenomena. The rate at which heat is lost will be a function of time and reduce constantly as the temperature of engine solids goes down. However, owing to practical reasons relating to available computational resources, the transient solver cannot be used with the 3D CFD model in STAR-CCM+. Hence it is imperative to use an alternate method for capturing this process with the coupled tool. The CosimFileBased template in GT-SUITE that is used for coupling the two models offers the option of controlling the interval of physical time after which data is exchanged between them. By employing the steady state solver in the 3D CFD model and keeping the boundary temperatures constant during these intervals, an approximation is made. This approach considers the heat dissipation process to occur through a series of "small steady state processes" whereby the temperatures of the engine solids stay constant. The shorter the interval will be for data exchange between the two software, the more transient will be the nature of the simulation will be in nature. Different exchange rates varying from once every 5 seconds to once every 60 seconds are compared for the developed tool in the "Results" section.

3.4.3 Coupling Strategy

One of the biggest advantages of simulation models over physical testing methods is their ability to analyze a myriad of scenarios while spending minimal physical resources for the changes made. But to analyze these scenarios with the help of simulations and for any model to be ultimately useful for the industry, reduction in computational time is highly desired as long as the results are accurate enough. In order to achieve this, an efficient strategy is desired which can eliminate redundancies and wastage of computational resources.

As described earlier, the Java macro reads the parameters in the workspace of the 3D model, more specifically, from the temperature reports set-up in the STAR-CCM+ model and also assigns the values of the temperatures to the boundaries ?? defined in it. It is upgraded for implementing an active coupling-decoupling algorithm which saves computational resources. This is done by splitting the drive cycle into two main phases on the basis of the engine state, i.e when the engine is on and off respectively, and using different simulation routines for them. Further elaboration of the same is done below by considering the described phases as two separate cases:

- **Case 1, Engine is switched on:** For the periods when the engine is running, forced convection occurs in the engine bay due to the incoming air from cooling inlets and the working of the cooling fan. The heat lost due to natural convection is insignificant in such a scenario and performing simulations with the 3D model would hence be futile. Therefore, the macro detects the engine state from the data exported by GT-SUITE and skips the simulations with the 3D CFD model. Instead, nominal values ranging between 500 to 1000 watts are imposed for the rate of heat loss from the engine solids. Thus, the computationally expensive 3D model is effectively paused, resulting in reduced demand for computational resources.

- **Case 2, Engine is switched off:** This case refers to the scenarios when the car is standing in a quiescent environment and the engine has been switched off. Buoyancy driven convection and radiation modes of heat transfer dominate under such conditions and simultaneous coupling between the two models is kept active. A convergence criteria is set for the 3D simulations to ensure that they stabilize before the values for the heat transfer rate are exported to GT-SUITE. A range of ± 2 Watts for the variation in values of heat transfer rate between successive iterations of the 3D model is assigned for this purpose. Once the fluctuation of these values for all the engine solids are found to fall within this range, the simulation is deemed to have converged and the values are exported. To minimize the possibility of numerical errors, a minimum of three iterations are enforced and the convergence checks are performed only after the third iteration for any instance of simulation with the 3D CFD model.

This strategy is better illustrated with the help of the schematic depicted in figure 3.20 while the complete code for the Java macro is provided in the appendix A.1

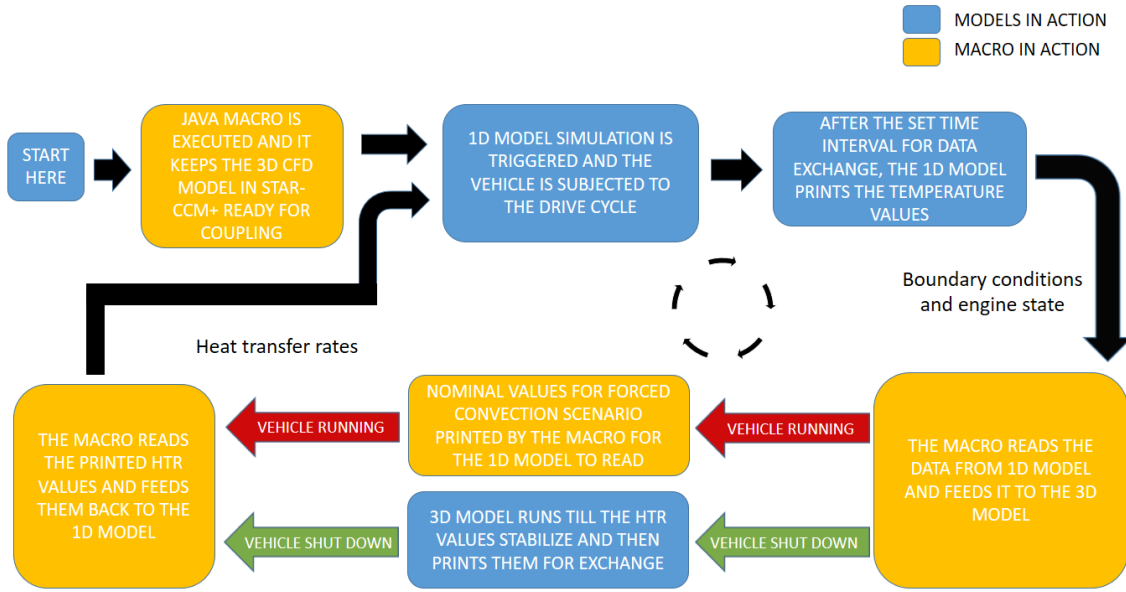


Figure 3.20: Schematic of the coupling strategy employed by the Java macro for the exchange of data between the two software

3.4.4 Customized Drive Cycle

Finally, for the analysis of the coupled simulation tool, a customized drive cycle is constructed. This cycle, illustrated in figure 3.21, utilizes the Class 3 Worldwide harmonized Light vehicles Test Cycle (WLTC), which is presently the standard for passenger vehicles with a power to weight ratio greater than 34 kW/Tonne, s in the case of our subject vehicle. It consists of an initial driving phase followed by a long shut-down phases of approximately 2 hours. The cycle ends with another similar sequence albeit with shorter driving and shut-down phases. It has an overall duration of 5 hours. The primary motive behind the construction of this cycle is to analyze the stability and robustness of the coupling tool during both driving and long shut-down phases, as well as the instances of transition between them. The vehicle is subjected to this cycle and the temperature of the oil in the engine sump is monitored over time.

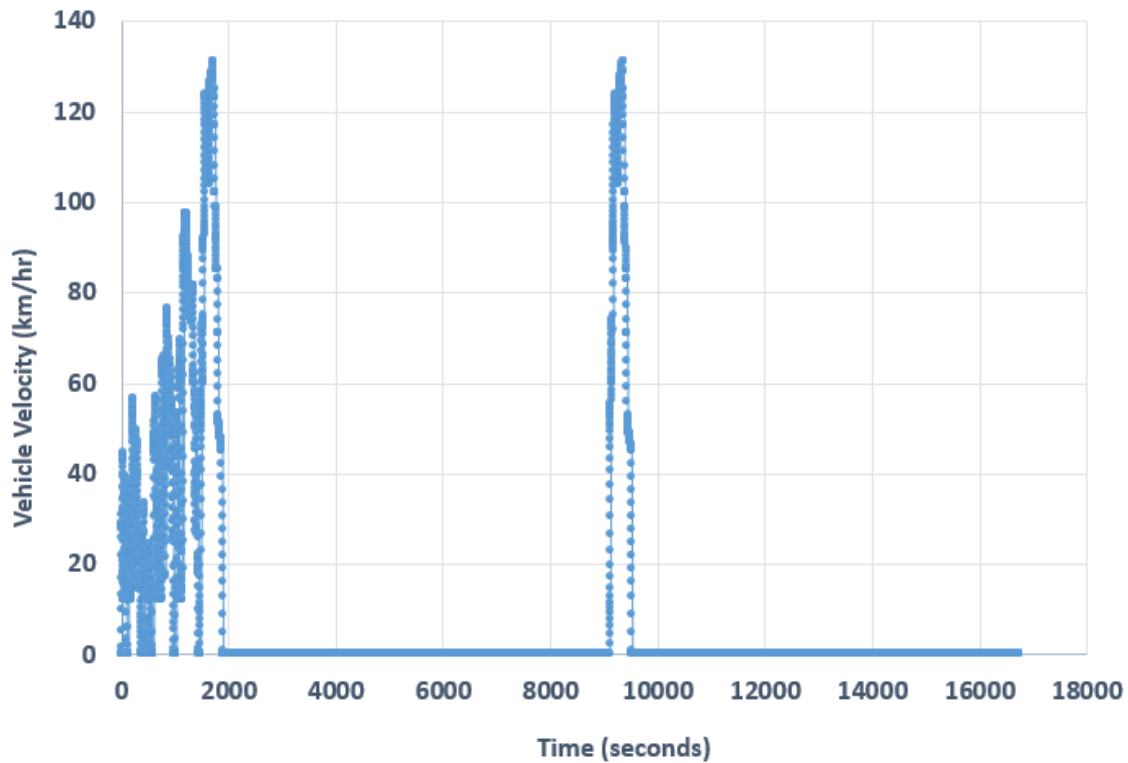


Figure 3.21: Customized drive cycle constructed for analysis of the coupled simulation tool

3.4.5 Ambient Conditions

For the simulations performed with the coupled model, a set of ambient conditions are employed as summarized in table 3.4 below:

| Parameter | Value |
|---|---|
| Outside Ambient Temperature (Car Running) | 20°C |
| Garage Ambient Temperature (Car Parked) | 40°C |
| Ambient Pressure | Atmospheric Pressure (For all cases) |

Table 3.2: Ambient conditions employed for coupled simulations

Results and Discussion

4.1 Predicted Oil Temperatures

The final coupled model is able to predict the variation of the mass averaged temperature of oil in engine sump over the duration of the defined custom drive cycle 3.21. A plot depicting this variation, when an exchange interval of 15 seconds was used for the coupled model, is presented in figure 4.1. The oil temperatures can be observed to fall after the two instances of engine shut-down occurring after 30 and 160 minutes respectively after the start of the cycle. This fall is expectedly sharp immediately after the shut-down and reduces at a lower rate later on. However the rate of fall in temperature is almost constant after a point while the common intuition suggests that it should be rather digressive in nature. This is primarily due to the scale of variation in the temperature values which is not so large, and hence, there is a lesser fall in the rate of temperature loss. Another possible explanation for this is the steady state approximation, that had to be employed with the 3D model. This approximation requires the heat transfer rates to be constant for a series of intervals of time instead of varying continuously with time. Also, the convergence criteria used for the 3D CFD model, whereby the the heat transfer rates are allowed to vary in the range ± 2 Watts, could have allowed for errors in the results and stricter tolerances for the same can be explored.

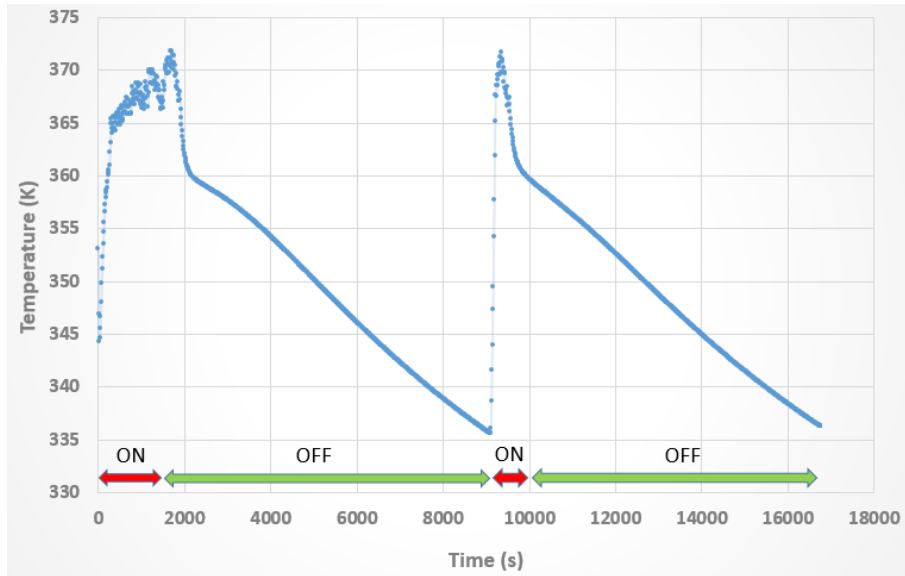


Figure 4.1: Variation of temperature of oil in the sump for simulation number discussed in section

4.2 Sensitivity Analysis

In order to study the effect of approximation made with respect to the transient nature of heat dissipation in the engine bay, a sensitivity analysis is performed. This is done to gauge the extent to which the steady-state approximation can be implemented without significant deviation in the results. Five simulations were run whereby intervals of 5, 15, 30, 45 and 60 seconds of physical time were employed for data exchange between the two software. All simulations were performed with custom 5-hour drive cycle constructed before. The variation of the mass-averaged oil sump temperatures from this study are presented in figure 4.2.

The rate at which heat dissipates from the external surface of the engine solids is a function of the effective heat transfer coefficient and the temperature difference applicable for their interface with the surrounding air.

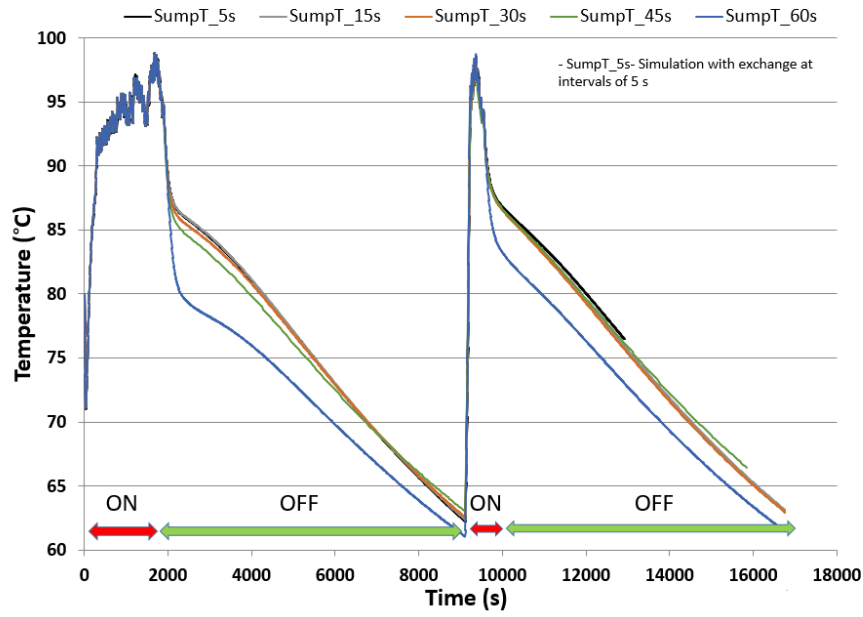


Figure 4.2: Mass-averaged oil sump temperatures from coupled simulations

As their temperature goes down after the engine shut-down, the heat loss rate also reduces. Hence, from a theoretical perspective, it is expected that different exchange intervals between the two software will lead to different temperatures of the oil. For the simulations where data is exchanged after longer intervals, such as 60 seconds, the rate of heat loss to the surroundings that is imposed on the engine solids is maintained at a higher value for longer intervals than in the actual scenario. In other words, the simulations performed with higher rate of data exchange would be "more transient" in nature and closer to what happens in reality. The simulation with exchange interval of 5 seconds can hence be considered as the benchmark among these simulations from the point of view of accuracy.

This is evident in the plotted results 4.2. At the same time, it can also be noticed that the difference in the predicted temperature values diminish after a point. The variation in oil temperature values for the simulations performed with 5, 10 and 15 seconds of exchange interval are found to be less than 1° C for the entire cycle. On the other hand, the simulations with 60 seconds of exchange interval showed upto 7 degrees of difference from the values predicted by the simulation with exchange interval of 5 seconds. This leads to the conclusion that performing coupled simulations with the developed tool with data exchange intervals of 30 seconds or less are sufficiently precise for most applications.

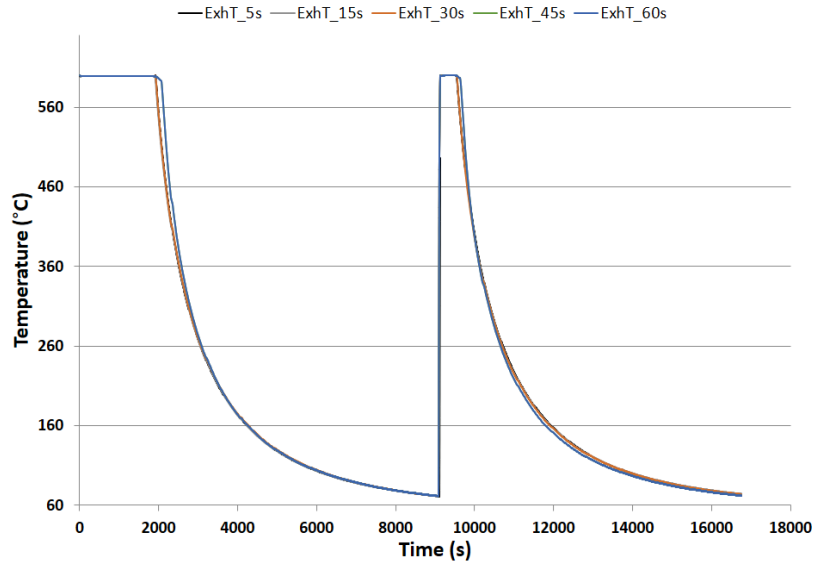


Figure 4.3: Mass-averaged exhaust manifold temperatures from coupled simulations

4.3 Thermal Soak Phenomenon

Thermal soak is a phenomenon that results from prolonged driving of a vehicle at high load, followed by shutting off the engine. It results in increased temperatures of the engine solids and surrounding components, namely the parts in close proximity to the exhaust manifold, after the engine is shut down. This occurs because the exhaust manifold is still very hot when the engine is shut down hence the parts close to it continue to receive energy in form of radiation. Also, once the engine is shut down, the cooling system ceases its duty of cooling down the engine. The rate at which heat is lost from these engine solids through convection is overshadowed by the rate at which they gain the heat energy being radiated by the hot exhaust manifold.

As the 3D CFD model of the developed simulation takes into account the effects of the convection and radiation, the phenomenon of thermal soak can be observed in the simulation results. The variation of mass averaged temperature of the engine cylinder head are presented in figure 4.4. The rise in temperature of the cylinder head during the 10 minutes (approximately) that follow the two occurrences of engine shut down in the drive cycle can be observed in this plot.

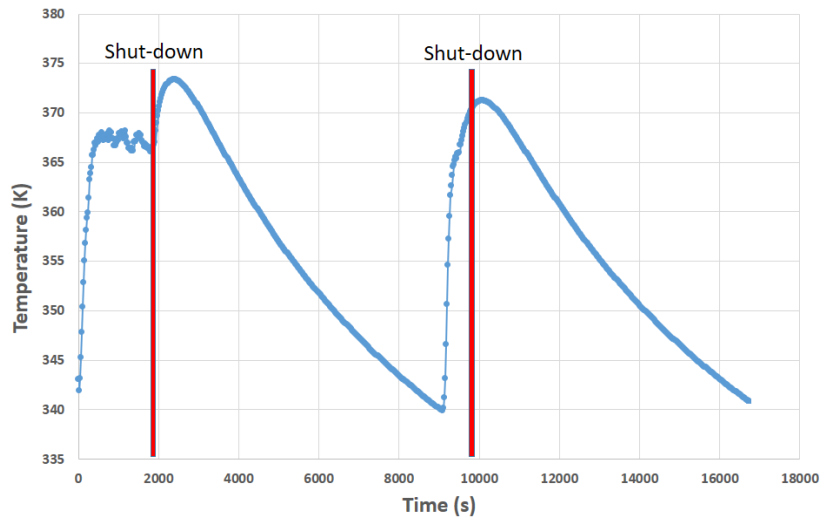


Figure 4.4: Thermal soak phenomenon observed for the cylinder head

Concluding Remarks

This work presents the development of a coupled 1D-3D simulation tool for numerical analysis of the radiation and natural convection driven conjugate heat transfer in the engine bay of a passenger vehicle. Following are the primary specifications and functionality of this tool:

- A fully integrated, 1D model of the powertrain cooling systems of the Volvo S80 sedan in GT-SUITE software is utilized. This model has been validated against experimental data. A mean value model of the engine was employed for computing the rejected combustion heat. The exhaust manifold has been modelled as a thermal mass that can receive heat from the exhaust gases and dissipate it to its external surroundings. A time-dependent profile of the mass averaged temperature of the exhaust manifold and the oil in the sump are captured by this model.
- A 3D CFD model of the natural convection and radiation modes of heat transfer in the engine bay is employed in STAR-CCM+.
- Communication of surface temperatures of the engine solids and their heat transfer rates, between the two software, is achieved through a Java macro which uses comma separated values files to exchange these data.
- The coupled simulation tool is versatile and can be used to perform simulations for any user defined drive cycle.
- Runtimes approximately equal to 4 X Physical-time were achieved for the coupled model was while utilizing 200 computing cores for the 3D CFD simulations. This is valid for an exchange interval of 15 seconds and a tolerance of +/- 2 Watts in the heat transfer rates calculated by the 3D model.
- Through further improvements, the developed tool has potential for quantifying the fuel saving benefits of thermal encapsulations for vehicle engines.

6

Future Work

The present study provides a strong basis for further improvements in the following areas:

- Setting up a more detailed 3D CFD model in terms of spatial resolution of the thermal boundaries. This improvement will result in more accurate predictions of the external flow field around the powertrain components, improving the overall fidelity of the model.
- Performing transient 3D CFD simulations instead of currently employed steady-state approximation, and comparing the results from the coupled simulations for these two setups.
- Collecting experimental data for the surface temperatures of the engine solids for validation purposes and further calibration of the models to improve their reliability and robustness.
- Replacing the existing Boolean fan controller in the 1D model with a more comprehensive one to improve the accuracy of the predicted temperatures for the cooling system.
- Improving the thermostat model to make it possible to control its functioning based on coolant pressure, independent of the coolant temperatures.
- Calibration of the turbocharger oil circuit branch to better predict the heat rejected by them.

Bibliography

- [1] I. Dincer H.S. Hamut and G.F. Naterer. Performance assessment of thermal management systems for electric and hybrid electric vehicles. *int. j. energy res.*, 37: 1–12. doi:10.1002/er.1951. Technical report, International Journal of Energy Research, 2013.
- [2] Ulf Nilsson Emil Ljungskog. Cfd for underhood modeling: Development of an efficient method, master’s thesis 2014:37, issn 1652-8557. Technical report, Chalmers University of Technology, 2014.
- [3] Mercedes ECO thermo engine cover. <http://www.greencarcongress.com/2014/03/20140318-eco.html>.
- [4] Parviz Merati, Charles Davis, K-H Chen, and JP Johnson. Underhood buoyancy driven flow—an experimental study. *Journal of Heat Transfer*, 133(8):082502, 2011.
- [5] Blago B Minovski, Lennart Lofdahl, and Peter Gullberg. Numerical investigation of natural convection in a simplified engine bay,” sae technical paper 2016-01-1683, 2016, doi:10.4271/2016-01-1683. Technical report, SAE Technical Paper, 2016.
- [6] Yunus A Cengel, Sanford Klein, and William Beckman. *Heat Transfer: A practical approach*, volume 2nd Edition. McGraw-Hill New York, 1998.
- [7] Mahajan Sammakia Gebhart, Jaluria. *Buoyancy Induced Flows and Transport*, volume Textbook Edition. Taylor and Francis, 2003.
- [8] Water Pump. <http://asc-ind.com/wp-content/uploads/2010/10/PUMP-w-raw-mat-callouts.jpg>.
- [9] Thermostat. <http://www.ustudy.in/sites/default/files/images/thermostat.jpg>.
- [10] Dinakar P. Heilbronn D. Jurik J. Kumar S. Rajeeve G. Sarovec M. Coupled unsteady simulations of powertrain cooling systems. Technical report, Chalmers, 2015.
- [11] Intercooler. <http://www.b15u.com/attachment.php?attachmentid=57&stc=1&d=1248287952>.
- [12] Radiator. <http://radiatorx.blogspot.se/2013/01/car-radiator-fluid.html>.
- [13] Minovski, Blago B and Lofdahl, Lennart: Gothenburg, Chalmers University of Technology, 2013. Diploma work: Study of software system integration for transient simulation of future cooling system for heavy truck applications , ISSN 1652-8557; 2013:40, 2013. SAE Technical Paper, 2014 .
- [14] Engine Cooling Fan. http://repairpal.com/images/managed/content_images/encyclopedia/CM_Engine/Engine_Cooling_Fan_09.11.png.
- [15] Intercooler. <http://www.zoombd24.com/wp-content/uploads/2015/02/Expansion-tank.png>.
- [16] Gamma Technologies official website. <http://www.gtisoft.com>.
- [17] Mattias Wahde. *Biologically inspired optimization methods: An introduction*. WIT press, 2008.
- [18] Morel, T and others, Manual for GT Power version 7.5, Gamma Technologies, 2014.
- [19] Ronald L Iman. Latin hypercube sampling. *Encyclopedia of quantitative risk analysis and assessment*, 2008.

A

Appendix

A.1 Java Macro Code

```
package WriteTest;

import java.util.*;
import java.io.*;

import star.base.neo.*;
import star.base.report.Report;
import star.common.*;
import star.vis.*;
import star.flow.*;
import star.energy.*;

public class CyberdyneS103 extends StarMacro {int k=0;
    public void execute() {
        execute0();
    }
    private void execute0() {
        Simulation simulation_0=getActiveSimulation();
        File f = new File(resolvePath("Output.csv"));
        File fi = new File(resolvePath("Input.csv"));

        double doubleValue1=0.0; // Declaring and initializing engine state variable
        double doubleValue2=0.0;
        double doubleValue3=0.0;
        double doubleValue4=0.0;
        double doubleValue5=0.0;
        double doubleValue6=0.0;
        double doubleValue7=0.0;
        double doubleValue8=0.0;

        for (int i = 1; i <= 1600; i++){ // This loop specifies the maximum allowed number of
            exchanges between GT-SUITE and StarCCM
            if (i>1) { //The Input file written by GT-SUITE is read before the second
                exchange in order to allow StarCCM to make at least one iteration before being
                dependent on GT-SUITE

                if (!fi.exists()) { // Checking if Input.csv exists
                    simulation_0.println("Waiting for GT-SUITE to write Input.csv at exchange number " +
                        (i-1));
                }
                while (!fi.exists()) { // Wait until GT-SUITE writes Input.csv

                    try {
                        Thread.sleep(1000);
                    } catch (InterruptedException e) {
                        //Handle exception
                    }
                }
            }
        }
    }
}
```

```

try {
    //The try-catch statement ensures that the code doesnt crash in
    case something goes wrong

    FileReader fr = new FileReader(fi);
    BufferedReader br = new BufferedReader(fr);

    Scanner sc = new Scanner(br).useLocale(Locale.ENGLISH);
    sc.useDelimiter(","); //Defining delimiter for .csv format
    int j=1; //Counter for row number
    while (sc.hasNextLine() ) {
        if (j==2) {

            doubleValue2 = sc.nextDouble(); // Reading the first double on the second
            row of the .csv file
            simulation_0.println("Engine head temperature read in Kelvin " +
            doubleValue2);
            // Updating boundary conditions in STARCCM simulation
            Region region_0 = simulation_0.getRegionManager().getRegion("Region");
            Boundary boundary_0 =
                region_0.getBoundaryManager().getBoundary("Engine_head");
            StaticTemperatureProfile staticTemperatureProfile_0 =
                boundary_0.getValues().get(StaticTemperatureProfile.class);
            Units units_0 = ((Units) simulation_0.getUnitsManager().getObject("K"));
            staticTemperatureProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
            staticTemperatureProfile_0.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
            //
            String s;
            s = sc.next();
            doubleValue3 = Double.parseDouble(s);
            simulation_0.println("Engine block temperature read in Kelvin " +
            doubleValue3);
            //Updating boundary coinditions in STARCCM simulation
            Boundary boundary_1 =
                region_0.getBoundaryManager().getBoundary("Engine_block");
            StaticTemperatureProfile staticTemperatureProfile_1 =
                boundary_1.getValues().get(StaticTemperatureProfile.class);
            Units units_1 = ((Units) simulation_0.getUnitsManager().getObject("K"));
            staticTemperatureProfile_1.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
            staticTemperatureProfile_1.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
            //
            String b;
            b = sc.next();
            doubleValue4 = Double.parseDouble(b);
            simulation_0.println("Oil sump temperature read in Kelvin " +
            doubleValue4);
            //Updating boundary coinditions in STARCCM simulation
            Boundary boundary_2 =
                region_0.getBoundaryManager().getBoundary("Oil_sump");
            StaticTemperatureProfile staticTemperatureProfile_2 =
                boundary_2.getValues().get(StaticTemperatureProfile.class);
            Units units_2 = ((Units) simulation_0.getUnitsManager().getObject("K"));
            staticTemperatureProfile_2.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
            staticTemperatureProfile_2.getMethod(ConstantScalarProfileMethod.class).getQuantity().set

            String d;
            d = sc.next();
            doubleValue1 = Double.parseDouble(d);
            simulation_0.println("Current Engine State is " + doubleValue1);

            String bb;

```

```

        bb = sc.next();
        doubleValue5 = Double.parseDouble(bb);
        simulation_0.println("Manifold temperature read in Kelvin " +
            doubleValue5);
        //Updating boundary coinditions in STARCCM simulation
        Boundary boundary_3 =
            region_0.getBoundaryManager().getBoundary("Exhaust_manifold");
        StaticTemperatureProfile staticTemperatureProfile_3 =
            boundary_3.getValues().get(StaticTemperatureProfile.class);
        Units units_3 = ((Units) simulation_0.getUnitsManager().getObject("K"));
        staticTemperatureProfile_3.getMethod(ConstantScalarProfileMethod.class).getQuantity().set
        staticTemperatureProfile_3.getMethod(ConstantScalarProfileMethod.class).getQuantity().set

        String ot;
        ot = sc.next();
        doubleValue6 = Double.parseDouble(ot);
        simulation_0.println("Oil temperature read in Kelvin " + doubleValue6);

        String ch;
        ch = sc.next();
        doubleValue7 = Double.parseDouble(ch);
        simulation_0.println("Cylinder head temperature read in Kelvin " +
            doubleValue7);

        String tm;
        tm = sc.next();
        doubleValue8 = Double.parseDouble(tm);
        simulation_0.println("Time read in s " + doubleValue8);

        if (doubleValue1==1){
            simulation_0.println("The engine is currently running, feeding standard
                values for heat transfer...");
            k=1;

        }

        else { simulation_0.println("The engine is currently shut down, STAR-CCM+
            will compute the HTR values...");

            k=2;}
    }

    // Deleting Input.csv after reading it
    else {

        sc.nextLine();
    }
    j++;
}
    br.close();
    fr.close();
}

catch (Exception ex) {
    simulation_0.println("ERROR reading file."); // In case anything goes wrong
}
// Deleting Input.csv after reading it
try{
if(fi.delete()){

```

```

        simulation_0.println(fi.getName() + " is deleted!");
    }else{
        simulation_0.println("Delete operation is failed.");
    }
}

}catch(Exception e){

    e.printStackTrace();

}

    }
    // Checking if Output.csv exists
    if (f.exists()) {
        simulation_0.println("Waiting for GT-SUITE to read and delete Output.csv at exchange
            number " + (i-1));
    }
    while (f.exists()) { // Wait until GT-SUITE deletes Output

        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            //Handle exception
        }

    }

}

if(k==2){
    simulation_0.println("Preparing output for GT-SUITE for exchange number " +(i-1));
    double residual_os=0.0;
    double residual_eb=0.0;
    double residual_em=0.0;

    simulation_0.getSimulationIterator().step(2,true);
    Collection<Report> colrep = simulation_0.getReportManager().getObjects();
    double[] repValchecker = new double[colrep.size()];

    simulation_0.println("\n\nSimulation will now iterate till the HTR values
        stabilize...");
    for (int z = 1; z <= 60; z++){

        simulation_0.println("Starting iteration number " + z + " for exchange process number
            " +(i-1)+'\n');
        for (Report rep : colrep){
            if (rep.getPresentationName().contains("3")) {
                repValchecker[0] = rep.getReportMonitorValue();}}

        for (Report rep : colrep){
            if (rep.getPresentationName().contains("2")) {
                repValchecker[2] = rep.getReportMonitorValue();}}

        for (Report rep : colrep){
            if (rep.getPresentationName().contains("7")) {
                repValchecker[4] = rep.getReportMonitorValue();}}

        simulation_0.getSimulationIterator().step(1,true);
    }
}

```

```

        for (Report rep : colrep){
            if (rep.getPresentationName().contains("3")) {
                repValchecker[1] = rep.getReportMonitorValue();}}

        for (Report rep : colrep){
            if (rep.getPresentationName().contains("2")) {
                repValchecker[3] = rep.getReportMonitorValue();}}

        for (Report rep : colrep){
            if (rep.getPresentationName().contains("7")) {
                repValchecker[5] = rep.getReportMonitorValue();}}

        simulation_0.println("\nIteration values saved...");

        residual_os=(repValchecker[1]-repValchecker[0]);
        residual_eb=(repValchecker[3]-repValchecker[2]);
        residual_em=(repValchecker[5]-repValchecker[4]);

        simulation_0.println("Current value of Oil Sump HTR residual is " +residual_os);

        simulation_0.println("\nCurrent value of Engine Block HTR residual is " +residual_eb);

        simulation_0.println("\nCurrent value of Exhaust manifold HTR residual is "
            +residual_em);

        if (residual_os<2 && residual_os>-2 && residual_eb<2 && residual_eb>-2 &&
            residual_em<2 && residual_em>-2)

        {simulation_0.println("Residuals annulled! Breaking iteration loop now...");break;}

    }

}

if(i>1){writeToFile(simulation_0); writeToFile1(simulation_0, doubleValue5, doubleValue6,
    doubleValue7, doubleValue8);}

}

}

private void writeToFile(Simulation simulation_0) {
    Collection<Report> colrep = simulation_0.getReportManager().getObjects();
    double[] repVal = new double[colrep.size()];

    String line = "";
    File buffer = new File(resolvePath("Buffer.csv"));
    File forig = new File(resolvePath("Output.csv"));

```



```

try {

    BufferedWriter bw = new BufferedWriter (new FileWriter(buffer,true));

    String headerline;
    headerline =
        "\"HeatTransfer1\\\", \"HeatTransfer2\\\", \"HeatTransfer3\\\", \"HeatTransfer7\\\", \"HeatTransfer5\\\", \"HeatTr

    bw.write(headerline);
    bw.newLine();

    if (k==1){

    for (Report rep : colrep) {
        if (rep.getPresentationName().contains("1")) {
            repVal[0] = -1200;
            repVal[4] = -800;
        }

        if (rep.getPresentationName().contains("2")) {
            repVal[1] = -1500;
        }

        if (rep.getPresentationName().contains("3")) {
            repVal[2] = -600;
        }

        if (rep.getPresentationName().contains("7")) {
            repVal[3] = -5000;
        }
    }

    }

    else { for (Report rep : colrep) {
        if (rep.getPresentationName().contains("1")) {
            repVal[0] = Math.round(rep.getReportMonitorValue()*200/3)/100;
            repVal[4] = Math.round(rep.getReportMonitorValue()*100/3)/100;
        }

        if (rep.getPresentationName().contains("2")) {
            repVal[1] = Math.round(rep.getReportMonitorValue()*100/2)/100;
        }

        if (rep.getPresentationName().contains("3")) {
            repVal[2] = Math.round(rep.getReportMonitorValue()*100)/100;
        }
        if (rep.getPresentationName().contains("7")) {
            repVal[3] = Math.round(rep.getReportMonitorValue()*100)/100;
        }
    }

    }

    simulation_0.println("Outputs to GT-SUITE:" + repVal[0] + ", " + repVal[1] + ", " +
        repVal[2] + ", " + repVal[3] + ", " + repVal[4] + '\n' + '\n' + '\n' + '\n');
}

```

```

        line = String.valueOf(repVal[0]) + "," + String.valueOf(repVal[1]) + "," +
            String.valueOf(repVal[2]) + "," + String.valueOf(repVal[3]) + "," +
            String.valueOf(repVal[4]);

        bw.write(line);
        bw.newLine();
        bw.close();
    } catch (IOException iOException) {
    }
    buffer.renameTo(forig);
}

//
private void writeToFile1(Simulation simulation_0, Double doubleValue5, Double doubleValue6,
    Double doubleValue7, Double doubleValue8) {

    String line = "";

    try {
        File f = new File(resolvePath("OutputTemp.csv"));
        BufferedWriter bw = new BufferedWriter (new FileWriter(f,true));

        String headerline;
        headerline = "\"Time\",\"Oil temp\",\"Exhaust temp\",\"Head temp\"";
        bw.write(headerline);
        bw.newLine();

        simulation_0.println("Logged Data:" + doubleValue8 + ", " + doubleValue6 + ", " +
            doubleValue5 + ", " + doubleValue7 + '\n' + '\n' + '\n' + '\n');
        line = "" + doubleValue8 + "," + doubleValue6 + "," + doubleValue5 + "," + doubleValue7;

        bw.write(line);
        bw.newLine();
        bw.close();
    } catch (IOException iOException) {
    }
}

}

```

A.2 Distribution of Content

The contribution of the students writing this report to the content presented in it is shown in table A.1 below for future references:

| Parameter | Author |
|--|--------------------------------|
| Front Matter (Cover, Abstract, Preface & Introduction) | Saket Kumar |
| Nomenclature | Daniel Pascual and Saket Kumar |
| Theoretical and Modelling Background (Pages: 2, 5-10) | Daniel Pascual |
| Theoretical and Modelling Background (Pages: 3-4, 11) | Saket Kumar |
| Methods (Pages 18-26) | Daniel Pascual |
| Methods (Pages 12-17, 27-31) | Saket Kumar |
| Results and Discussion (Pages 32-34) | Saket Kumar |
| Concluding Remarks (Page 35) | Saket Kumar |
| Future Work (Page 36) | Saket Kumar |
| Appendix (Page I-VIII) | Saket Kumar |
| Bibliography | Daniel Pascual and Saket Kumar |

Table A.1: Distribution of content in the report