Sasol Advanced Fuels Laboratory

UNIVERSITY OF CAPE TOWN

The Development of an Enhanced Autoignition Sub-model for use in CFD Combustion Simulations

Author:

Ryan Cox

Supervised by:

Prof. Andy Yates and Dr Chris Meyer

A dissertation submitted to the Department of Mechanical Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Science in Engineering

Cape Town, South Africa
20 February 2006

Declaration

1. *Plagiarism is the illegal use of another's work and pretending that it's one's own.*
2. *I have used the convention for citation and referencing. Each significant contribution to, and quotation in, this project from the work(s) of other people has been attributed, and has been cited and referenced.*
3. *This project is my own work.*
4. *I have not allowed, and will not allow anyone to copy my work with the intention of passing it off as his or her own work.*

Signature:

Date: 13/02/06

Acknowledgements

To my supervisors Prof. A. Yates and Dr. C. Meyer for providing assistance when ever it was required

To Andre Swarts for not closing his office door and providing invaluable assistance.

To Carl Viljoen for providing help with the Chemkin® models and general chemical questions.

To Sasol for providing the financial backing and giving the opportunities to make this degree more beneficial.

Again to Prof. A. Yates for providing assistance in the writing of this report.

To Lindsay Kalis for proof reading this thesis

To all those that have helped along the way

Abstract

With the ever increasing pressure to manufacture more efficient engines that produce lower exhaust emissions, there is a corresponding need for a greater understanding of the combustion processes within these engines. Specifically, it is the interaction between the fuel and the engine that represents one of the greatest research challenges. As the systems become increasingly sophisticated, the fuel companies are experiencing an increased demand for high specification fuels with tighter tolerances. Now, more than ever, the fuel design and engine design need to work as one integrated system to meet these expectations.

In this project, an attempt was made to produce a computationally efficient mathematical model of the fuel ignition characteristics that could be used in a CFD simulation of an internal combustion engine or any other generic combustion system. The report follows the following format:

- Literature review of the available numerical fuel models.
- Development of the fuel model and implementation in CFD code.
- Validation of the model using experimental data of methane fuel in a continuous flow reactor.
- Develop a simulation of n-Heptane evaporating in Cetane test conditions and investigating the correlation with experimental data.

The validation of the model provided useful insight into the need for good quality experimental data for the fuels of interest, highlighting the need for pure ignition delay curves without engine effects, which is a limitation of many of the current models.

When modelling a single fuel droplet, the importance of the temperature profile in the vicinity of the evaporating particle was clearly illustrated, as well as the variation in the local air/fuel ratio. These effects were shown to play off against each other - the centre of the droplet being coldest and hence yielding a longer ignition delay while, at the same time, the high equivalence ratio near the centre had the effect of shortening the ignition delay. Using a simple, two dimensional model and using n-Heptane as the fuel, a realistic prediction of the overall ignition delay was obtained. More importantly, the critical zone of the initial auto-ignition was identified. These simulations show how this model can be used in an environment that exhibits both gradients of temperature and equivalence ratio. It also shows

the importance of including such in-homogeneities when creating engine models that include fuel injection. This approach can easily be extended into any type of combustion simulation involving fuel droplets where local temperature and equivalence ratios have a controlling effect on the ignition.

Some recommendations for future work include the modelling of the IQT™ with the possibility of reconciling the ignition delay of the IQT™ and the cetane test.

Contents

List of Figures

List of Tables

Nomenclature

E_a	activation energy
R	universal gas constant
A	pre-exponential constant of proportionality
B	constant multiplier for temperature
n	pressure exponent
k	rate coefficient
τ	ignition delay
t_c	time of autoignition
t_0	start time (fuel injection)
ϕ	fuel/air ratio
e	constant exponential for phi

Acronyms

CFD	Computational Fluid Dynamics
PRF	Primary Reference Fuels
IQT	Ignition Quality Tester
UDF	User Defined functions
UDS	User Defined Scalar
HCCI	Homogeneous Charge Compression Ignition
CFR	Coordinating Fuels Research
GUI	Graphical User Interface
RPM	Revolutions Per Minute
OCTN	Octane Number

1 Introduction

1.1 Fuel Research

With the ever increasing pressure to manufacture more efficient engines that produce lower exhaust emissions, there is a corresponding need for a greater understanding of the combustion processes within these engines. Specifically, it is the interaction between the fuel and the engine that represents one of the greatest research challenges. As the systems become increasingly sophisticated, the fuel companies are experiencing an increased demand for high specification fuels with tighter tolerances. Now, more than ever, the fuel design and engine design need to work as one integrated system to meet these expectations.

In this project, an attempt was made to produce a computationally efficient mathematical model of the fuel ignition characteristics that could be used in a CFD simulation of an internal combustion engine or any other generic combustion system. The scope of the project was limited to the determination of the points of autoignition, both in time and geometric position. In this way, it was anticipated that potential fuels could be tested in a virtual environment to provide a means of predicting the overall combustion performance. The proposed analysis technique would also offer the potential to compare the sensitivity of the computational method with other, less sophisticated fuel models and to quantify the associated differences.

1.2 Background

This is a brief description of the two different internal combustion engine ignition mechanisms to highlight the differences and to introduce the reader to some of the terms.

1.2.1 Compression Ignition

In a compression ignition engine air is drawn into the combustion chamber during the intake stroke. The air is then compressed. This compression elevates the temperature and pressure of the air in the combustion chamber. Diesel is then injected into the chamber. Due to the conditions in the chamber the fuel auto ignites (in zones where the local air fuel ratio is within

the combustible range). This requires the fuel to have a propensity to spontaneously ignite quickly and easily, and the engine has a high compression ratio to ensure adequate temperature and pressure in the combustion chamber to promote the ignition of the fuel.

1.2.2 Spark Ignition

In spark ignition engines, the petrol and air are premixed and drawn into the engine during the intake stroke. This leads to a homogeneous charge in the engine. This charge needs to be approximately stoichiometric (sufficient oxygen for the fuel to react with so that it is all converted to CO_2 and H_2O). This means that fuel and air needs to be regulated in a relatively fixed ratio for engine load control. This is one of the causes of inefficiency in the spark ignition engines (up to 15% loss, referred to as pumping losses, in comparison to diesel engines).

Ignition timing is controlled by using the spark plug. The spark causes the fuel/air mixture in the vicinity of the plug to ignite and this leads to a flame front developing and propagating outwards.

This places some particular requirements on the fuel and the engine.
Fuel:

The fuel must only ignite due to the spark and developing flame front and should not auto ignite (details on this mechanism will be discussed further on in the report)
Engine:

This implies that the compression ratio in the engine has an upper limit. This is to limit the temperature and pressure that the charge experiences. The higher the temperature and pressure in the engine the increased risk of autoignition of the fuel occurring.

The size of the combustion chamber, and more importantly the distance the flame front needs to propagate to consume all the charge, has a direct impact on the time that the end gas is exposed to the elevated temperature and pressure. This implies that there is also an upper limit on the practical size of the cylinder bore of a spark ignition engine (typically less than ninety millimetres)

2 Literature Review

2.1 Autoignition Scales

As seen previously the compression ignition and spark ignition engines have very different requirements in terms of the fuel's autoignition requirements, with petrol requiring autoignition resistance and diesel having a propensity to auto ignite. There are two measurement scales – one for petrol and the other for diesel – but both describe the same phenomenon in different ways. The octane scale is used for petrol and the cetane scale is used for diesel.

2.1.1 The octane scale

The octane scale is defined as the ratio of iso-octane and n-heptane that will yield an equivalent autoignition resistance to the test fuel in a defined engine environment. These two fuels were selected as primary reference fuels (PRF) as they have similar physical properties. Their resistance to autoignition is, however, very different. Iso-octane was assigned the octane value of one hundred and n-heptane was assigned the value of zero. All other fuels are then compared to a mixture of these two primary reference fuels. Therefore 97 octane petrol has the same autoignition characteristics in the octane test engine as a mixture of 97% iso-octane and 3% n-heptane.

2.1.2 The cetane scale

The cetane scale was created in a similar way as the octane scale but different reference fuels were used – normal Cetane (n-hexadecane) and isocetane (heptamethylnonane, HMN). Cetane is assigned to have the value 100 cetane and isocetane has the value 15. The scale is defined by a ratio of these fuels that yield an ignition delay that is the same as the test fuel under the same engine operating condition. In this case the cetane number is the percentage of cetane plus 0.15 times the percentage of the isocetane. The minimum cetane value recommended in South Africa is 45 (SANS 342:2003).

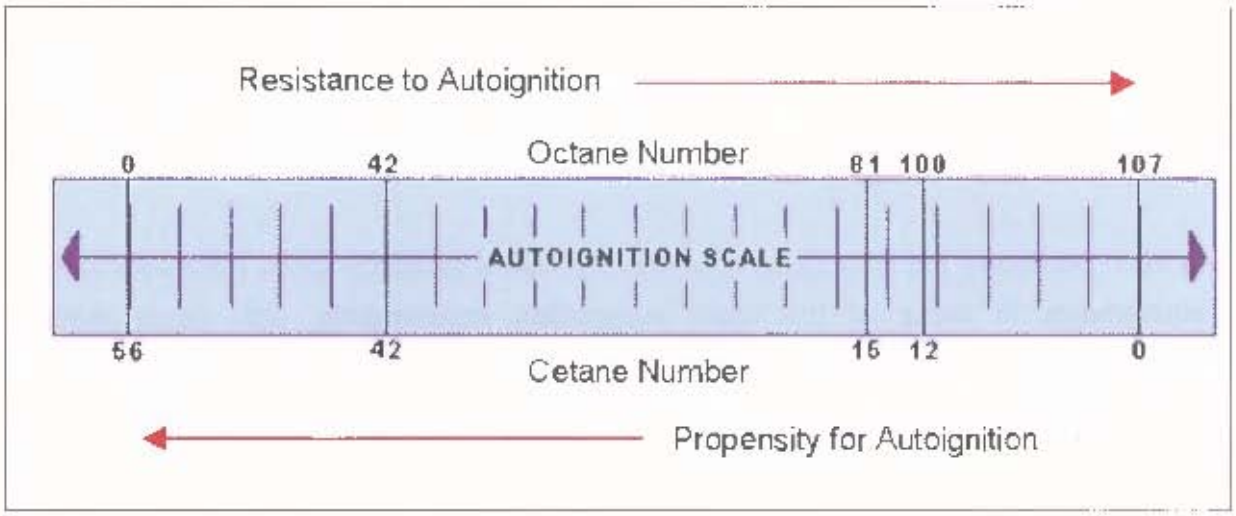

Figure 2-1: Schematic of the autoignition scale

The figure above shows the autoignition scale with both the octane numbers and cetane numbers on it for comparison. In must be noted that due to the complexities whereby various fuels respond differently to changes in the engine environment the scales are not universally comparable. The figure above shows selected points that are approximately comparable for particular fuels.

It is important to note that with both scales, a test fuel is compared to reference fuels at defined operating test conditions and with a specified test engine.

2.1.3 Characterising fuels

As discussed above the octane scale and the cetane scale are both defined relative to reference fuels, but this is not an absolute characterisation. This is made more apparent with the octane rating, which actually has two scales based on two tests. Both tests are conducted using the same engine and in a similar manner. The differences in the tests lie with the air inlet temperature and engine speed. Carrying out the two tests usually results in different octane ratings for test fuels. The relationship between the two octane ratings is not reconcilable in a standard way for all fuels. For production requirements this system is sufficient but in the interest of research and fuel characterisation (especially with the advent of synthetic fuels) this method is unsatisfactory.

2.1.4 Arrhenius Function

This is a chemically based equation to express the reaction rates as a function of activation energy and temperature:

$$k = A \times \exp\left(\frac{-E_a}{R \times T}\right)$$ Equation 2-1

Where: E_a is activation energy
R is the universal gas constant
A is the pre-exponential constant of proportionality
k is the rate coefficient

This formula is considered a fundamental chemical equation and is believed to be a good representation of the fundamental chemistry. It indicates the rate at which global reactions occur based on the temperature of the environment that the reaction is taking place.

To transform this equation from a rate based equation to a global ignition delay the correlation below is used:

$$\tau_G = \frac{Const}{k_G}$$ Equation 2-2

Where: τ_G is the mean lifetime of all the reactant molecules
Const is the constant (see below)
k_G is the rate constant

In first order reactions, the constant is taken as one, this is an assumption but in most cases holds true. This means that the average life of the fuel molecules before autoignition is the inverse of the rate constant. This forms the basis of the general equation proposed by Livengood and Wu (1955). The equation proposed is a model of the global reactions.

$$\tau = Ap^{-n} \exp\left(\frac{B}{T}\right)$$ Equation 2-3

Where: τ is ignition delay

A is the pre-exponential constant of proportionality
B is the constant multiplier for temperature
n is the exponential for the pressure term
T is the temperature (in Kelvin)
p is the pressure (in bar)

As can be seen, provision for an effect of pressure has been included, due to pressure dependence on the reaction rate of fuel species being indicated in early experimentation.

Equation 2-3 assumes that the temperature and the pressure are constant throughout the experiment. This would apply to a rapid compression machines (RCM). In a normal engine, temperature and pressure are transient which introduces a complication in the analysis.

Livengood and Wu (1955) proposed technique for dealing with the difficulty. The ignition delay integral assumes chemical reactions leading to the autoignition are cumulative in effect. This allows delay data obtained from constant pressure and temperature experiments to be used in a changing environment, such as an engine. This leads to the equation below.

$$1 = \int_{t_0}^{t_c} \frac{dt}{\tau} \qquad \text{Equation 2-4}$$

Where t_c is the time to autoignition.
t_0 is the time the start time (fuel injection)
τ is ignition delay

Which expands to:

$$1 = \int_{t_0}^{t_c} \frac{dt}{Ap^{-n}(t)\exp\left[\frac{B}{T(t)}\right]} \qquad \text{Equation 2-5}$$

The Arrhenius function proposed by Douaud and Eyzat (1978) which was based on experimentation done on the CFR engine is shown below:

$$\tau = A\left(\frac{OCTN}{100}\right)^{3.40017} p^{-n} \exp\left(\frac{B}{T}\right) \qquad \text{Equation 2-5}$$

Where: OCTN is the octane number of the fuel

For primary reference fuels (PRF) which are blends of n-heptane and iso-octane Douaud (1978) proposes the values:

A: 0.01869

n: 1.7

B: 3800

The equation is, however, limited to fuels with an octane rating of between eighty and a hundred.

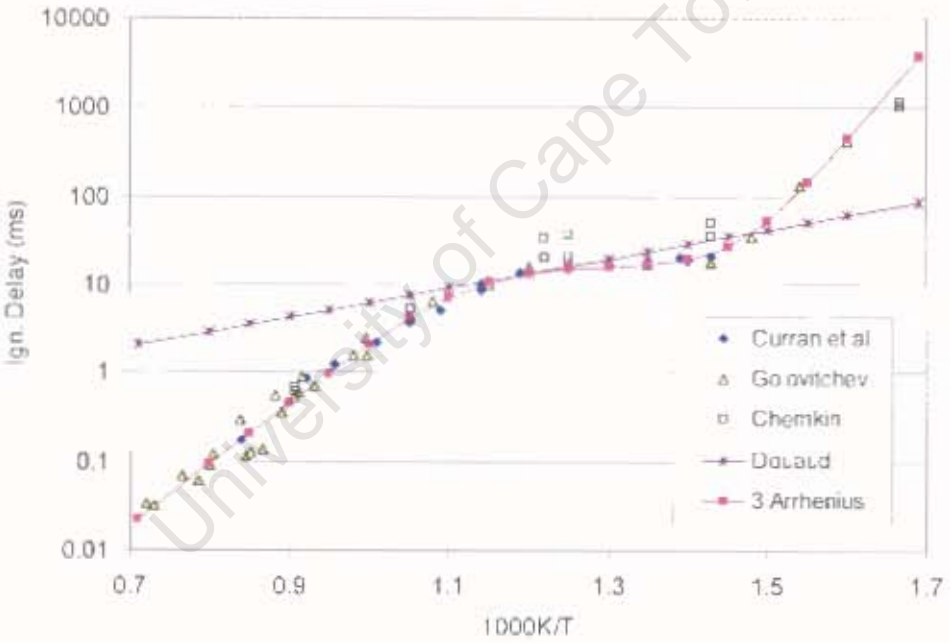

Figure 2-2: Iso-Octane at 18 bar

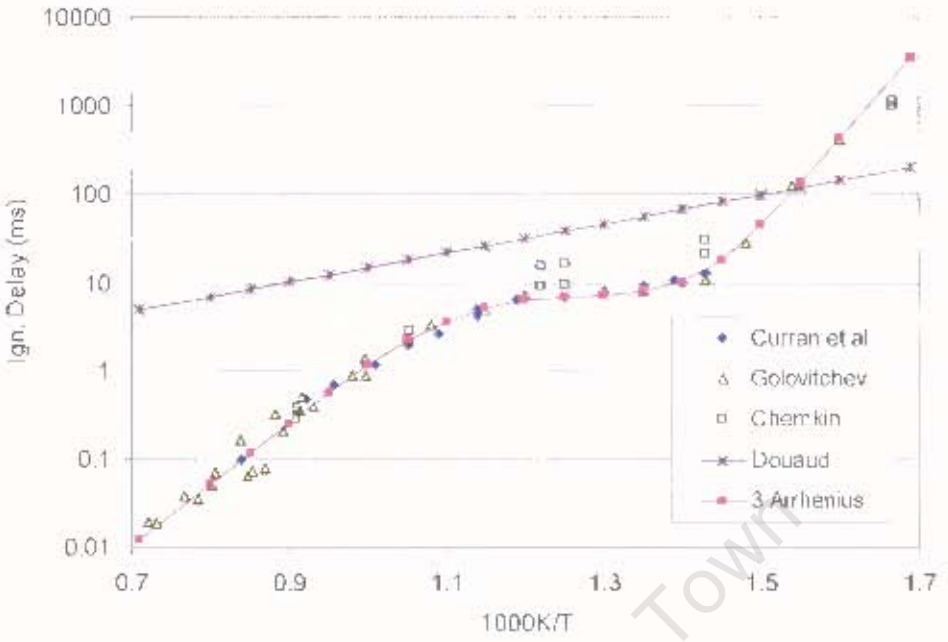

Figure 2-3: Iso-Octane at 30 bar

In the figures above the function proposed by Douaud and Eyzat (1978) shows a limited correlation with the experimental data. In their paper the calculated equivalent temperature and pressure of the test is 21 bar and 1010 K (Douaud and Eyzat, 1978) as can be seen for these conditions the correlation is good (figure 2-2) over a limited range of temperature. As the pressure changes it can be seen that the theoretical model predictions deviate from the experimental data (figure 2-3). It can also be seen that at areas of high or low temperature the model shows very little correlation with the experimental data.

The experimental data shown in figs 2-2 and 2-3 reveals that there are two underlying reaction paths, a high temperature path and a lower temperature path. At high temperature, the one path is the preferred route and at lower temperatures the other is the preferred route, the area where the preferred routes change give rise to a negative temperature gradient region, an area where as the temperature gets higher the ignition delay is essentially constant. In the work done by Yates et al (2005) this phenomenon has been conveniently described in a single ignition delay model. This was done by making use of 3 Arrhenius functions, these functions were added in the following way.

$$\tau_{total} = \frac{1}{\left(\frac{1}{(\tau_1+\tau_2)}+\left(\frac{1}{\tau_3}\right)\right)}$$ Equation 2-6

It has been found that this gives a far better correlation with the experimental data. Values for the Arrhenius functions are given in the table below. These values are calculated to correlate to experimental data for iso-octane. This version will be referred to as the modified Arrhenius function for the remainder of the report.

Table 2-1: Values for modified Arrhenius functions

	A	n	B
τ_1	8.04212E-15	0.018	24032
τ_2	3301.4806	1.602	-522.57
τ_3	6.12E-06	1.12	16086

In figure 2-2 and 2-3 the experimental data with the best fit modified Arrhenius function is shown. The constants for the function shown are given in table 2-1. As can be seen in the figures the experimental data is more closely approximated by the Yates model compared to the Douaud (1978) model. While the Yates model does provide a better approximation of the experimental data, it is only as good as the experimental data. In this model there are now nine unknown constants as opposed to Douaud (1978) having three. Therefore the Yates model needs at least nine experimental data points, further these data points need to be in the three different regions, the low temperature region, the high temperature region and the negative temperature coefficient region. This generally means many more data points are required to be sure that the three regions are properly included. This has limited the model to the following fuels (for which enough data is available) n-heptane, i-octane, 1-hexene, toluene, ethanol, butane and a model diesel type fuel blend (Yates et al, 2005). A simple blending rule for these fuels has been evaluated and it has been found that for binary blending a simple linear by volume rule gives a reasonable approximation for the ignition delay.

Also notice that by making the constants for two of the Arrhenius functions of the Yates (205) model zero the model reduces to the same as that of Douaud (1978).

The work discussed above applies for stoichiometric air/fuel ratios. For naturally aspirated petrol engines this is adequate, this is due to these engines running at stoichiometric air fuel ratios and the charge being homogeneous. With the development of direct injection petrol engines and increased popularity of diesel engines the need for including variable air/fuel ratios is an important aspect, due to diesels running in lean operations and the in-homogeneity of the charge. A method for including this air/fuel ratio as described in Yates et al (2005) is as follows:

$$\tau_{\phi} = \tau_{\phi=1}\phi^{e} \qquad \text{Equation 2-7}$$

Where: $\tau_{\Phi=1}$ is ignition delay at a specific fuel/air ratio

τ_{Φ} is ignition delay at stochiometric

ϕ is the fuel/air ratio

e is the constant exponential for ϕ

This has been determined by a chemical kinetic modelling. The value for e is dependant on the fuel being investigated; the two fuels investigated in the report are methane and n-heptane the values for which are -0.2335 and -0.7887 respectively. Both Douaud and Eyzat (1978) and Fish et al (1969) correspond to this in the lean region but show an increasing ignition delay when the fuel goes to rich. It is proposed (Yates et al, 2005) that this discrepancy is caused by the miscalculation of the temperature of the gas due to cooling effects. There is limited information on the effect of air-fuel ratio and this is a possible area warranting further research.

All the effects and components discussed above are important when modelling autoignition in an engine, and so for the CFD modelling it is of utmost importance to include this correctly to get an accurate representation of the autoignition in an engine.

2.2 Experimental Data in the Literature

There is much experimental work in the area of attempting to characterise the autoignition properties of fuels. There are many techniques for the experimental procedure. Five common methods will be discussed below:

- continuous flow reactors

- shock tubes
- motored engines
- rapid compression engines
- combustion bombs

Each of these methods has unique advantages and disadvantages which will be discussed in the following text.

2.2.1 Continuous Flow Reactor

The simplest manifestation of a continuous flow reactor is a Bunsen burner. The continuous flow reactor is effectively a tube where air is allowed to flow through at a controlled velocity and is compressed to the required pressure and heated to a desired temperature. Fuel is injected in to the flow, as finely atomized as possible. The injection is important as it must be as homogeneous as possible while affecting the flow as little as possible. Sensors further down the tube indicate the position of combustion, thus the ignition delay can be calculated. This is one example of a flow reactor. There are different variations of this, but the basic principles remain the same. This is generally limited to relatively low temperatures and pressures (see table 2-2).

2.2.2 Shock Tubes

A shock tube is a device providing well controlled conditions (pressure, temperature and air fuel ratio) to investigate ignition delay of fuels. A shock tube in its simplest form is two sections of pipe, separated by a diaphragm. The "driver side" has a high pressure gas in it and the "driven side" has the combustible mixture in it. The diaphragm is then ruptured, either by mechanical piercing or by physical failure due to the pressure difference. At rupture a shock wave forms and thus the mixture undergoes a step change in temperature and pressure. Thus the conditions of the mixture are well known during the ignition delay. The advantage of this system is that the mixture is homogeneous and there are no mass flow effects. Shock tubes also allow for high temperature and pressure testing.

2.2.3 Motored Engines

Engines are sometimes used for fuel research, but are generally limited to the study of global effects. The most common example would be the CFR engine, the standard for testing octane in petrol. The biggest drawback for the motored engine is the transient nature of the cycle and the uncertainty of measuring the instantaneous, transient temperature. The advantage of motored engines is that they can give more global results which can closely resemble actual engine operations. They offer poor results for precise fuel characterization.

2.2.4 Rapid Compression Engines

A Rapid Compression Machine (RCM) is an experimental device that represents a single cycle of an internal combustion engine. The advantage of the RCM over a normal engine is the compression time is reduced as much as possible thus minimising variation in pressure and temperature making it possible to measure the temperature indirectly and enabling the determination of the Arrhenius function without needing the integral technique.

2.2.5 Combustion Bombs

A combustion bomb is a fixed volume device that can either be operated as a homogeneous device (stirred reactors) or non-homogeneous as in the case of the IQT™. The IQT™ (ignition quality tester) is a combustion bomb that is used to calibrate the cetane value of diesel type fuels. In the case of the IQT™ the difficulty of fuel characterisation is that during injection the local air-fuel ratio changes through out the bomb, the temperature also changes due to the evaporation of the fuel.

Below is a table of some experimental devices and the pressure and temperature ranges at which they can operate.

Table 2-2: Examples of experimental work including the temperatures and pressures

Flow Reactor		
Reference	Temperature Range	Pressure Range
Lefebvre [1986]	up to 1133K	1-10 atm
Held [1997]	940 - 1085K	3 atm
Wang [1999]	610K - 810K	8 atm
Samuelsen [2003]	915 - 922 K	4 - 5 atm
Shock tubes		
Reference	Temperature Range	Pressure Range
Blumenthal et al [1996]	800 - 1335K	3 - 14 atm
Hidaka et al [1999]	1350 - 2400K	1.6 - 4.4 atm
Brown and Thomas [1999]	1073 - 2211K	20 atm
Rapid Compression Machines		
Reference	Temperature Range	Pressure Range
Ribaucour et al [1999]	640 - 900 K	0.4 - 0.5 atm
Lee and Hochgreb [1998]	1200 K	69.1 atm
Clackson et al [2001]	875 -1000 K	16 - 20 atm
Motored Engines		
Reference	Temperature Range	Pressure Range
Douaud (1978)	1010 K	22 atm
Combustion Bombs		
Reference	Temperature Range	Pressure Range
Advanced Engine Technology [2002]	565 K	21.09 atm
Hoskin et al [1992]	600 - 1300 K	30 atm
	up to 650 K	15 atm
Siebers [1985]	700 - 1200 K	20 - 50 atm

2.3 CFD

2.3.1 Autoignition Model

At the start of this work, there was no autoignition model available in Fluent®. During the progression of this work a new version of Fluent® became available. In the new version an autoignition model is included. This model makes use of the ignition integral model as proposed by Douaud (1978). In the process to make the modelling process simple it offers a clear GUI (Graphical User Interface) with options to select octane number and the constants A, n and B. The interface appears as follows:

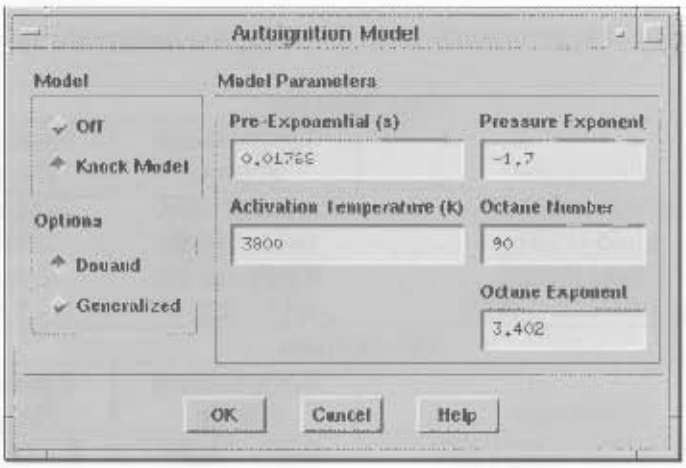

Figure 2-4: Fluent® autoignition model

As can be seen, this model provides a simple model for autoignition but no versatility in defining the underlying characteristic function. In the interest of fuel research more flexibility in the equation would be desirable.

As discussed previously the model proposed by Douaud (1978) is limited in accuracy, and is inappropriate for the modelling of diesel engines. This model may be useful for simple engine models but in the case of fuel modelling it is very limited. The generalized model (as referred to by Fluent® is the similar as the Douaud (1978) model but it includes other parameters as can be seen in equation 2-8.

$$\tau = A\left(\frac{OCTN}{100}\right)^{\alpha} p^{b} T^{c} RPM^{d} \phi^{e} \exp\left(\frac{-E_{a}}{RT}\right) \qquad \text{Equation 2-8}$$

Equation 2-8 is available for both diesel and petrol type models. As discussed earlier the octane scale refers to the petrol model and is not reconcilable with the cetane scale, it is not discussed in the help files what values to use for the octane number when modelling a diesel type fuel.

The default values for all the exponents (a, b, c, d and e) in the general model are one. for this model to be used with any confidence these values can not be used. To highlight this, if the T, RPM and phi terms are given an exponent of 0 then the equation becomes the equivalent to the Douaud (1978) equation. the pressure term in the Douaud (1978) equation

is -1.7, then the calculated value for the exponent for the phi term is -0.7887 for n-heptane, which indicates that the air/fuel ratio will have the inverse effect using the default generalised equation.

There is a third model available in Fluent® which is specifically used for diesel type models.

$$\tau = \left(\frac{0.36 + 0.22 S_p}{6 \times RPM}\right) \exp\left[E_a\left(\frac{1}{RT} - \frac{1}{17190}\right) + \left(\frac{21.2}{p - 12.4}\right)^n\right] \qquad \text{Equation 2-9}$$

This is an equation developed by Hardenburg and Hase (1979). This model is a global model and can not give accurate results for local ignition delay as is the case in a CFD model. The reason for this is there is no air/fuel ratio term. In a diesel engine this is fundamental aspect which needs to be modelled, as there is a fuel sprayed into a chamber of air, the spray plume having a very high fuel/air ratio and the rest of the chamber being air.

The models provided by fluent must be used with great care, especially when choosing the model to use and defining the values.

3 CFD background

3.1.1 Numerics – background

Computational Fluid Dynamics (CFD) is the analysis of any system involving fluid flow, heat transfer and phenomena including chemical reactions (combustion). CFD uses numerical techniques to simulate these real life events. This process has a wide area of applications, from engine modelling to supersonic flight. CFD has the advantage of being able to model areas and processes where experimental results are difficult or impossible to obtain. CFD can be used to reduce lead times and the cost of designs, as many prototypes can be simulated using CFD and only a few that do well in the simulations need be built to verify the simulation results.

CFD codes are written in 3 parts: Pre-processor, solver and the post processor.

The pre-processor is used to create the geometry of the simulation, as well as constructing the mesh for the flow domain. Included in this are boundary and fluid properties. The number of cells in the mesh governs the accuracy of the solution. Computational time increases as the mesh increases, so it is normally a compromise between mesh resolution and computational requirements. This normally results in a non-uniform mesh with a finer mesh in areas of high variation and a coarser mesh in areas with little variation.
The solver uses the file that is created by the pre-processor. Fluent® (the commercial CFD package used at UCT) discretises and approximates the variables using the finite volume method. This method is the most well established and thoroughly validated method and is used by most of the main commercial codes. This numerical algorithm consists of the following steps:

- Integration of the governing equations (transport equations) of the flow for each of the control volumes (cells of the mesh) of the solution domain.
- Discretisation involves the substitution of various finite difference type approximations for terms in the integrated equation for the different flow variables, so the integral equation is converted into a system of algebraic equations.
- The system of algebraic equations is solved using an iterative method.

In an unsteady event such as an engine, the model is divided into discrete time steps. Each time step is solved as a steady state event, the computer then moves to the next time step using the previous time step as the starting point to solve the current step. The computer then runs through all the steps and providing the time steps are small enough the unsteady event will be represented by the collection of steady state events.

The post processor simply contains visualisation tools; with the advent of more powerful personal computers the graphical facilities have been developed. Some of the tools included are:

Domain geometry and grid display
Vector plots
Contour plots
Particle tracking

For the contour and vector plots most variables (that are relevant) can be plotted.

3.1.2 Transport Equation

The process by which the CFD code solves all the variables is by making use of the transport equation. The transport equation is a generic equation by which all the variables can be described. This equation is as shown below:

$$\frac{\partial \rho \alpha_k}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i \alpha_k - \Gamma_k \frac{\partial \alpha_k}{\partial x_i}\right) = S\alpha_k \qquad \text{Equation 3-1}$$

$\frac{\partial \rho \alpha_k}{\partial t}$ Unsteady term.

$\frac{\partial}{\partial x_i}(\rho u_i \alpha_k)$ Convection term

$\frac{\partial}{\partial x_i}\left(\Gamma_k \frac{\partial \alpha_k}{\partial x_i}\right)$ Diffusion term

$S\alpha_k$ Source term

By choosing relevant coefficients for the different terms in the equation, the equation can be manipulated to give the differential equations (mass, momentum and energy), as shown below:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} \rho u = 0$$

Mass (Equation 3-2)

$$\frac{\partial (\rho U)}{\partial t} + \frac{\partial}{\partial x_i} \rho u U = \frac{\partial p}{\partial x} + \frac{\partial}{\partial x_i} \left(\mu \frac{\partial u}{\partial x_i} \right) + S_{Mx}$$

Momentum – x (Equation 3-3)

This can be done for all the different governing equations.

3.1.3 Meshing

There are two types of meshing when using the finite volume method, the Eulerian and the Lagrangian methods. Below is a basic explanation of the two methods. This is important to note particularly when using particle models in CFD.

The Eulerian meshing technique is the main technique used for the discretising a domain in CFD. The Eulerian method is a fixed mesh where the fluid is allowed to flow through the mesh. So properties of the fluid are calculated including convection and diffusion as in the transport equation.

The Lagrangian mesh method is generally used in solid mechanics, where the movements of the fluid are small. In the case of Lagrangian meshing there is no flow across the boundaries of the cells, in the case where there is movement the mesh moves with domain. In CFD there is a case where the Lagrangian mesh is used, this is in the case of discrete phase modelling (discussed in more detail further in the report), and the particle is designated a matrix containing all the physical properties, this matrix is attached to the particle.

3.1.4 Particles

"The Lagrangian discrete phase model in FLUENT® (described in Chapter 23) follows the Euler-Lagrange approach. The fluid phase is treated as a continuum by solving the time-averaged Navier-Stokes equations, while the dispersed phase is solved by tracking a large number of particles, bubbles, or droplets through the calculated flow field. The dispersed phase can exchange momentum, mass, and energy with the fluid phase." (Fluent® Help 2005)

This is a model that could be used in a number of applications, in this case to model the injection of fuel. A key aspect to remember when using a particle model it that the particles occupy a low volume fraction (less than ten percent). When using the particle model in the fuel air context there is an injection sub-model. This enables a global definition of the way in which the fuel enters the domain. An important detail to note it that the number of particles do not necessarily represent the number of droplets present, but represent a number of "packets" or groups of droplets. The formulation of this is as a single particle but the formulas for drag, evaporation, surface areas, etc are calculated for a group of droplets.

4 Methodology

4.1 Introduction

The purpose of this project was to create a simple code that would predict the start of autoignition within a CFD model. It was required that the code would not affect the model and would not have a great effect on the computational time.

Commercial codes provide a facility to this end as in the case of Fluent®, which can be incorporated with Chemkin® (a commercial chemical kinetics solver). The reaction mechanisms can be provided for Chemkin® and as the CFD model progresses it interacts with the chemical kinetics model to calculate the progression towards autoignition. It must be noted that the Chemkin® model is a science in itself and obtaining the kinetic models requires experimental data, and due to the complexity of the reactions within fuel, reduced mechanisms generally need to be used. The computational expense of running the two packages simultaneously becomes immense. The proposal for this project was to use the modified Arrhenius function, with its constants calculated using literature (experimental results) as the representation of the chemical kinetic model. This modified Arrhenius function is used in a CFD program to provide a simple, standalone method of calculating autoignition. During the period of this study a newer version of Fluent® became available and, as discussed in the literature review, includes an autoignition sub-model which operates in a similar manner to the methodology (the UDS method) proposed below.

A number of methods of integrating this modified Arrhenius function were investigated. These included using massless particles as placeholders to which the modified Arrhenius function was attached, a system of contour plots of the modified Arrhenius functions added together using an execute command function and finally using a UDS (user defined scalar) which is a user defined transport equation.

4.2 Outline of the principles

4.2.1 Particles

The first method investigated was to create virtual particles. The particles were massless "particles" which simply acted as numerical place holders to which the modified Arrhenius integral was attached. The philosophy around which this was designed was that the autoignition progress of the fuel was the parameter which was needed to be tracked. The massless particle could be regarded as a small particle of fuel. The modified integral would also be attached to these "real" fuel packets

The method investigated for this concept, was to have each injected particle to release a massless particle at each time step. The massless particles would then undergo the same body forces that the released fuel vapour experienced. In this way the "particles" could trace the movements of the vapour. The change in the ignition integral was then calculated for each time step, using the conditions of the continuum of the cell in which the particle occupied for that time step. Thus the point of first ignition could be established when the autoignition parameter attained a critical threshold.

During the development of this system it became clear that the number of particles required to obtain a realistic picture of the autoignition sites would be large (in the order of thousands to hundreds of thousands). There was a matrix attached to each particle with all degrees of freedom included. This interacts with the continuum matrix and so the computational expense becomes prohibitively expensive. Other reasons for not continuing with this method included: poor resolution, complexity of accurately programming, simplicity of other methods and concerns that the particles would not maintain an accurate track of the fuel vapour (the vapour has a diffusion term; a single particle can not diffuse)

4.2.2 Contour Plot and execute commands

This method was formulated as a simple start to gain some insight into autoignition. This method made use of a tool provided in Fluent®, which enables the user to make a contour plot (custom field function0 based on simple mathematical operators and any of the variables calculated within Fluent®, and then this function can be plotted in picture format. In this case

the ignition integral for each step is the calculated variable. By making use of the execute command function this calculated variable can be added to another function that acts as the cumulative ignition delay.

This method works well in situations where there is no flow, such as shock tubes and homogenous combustion bombs. The method can not take into account the moving of the fuel and thus ignores the possibility of the "modified Arrhenius function" moving (with the fuel).

Due to this limitation the method was not developed further and it was therefore deemed to be inappropriate for the requirements.

4.2.3 UDS Method

Another method investigated reversed the philosophy of tracking the fuel. In this case the air was the medium that was followed. A scalar was attached to the entire continuum, the modified Arrhenius function integral. Since the function was attached to the air, special care was needed when including the air-fuel ratio, including code for the case where it was pure air.

The main advantage of this method was that there was only one new equation to be solved and, since it was the same throughout the simulation, one would not incur increasing computation expense as the job progressed, as was the case of the particle method. Since the scalar was applied to the entire continuum there were no areas that were excluded as was the case with the particle method

4.2.4 Implementing the UDS method

UDS is the acronym for User Defined Scalar. This was a tool provided in Fluent® to provide more versatility in the program for the wide number of applications that it can be used. The UDS is a transport equation as discussed previously. Since a UDS is a transport equation which is completely undefined, it is up to the user to define all the required variables to produce an equation which will operate in a way that depicts the phenomenon that is being modelled.

In this case the UDS was required to track the mass flow of the fluid (air and fuel vapour) within the complete domain as accurately as possible. This means that diffusion is not permitted and convection is set to track the mass flow of the fluid.

The progressive development in the modified Arrhenius function integral was included in this scalar by adding it to the source term for each time step. At the beginning of the time step, the increase in the modified Arrhenius integral was calculated and added to the source of the scalar for each cell. Then, during the time step, the scalar was able to convect with the fluid in the domain as it moved.

Because it proved effective, this method was selected for further development and discussion in the remainder of this report.

4.2.5 UDF and C^{++} Code

A User Defined Function (UDF) is a library of different utilities. It operates much like a macro in Excel®, and can be used with loops, 'if functions' and any functions that can be used in normal code. The UDFs are programmed in C^{++} code. All the versatility of C^{++} is included and can be used in the UDF. The UDF simplifies the programming as it facilitates the interaction between the code and Fluent®. It also contains a large library of functions that can be invoked depending on the requirements of the user. When using a UDF, all the variables calculated by Fluent® are available.

The specific UDF library used in this case was the source function. Custom source terms could be used for any transport equations, for example a numerical fan, where momentum source is added in the area where fan blades would pass. In this case it was used on the UDS, to add "used up" ignition delay to the UDS.

As the define_source function is generic in nature, some of the functionality is not required for this case, but for the completeness it will be briefly described below.

DEFINE_SOURCE(name, c, t, dS, eqn)

name refers to the name of the UDF, as it will be seen in Fluent®

c Defines the cell to which the source term will be applied, within the specified thread.

t Pointer to the cell thread, a common group of cells

dS The derivative of the source term can be included if the source term includes the dependant variable of the transport equation

eqn Equation number

As discussed earlier the modified Arrhenius function is included by adding the change in the integral to the UDS at each time step. The following is a discussion about the actual implementation of the source term.

Since the modified Arrhenius function operated over the entire domain, the cell thread and cells do not need to be defined, Fluent® then defaults to looping over the entire domain. Because the modified Arrhenius function source term does not depend on the value of the modified Arrhenius function itself, the source term can be calculated explicitly, so the dS value is set to 0.0.

The source was set as equal to the Modified Arrhenius formulation. The UDF then returned a single real value for each cell. This is hooked into the defined scalar equation.

Hook is a term used by Fluent® that refers to the way in which the UDF is invoked by the program. In this case the UDF is included in the boundary conditions as a condition for the entire domain, thereby defining the cells and thread.

The diffusion term (as discussed previously) of the scalar transport equation must be defined. This is defined in the material model. It must be noted that the diffusion value must be defined for all the materials; this includes the fluid and the mixture. Since the scalar needs to trace the fuel mixture exactly the diffusion term must be defined as zero.

The convection term needed to be defined. This is defined when the UDS is first selected. The pop-up menu requests the number of UDSs to be defined, the method of convection and the definition for the unsteady term. The method for convection was selected as mass flow. When this was selected it is implied that any convection of mass means the same proportion of the UDS flows and with the same velocity as the mass flow. In Fluent®, the default

unsteady formulation is defined. This takes into account that there will be an increase in the source term and thus operates to keep the equation balanced for this increase.

For details on the specification of the UDS and the actual code for the UDS, refer to the tutorial in the appendices.

4.2.6 Excel®

In order to check the CFD model, a very simple Excel® model was created. While this does not prove that the modified Arrhenius function is the correct model to use, it provides a check that no numerical effects are created in the CFD model. The model created in Excel® was a simple one-dimensional system that calculates the ignition delay at a constant air-fuel ratio, pressure and temperature. All these variables can be changed to test a wide range of applicability.

The model was subsequently upgraded to include the possibility of step changes, in other words from one pressure and temperature for a set amount of time to another. This could then be simulated in the CFD model and the responses could be validated.

4.3 Implementation

4.3.1 Steps

The first point of departure after selecting the most appropriate method was to perform some basic tests of rationalization. Due to the complexity of the injection process, and the effect it has on all the important properties, air-fuel ratio, pressure and temperature, the initial tests were without injection. It this way all the variables could be controlled as known values. This was done by creating a simple tube: geometrically with a relatively narrow diameter. A flow was created through this domain, with a known velocity. The velocity and the length were mapped onto the ignition delay scalar.

The method used for developing the code will be discussed below. The steps followed were:

1. Using a constant value for the source term.
2. Using an Arrhenius function without an air-fuel ration effect.

3. Including the air-fuel ratio effect.
4. Injecting a single particle.
5. Validating the model using experimental data.

Before implementing the Arrhenius function code it was deemed important to check that the operation of the particle model worked as expected. This check was required because of the complexity of the particle model and the options available to the user in the way the model was implemented. The most important aspect of this was the interaction between the two meshes. It was possible to set up this interaction in various ways depending on the nature of the experiment being simulated. In this case a transfer of the physical properties was required between the particle and the continuum. The figures 4-1 to 4-3 illustrate the transfer of variables from the continuum to the droplet and visa-versa.

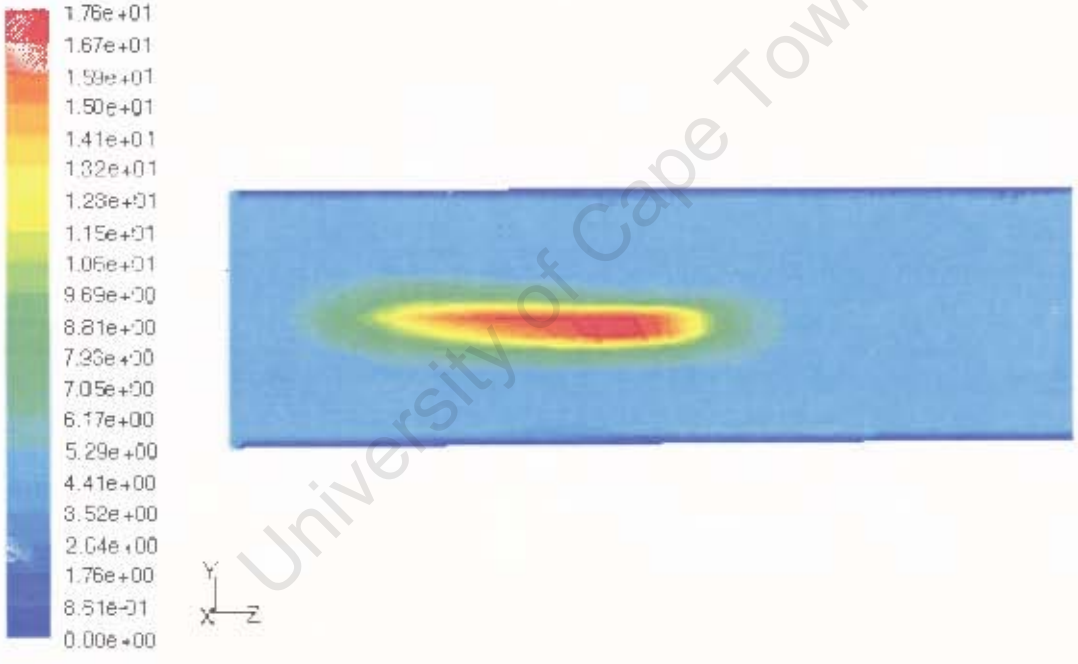

Figure 4-1: Velocity contour plot from a single droplet injection

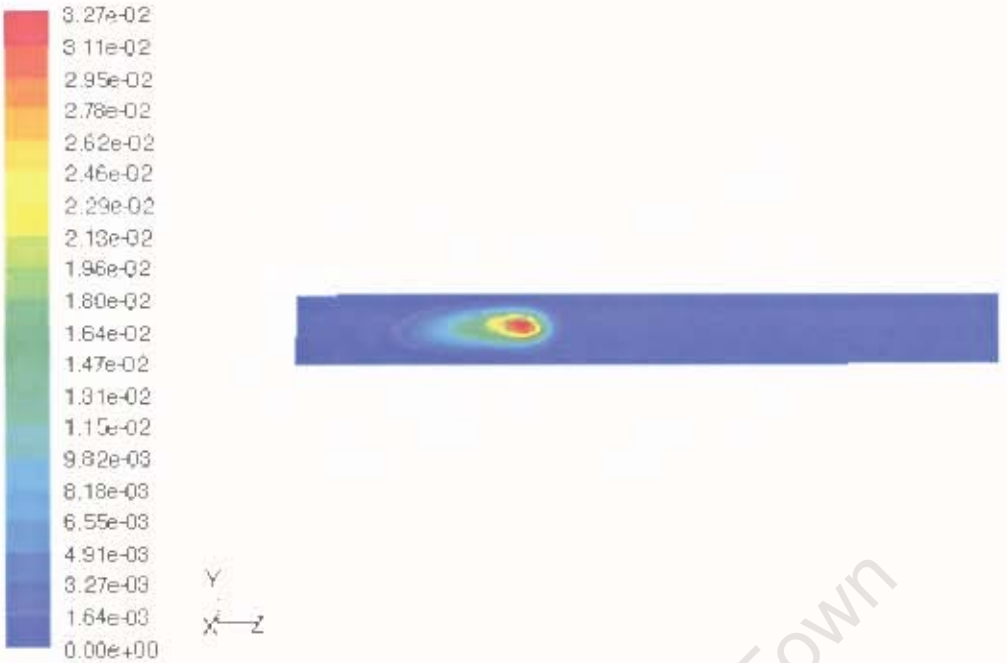

Figure 4-2: Fuel-Air ratio (phi) plot from a single droplet injection

Figure 4-3: Temperature plot from a single droplet injection

Figure 4-1 is the contour plot of the velocity magnitudes, indicating a transfer of momentum. Velocity from the droplet being imparted to the continuum, due to the drag exerted on the droplet, the velocity of the droplet consequently decreases. The following figure (figure 4-2) shows the species contour plot, indicating mass transfer through the evaporation of the droplet. Note that the droplet does decrease in size as the evaporation progresses. The third figure (figure 4-3) shows a temperature plot and demonstrates that energy is transferred from the continuum to the droplet (in the form of thermal energy).

While these illustrations confirm that there is a successful transfer of information from the particle to the continuum there are no definitive values given to make sure that variables such as momentum, etc are conserved. It had to be assumed that if there was transfer taking place that the program was operating as expected, and so it was assumed that Fluent® was doing the calculations correctly.

A further test was conducted to check that the mass transfer occurs correctly. A simple tube was used with a constant flow of air through it. Then a continuous injection took place. The fuel flow rate of the particles was known. The simulation was set up in such a way that the particles evaporated completely before the end of the tube. The flow of fuel out the tube was then calculated. The test verified that all the fuel from the Lagrangian mesh transferred to the Eulerian mesh.

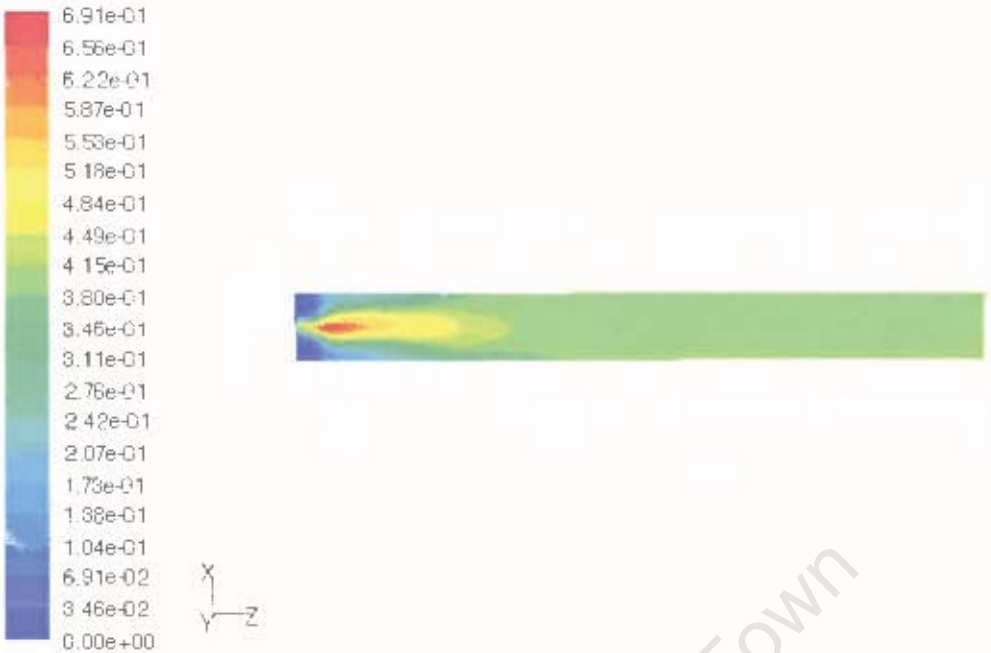

Figure 4-4: Contour plot of species from a continuous injection

The flow of species through the outlet was reported. This was compared to the flow of species into the tube. It was found that the results did not match exactly, but the average flow out matched the inlet flow: the small differences were due to slight fluctuations in the flow.

4.3.2 Designing a Simple model

To test that the UDS would operate in the desired manner, the source term was set to a constant value, in this way it could be seen if the UDS was working as expected. For this part of the model the conditions (pressure, temperature) do not affect the results and simply specifying the velocity flow rate was sufficient. This test highlighted a nuance of the transport equation. The source term is divided by the density when incorporated into the transport equation. This was clear in this simple model. If this check had been bypassed locating the source of subsequent errors would have been almost impossible to resolve, especially in conditions where the density was close to unity and the effect small. The constant had to be multiplied by the density of the cell to return the expected result. This was not obvious when studying the transport equation.

Once the simplest model was running correctly, a simple Arrhenius function was evaluated. These results were compared with the Excel® model to verify that the Arrhenius function was calculated as expected. At this stage the pressure and temperature needed to be specified to predetermined values and needed to remain constant throughout the domain and for the entire duration of the simulation. The only effect in the model was the velocity in the tube, effectively giving the variation in time for the Arrhenius integral.

At this stage it was discovered that the pressure term in the Arrhenius expression was proving to be problematic. The reference to pressure in the UDF refers to the relative pressure: the pressure above the operating pressure. This was resolved by including a small amount of code to call in the operating pressure.

Air-fuel ratio effects were then included in the model. This required additional code to calculate the air-fuel ratio (equivalence ratio or phi), since Fluent® only returns mass fractions. This also meant that the stoichiometric air-fuel ratio was required in the code.

4.3.3 Comparing CFD to the Formula Calculation

This involved in the simplest case a tube with a predetermined flow rate. The temperature and pressure in the tube were adjusted to test specific situations. The model was run until a steady condition was obtained and the position where the integral reached unity (see chapter 2-1) was then used to determine an ignition delay. This was then compared to the formula result to verify that the results were as expected. The CFD model in this case was globally a one dimensional model, and is thus directly comparable to the formula.

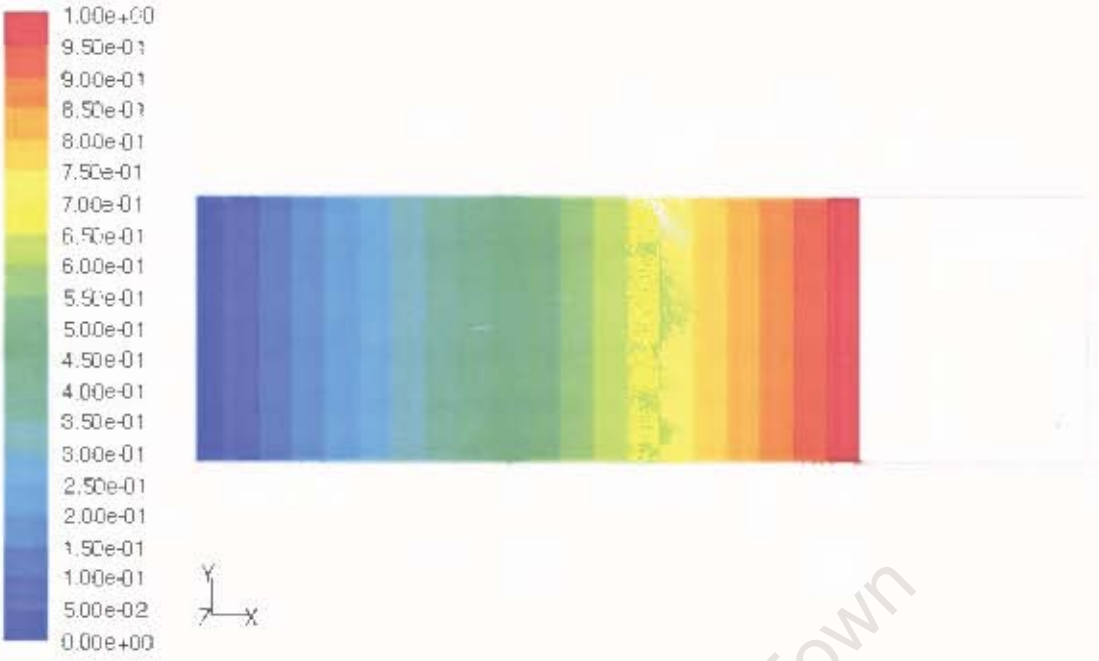

Figure 4-5: Integral development

In this model the air-fuel ratio was homogeneous, as was the temperature and pressure. The temperature was 844K, the pressure was 32 bar and the fuel-air ratio was phi equal to 0.8. These values were chosen to be similar to the test conditions for heptane in the cetane test, although the values could have been arbitrary as this was simply for comparison with the calculation. The phi value was taken as non-stochiometric to include the fuel-air ratio in the test; it was chosen to be fuel lean as this is where overall diesel combustion occurs. The velocity of the flow was 97 ms^{-1}, the distance to where unity was reached was 0.7498 m and therefore the ignition delay could be backed out as 7.73ms, this was exactly the same value as calculated (to two decimal places) using the formula. Although this shows that the CFD model predicts the same value it does not validate the formulas used. However, it demonstrates that the simple model using the complicated numerical calculations are not in error when compared to a formula result.

5 Validation

5.1.1 Experimental validations

Validation of the Fluent® code by means of a continuous flow simulation was chosen for its simplicity. In the simulations investigated the fuel is injected as a gas, through multiple holes to reduce the mixing time to as short as possible.

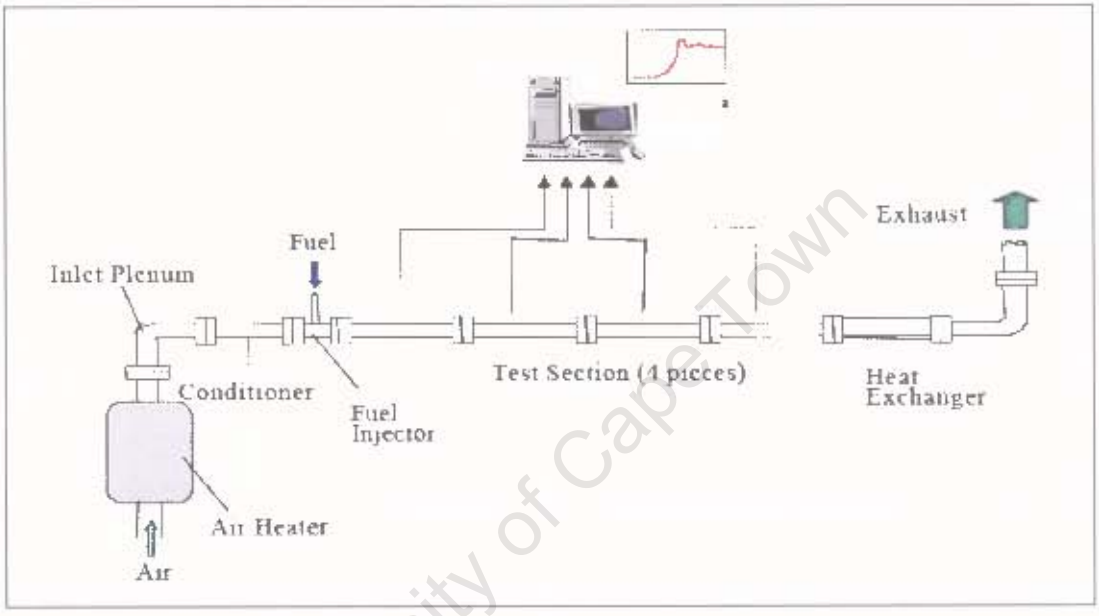

Figure 5-1: Schematic of the experimental set up (Samuelsen [2003])

The experimental set up is described by Samuelsen (2003). The air heating was achieved in two stages. The first stage comprised of an in-line 18kW heater. The second stage comprised of clam type furnace tube heaters situated along the outside of the test sections to actively counter heat loss. The fuel of particular interest for this work was methane. By injecting gaseous fuels the simulation was simplified because particle injection was not required. The fuel was heated using a 7.2 kW tube furnace. The test section was created by four sections of pipe, three of the sections were 863.6mm long each and had diameters of 38.1mm. The fourth section had ignition sensors in it and was 1143mm long. The total test length was 3733.8mm long, although this was adjustable by removing one or more of the three sections. Ignition was sensed by making use of six photodiodes and thermocouples, placed axially along the length of the test pipe. The injector used for the test procedure was a venturi type system, with injection holes around the circumference and an injector in the centre, with the fuel split 35% to the centre injector and 65% to the circumference.

Measurements determined that the stream was homogeneous (within 1% concentration deviation) in two diameter lengths.

Table 5-1 provides the results of the experimental work (Samuelsen, 2003).

Table 5-1: Experimental results (Samuelsen [2003])

Run	Ignition	air	t air	Methane	Temperature fuel	Temperature mixture	Pressure	Velocity mixture	ID min	ID max	ID (lit)
		m/s3	Kelvin	m/s3	K	K	kPa	m/s	(ms)	(ms)	(ms)
1	No	0.0132	927.6	0.0011	887.6	922.1	202.7	19.53	176	-	323
2	No	0.0132	924.8	0.0011	888.7	919.8	253.7	15.58	220	-	266
3	No	0.0132	929.3	0.0011	885.9	923.2	304.0	13.03	263	-	214
4	No	0.0132	930.4	0.0011	888.2	924.3	355.0	11.19	306	-	179
5	Yes	0.0132	927.1	0.0011	891.5	922.1	405.4	9.76	-	279	162
6	Yes	0.0132	928.2	0.0011	887.6	922.6	430.9	9.19	-	297	152
7	Yes	0.0132	927.1	0.0011	886.5	921.5	456.4	8.67	-	315	144
8	Yes	0.0132	921.5	0.0011	889.8	917.1	506.7	7.77	-	351	138
9	No	0.0132	926.5	0.0011	891.5	920.9	202.7	19.65	174		-
10	No	0.0132	927.1	0.0011	890.4	920.9	304.0	13.1	262		-
11	Yes	0.0132	927.6	0.0011	888.2	920.9	405.4	9.83	-	278	-
12	Yes	0.0132	924.3	0.0011	888.7	918.7	506.7	7.84	-	348	-

Due to six defined measurement points, the exact point of ignition can not be defined. In the experiment results table above, autoignition either occurred before the sensor array or autoignition did not occur at all. In the event of autoignition occurring, the time given as ID max was the time taken for the flow to reach the first sensor, as it was known that ignition occurred within this time. In the event of autoignition not occurring, the time given as ID min was the time taken for the flow to reach the last sensor, as it was known that ignition did not occur within this time. While it could be argued that the experiments are incomplete, they do provide limits to when autoignition occurred, despite the complexity of autoignition and the dependence on various parameters that controlling the position of ignition is difficult.

The ignition delay times according to literature were given referenced to Li and Williams (2000). The equation is as follows:

$$\tau_{ign} = \frac{2.6\times10^{-15}[O_2]_0^{-\frac{4}{3}}[CH_4]_0^{\frac{1}{3}}}{T_0^{-0.92}\exp(\frac{-13180}{T_0})}$$

Equation 5-1

The reference did not specify the units of the equation. In this project this equation was not used except to refer to the ignition delay values given in the work of Samuelsen (2003).

5.1.2 Arrhenius function

In the creation of the continuous flow simulation the Arrhenius function constants were required and were determined in the following manner. Since methane is a single stage reactant, it does not exhibit the negative temperature coefficient region. This means that it can be explicitly defined with a single Arrhenius function. Data was required to determine the characteristic constants. This was done in two ways. In the first case, flow reactor data was used, this data was extracted from experimental results postulated by Lefebvre in 1986.

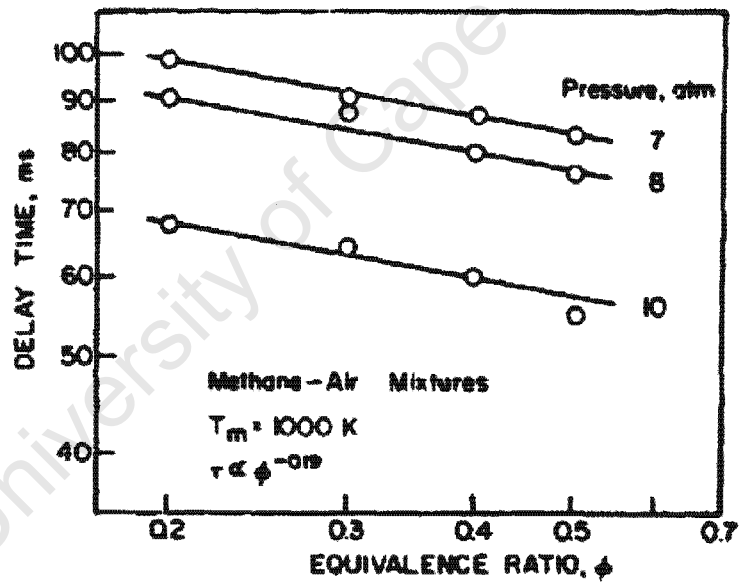

Figure 5-2: Experimental data from Lefebvre (1986) for varying phi

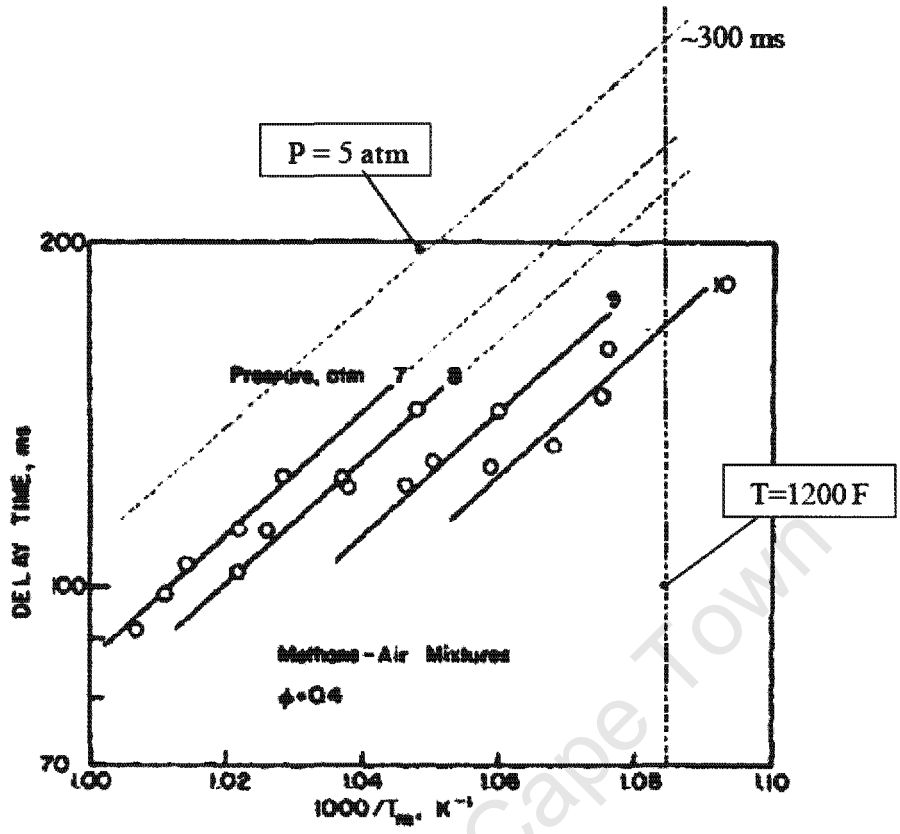

Figure 5-3: Experimental data from Lefebvre (1986) for phi = 0.4

The data above was scanned and digitized, and the constants for the Arrhenius function were solved as best fit to the data points. Solving these data points to an Arrhenius function resulted in the following equation:

$$\tau = 0.003649 p^{-1.0065} \exp\left(\frac{11922.98}{T}\right) \phi^{-0.15} \qquad \text{Equation 5-2}$$

A second Arrhenius function was deduced from a reaction mechanism provided by the Leeds University (Hughes, 2001) and this reaction mechanism was processed in Chemkin® (simulated as a numerical homogeneous constant volume combustion bomb). The ignition delay data was then resolved into an Arrhenius function.

$$\tau = 0.00002457 p^{-0.74} \exp\left(\frac{15575}{T}\right) \phi^{-0.2335} \qquad \text{Equation 5-3}$$

These functions were put into an Excel® model. Then conditions (temperature, pressure and air/fuel ratio) were used which represent the experimental data points. The model is a simple homogeneous one-dimensional model. The results are indicated below.

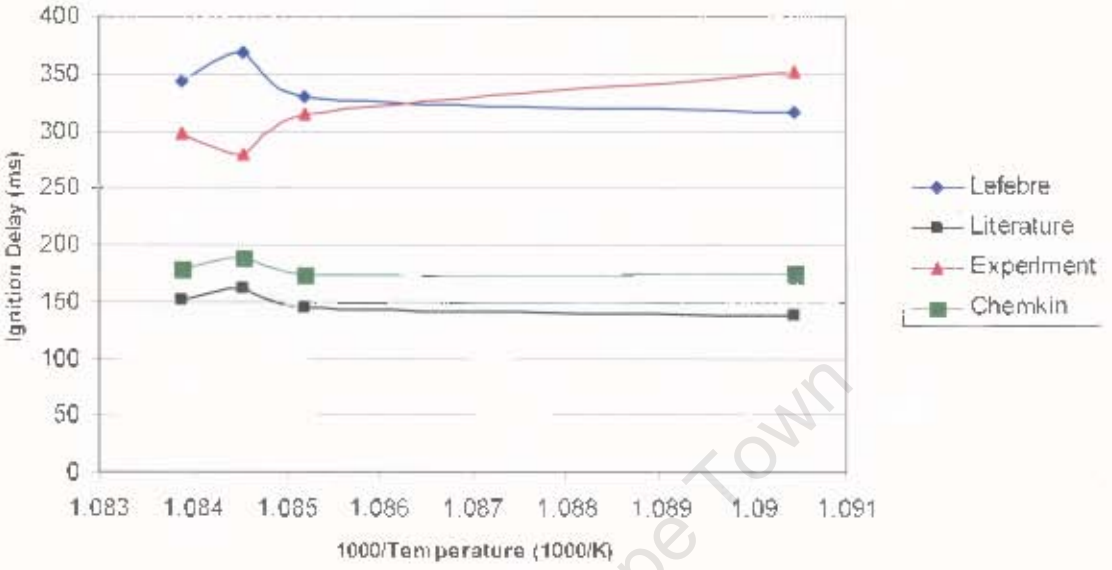

Figure 5-4: Excel® calculation of different models compared to experimental data points

As can be seen in the graph, the range of temperatures of the experiments is limited. The air-fuel ratios for above are all phi equal to 0.8. An effect that is included is pressure. The graph above represents a very small temperature and pressure range and so the values are subject to much experimental error over this small area.

It was decided to use the Lefebvre (1986) and the Chemkin® data for the Fluent® model. The Chemkin® data was chosen to be modelled as this represents a pure fuel model. The Lefebvre (1986) data was used as it was experimental data from a similar type of experiment.

Two simulations were modelled: first, a simple tube, with a homogeneous mixture entering the inlet; second, including a model of the injector to investigate the effect of mixing.

The simple model in Fluent® comprised of a pipe with a diameter of 38.1mm and a length of 3734mm, representing the test section of the flow reactor. The inlet was defined as mass flow inlet, with a flow rate calculated to give the correct velocity through the tube. The inlet was a homogeneous inlet with an air/fuel ratio set across the inlet. The outlet was defined as

a pressure outlet set to give the operational test pressure. The phi value was 0.8, the temperature was 922.1K and the absolute pressure was 405.5 kPa. This simulated one of the experiments described previously (Table 9-1, run 5). Below is a cross section of the flow tube with contours representing values of the Arrhenius integral.

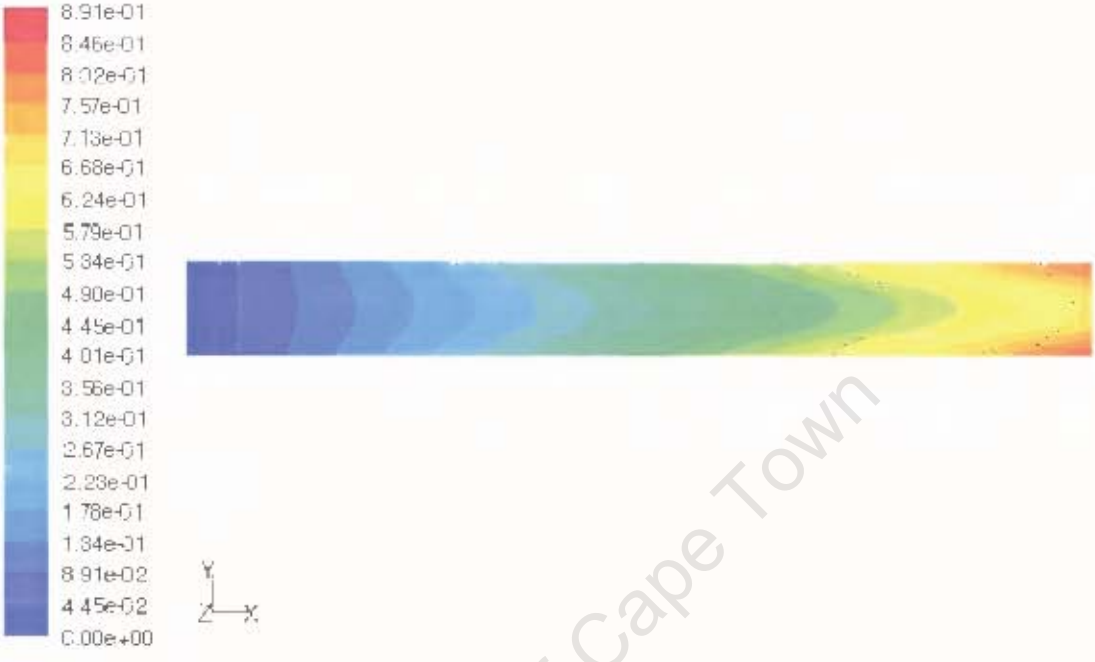

Figure 5-5: Contour plot of Lefebvre (1986) Arrhenius Function

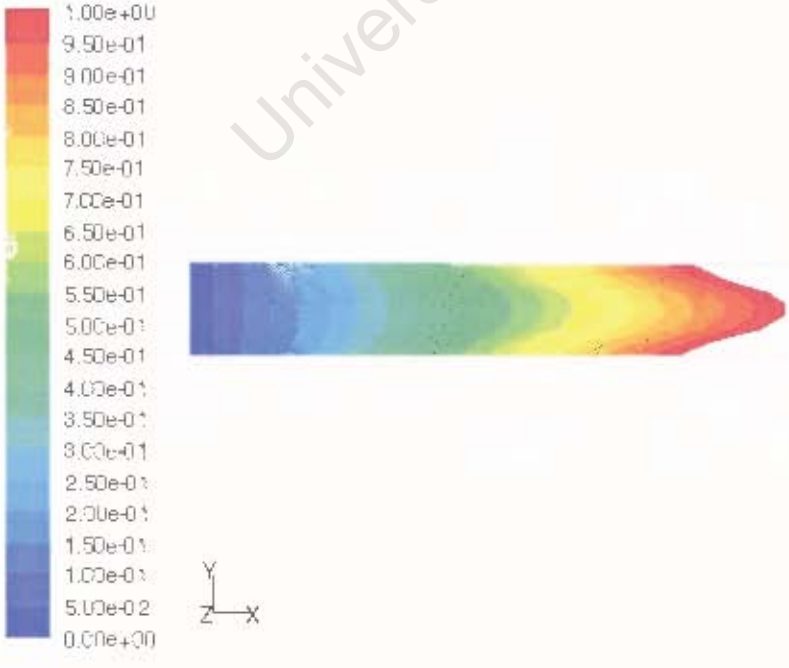

Figure 5-6: Contour plot of Chemkin® Arrhenius Function

The two simulations revealed very different results. The integral scale was limited to a maximum of unity so if the integral exceeded unity the area remains blank (white) which indicates ignition. Figure 5-5 (Lefebvre (1986) data) did not reach autoignition in the test section, where as the Figure 5-6 (Chemkin® data) reaches autoignition at the location two thirds down the tube. The physical location of autoignition is illustrated in the figure, but it was more useful to obtain an exact numerical value of the location. In Fluent® there is no direct method of getting position scales in the contour plots. It can, however, be obtained using an x-y plot, where the x axis is a direction axis of the domain and the y axis can be the value of a variable. These plots are depicted in Figure 1-7 and 1-8.for the simulations described above.

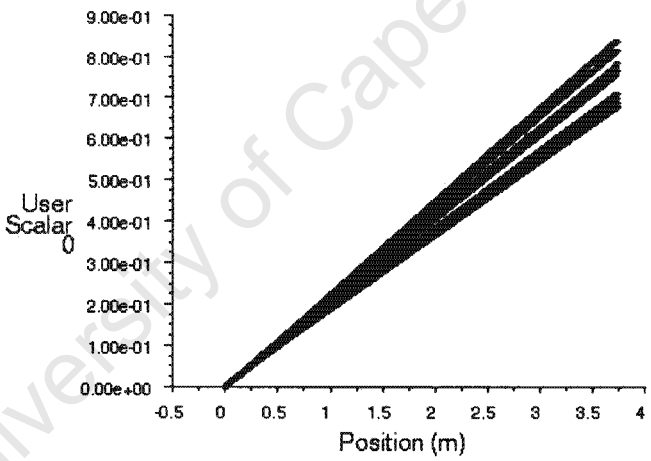

Figure 5-7: x-y plot of the Lefebvre (1986) Arrhenius function – user scalar is the ignition-delay integral (= 1 at autoignition)

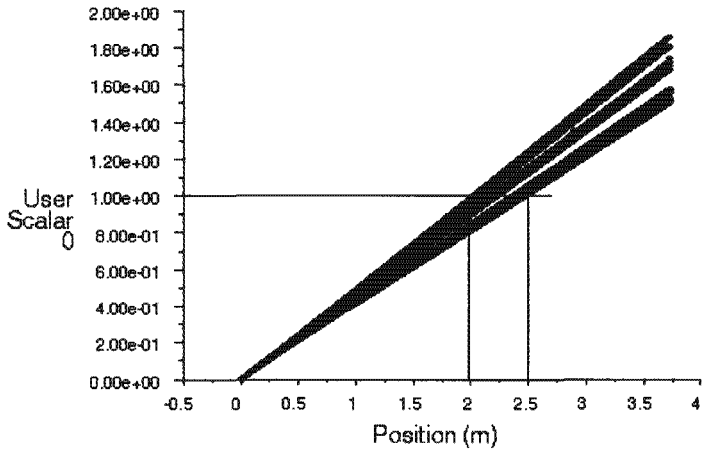

Figure 5-8: x-y plot of the Chemkin® Arrhenius function – user scalar is the ignition-delay integral (= 1 at autoignition)

In figure 5-7, the ignition integral attained a maximum value of 0.85 at the pipe exit. In figure 5-8 unity is attained within the domain. The lines included on the graph are for the integral equal to one and the corresponding minimum and maximum distances from the injection. The reason for the spread of the x values is due to the boundary layer effects. As a result of the homogeneity of the temperature, pressure and air/fuel ratio, all the x positions for the integral equal to one represent the same amount of time. The shortest x values represent the flow close to the boundary, which is a slower velocity, and then the largest x value represents the flow close to the centre of the tube, which has the fastest velocity. The fastest velocity in the flow is 15.4m/s and the average velocity is 10.95m/s. This corresponds to an ignition delay time; using the fastest flow and the largest x value gives an ignition delay of 164 ms, using the average flow rate and the mid point x value gives an ignition delay integral of 203.2 ms. This can be compared to the calculated ignition delay (using Excel®) which is 210.17ms and the experimental result of 276ms. It can be assumed, due to the simplicity of the model, that the CFD result should be close to that of the Excel® results, as in the case of the earlier validations, so the average result appears to give a better representation of the results.

The second model included the injector and the effects of variable air fuel ratios and temperatures. The tests were the same as above using the function created from the data of Lefebvre (1986) and the data from Chemkin®. The figures below show the effect of the mixing including the variation air fuel ratio and temperature.

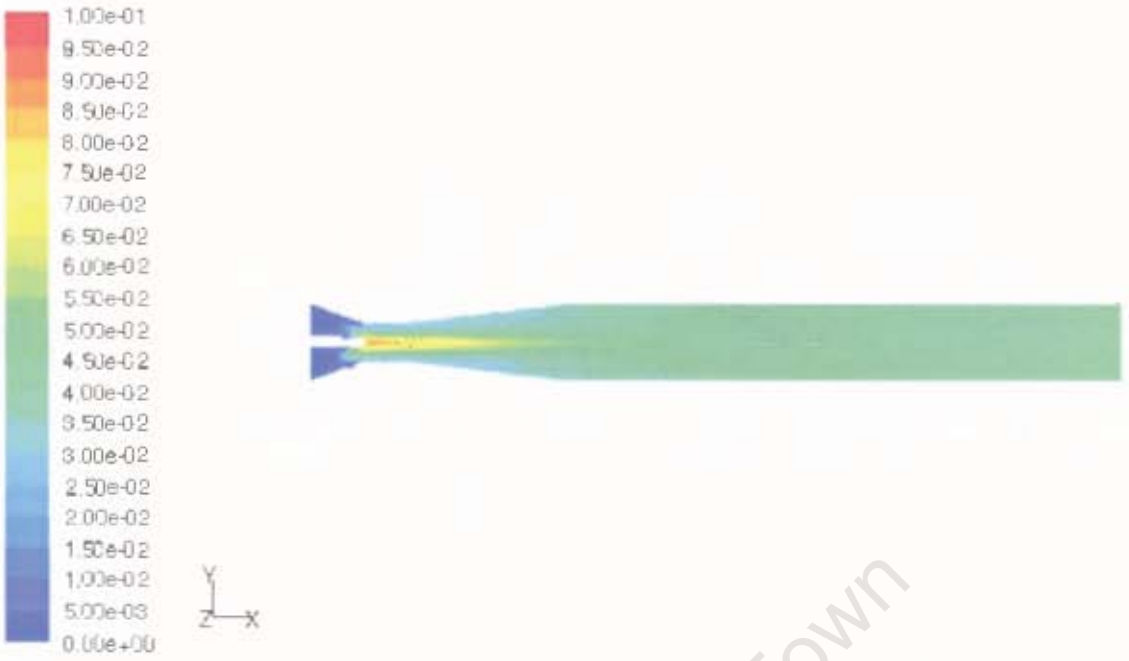

Figure 5-9: Contour plot of methane mass fraction in the continuous flow reactor

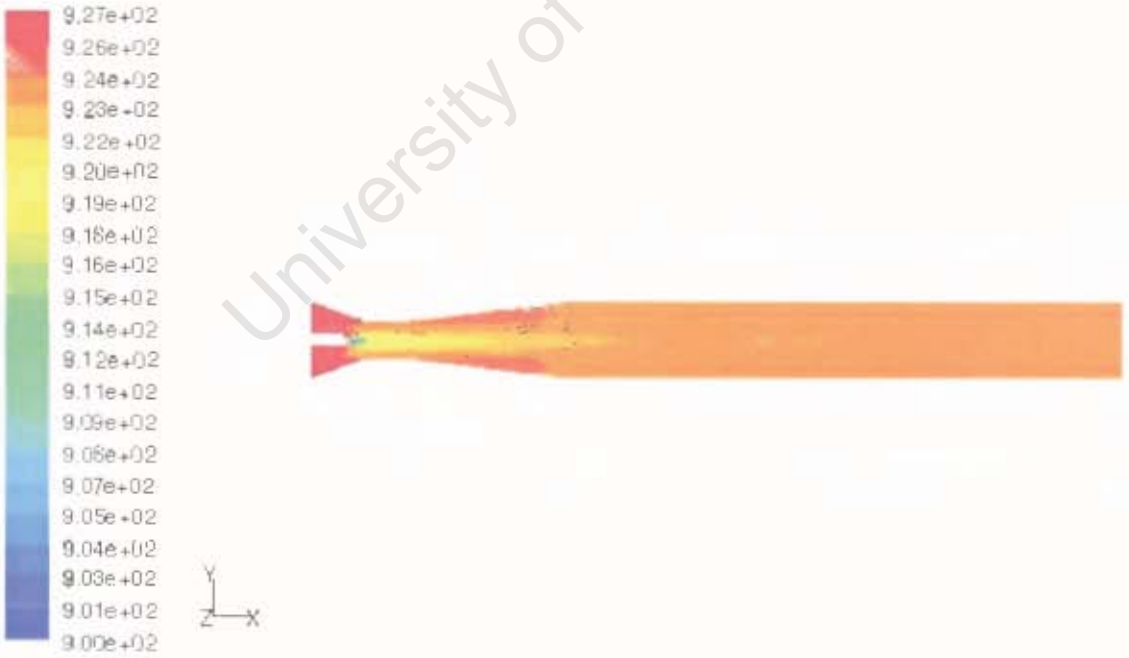

Figure 5-10: Contour plot of temperature in the continuous flow reactor

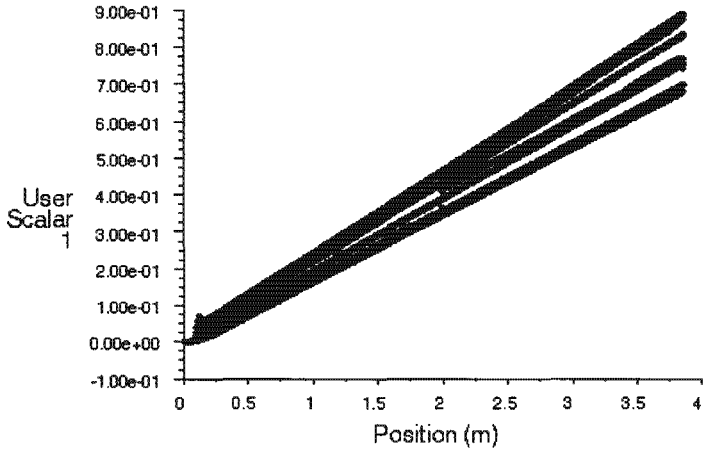

Figure 5-11: Ignition delay using the data from Lefebvre (1986) – user scalar is the ignition-delay integral (= 1 at autoignition)

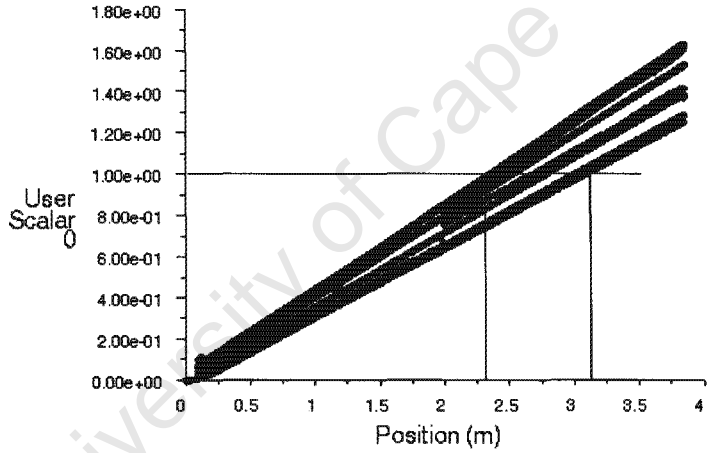

Figure 5-12: Ignition delay using the data from Chemkin® – user scalar is the ignition-delay integral (= 1 at autoignition)

Again the ignition delay times can be calculated and again the Lefebvre (1986) data did not reach unity and so can not be used. The maximum velocity was 15.2 m/s and the average velocity was 10.658m/s. Calculating the ignition delay from the Chemkin® data, based on the maximum velocity and the shortest distance, the calculated ignition delay was 199ms and using the average velocity and the mid point of the position, an ignition delay of 246.6ms was calculated. While this model is somewhat more complicated than the model above it is

assumed that the more accurate result will be the average velocity. This value approximates the experimental result more accurately than the literature.

6 Simulations and Discussion of Results

6.1 Petrol Models

Simple models for petrol type fuels were preformed in much the same manner as the simple model in the methodology chapter. This included homogeneous mixture, temperature and pressure. This results in effectively a non dimensional model that correlated with the ignition delay curves.

This does not prove to be of interest and the need for the extra computation expense is not validated. An area to which petrol models would be useful is in the case of direct injection engines and HCCI engines, in these engines the mixture is not homogeneous and a CFD model could provide useful insight into this effect. This would require a heat release model to be created. The ignition delay integral would then form the basis of the heat release model, this however falls outside the scope of this project.

6.2 Diesel Models

6.2.1 Model outline

Simple models are generally poor at describing the in-homogeneity of the diesel injection event and the subsequent combustion. The model discussed below investigates one of the simplest models including these in-homogeneities, a droplet evaporating in still air.

It was proposed to produce a simple simulation to investigate if this would give an ignition delay that is comparable to experimental data. The experimental data that was of interest is the combustion of n-Heptane in the CFR engine (in the cetane test, see section 2-1). The simulation conditions were therefore defined to approximate this; the conditions will be discussed later. The reason for selecting this data was that the ignition delay is well known for n-Heptane over a wide range of conditions, the test conditions for the cetane test are well described and understanding the combustion in the cetane test is of interest to Sasol. As the air and fuel were given no initial velocities, a fine mesh was required to give sufficient resolution in the temperature and air/fuel ratio gradients. Due to the fine mesh it was decided to use a 2-D model to reduce the computational expense. Since the model is theoretically symmetrical it was decided that using a 2-D model would not greatly affect the results. The domain for the simulation was a square with sides of 30mm, the boundaries were specified

as pressure outlets so that the pressure remained constant through out the simulation. The in-cylinder conditions for the cetane test when n-Heptane is injected are: a pressure of 32 bar and the temperature, calculated using a simple engine model is 844 K. The experimental ignition delay for n-Heptane in the cetane test is 2.41 ms, which corresponds to 13° crank angle as is prescribed by the test procedure.

In the process of developing the simulation it was found that using a single large drop was numerically unstable. However, to create a sufficient air/fuel ratio for the duration of the simulation a large drop was required (mm in diameter), this was injected in a single time step and it created large pressure rises causing the simulation to fail. To resolve this issue, multiple droplets were used. It was noted that this would increase the surface area over which the evaporation could occur, so a droplet size representative of the CFR engine injection was used (diameter 0.0175mm). An injection duration of 1ms was used (i.e. the smaller droplets were injected over a longer period to make up for the increased evaporation rate), this duration was selected to make sure that the droplets were all evaporated shortly before the ignition integral reached unity (within 0.1ms).

An important aspect of the simulation was ensuring that the material properties for the droplet were correct. The boiling point was increased from 350K (at atmospheric pressure) to 550K which is the critical boiling point approximately matched to the prevailing pressure. The latent heat of evaporation was changed according to the formula provided by J.W. Rose and J.R. Cooper (1977), this is temperature dependant, but in the Fluent® package it was not possible to include a formula so the latent heat of evaporation was defined as 101.6 kJ/kg which is correct for 550K.

The constants that were used for this simulation were calculated using best fit data from Chemkin®, this best fit was also in good agreement with data points given by literature. These constants are given in table 6-1, the value used for the phi exponential was -0.7887.

Table 6-1: Constants for the modified Arrhenius function of n-Heptane

	A	n	B
1	7.67E-09	-0.078	14721.42
2	1.11E+08	-2.082	-10262.86
3	3.81E-06	-0.927	16441.21

6.2.2 Results of Model

The figures below show the temperature, fuel/air ratio (phi) and the modified delay integral respectively.

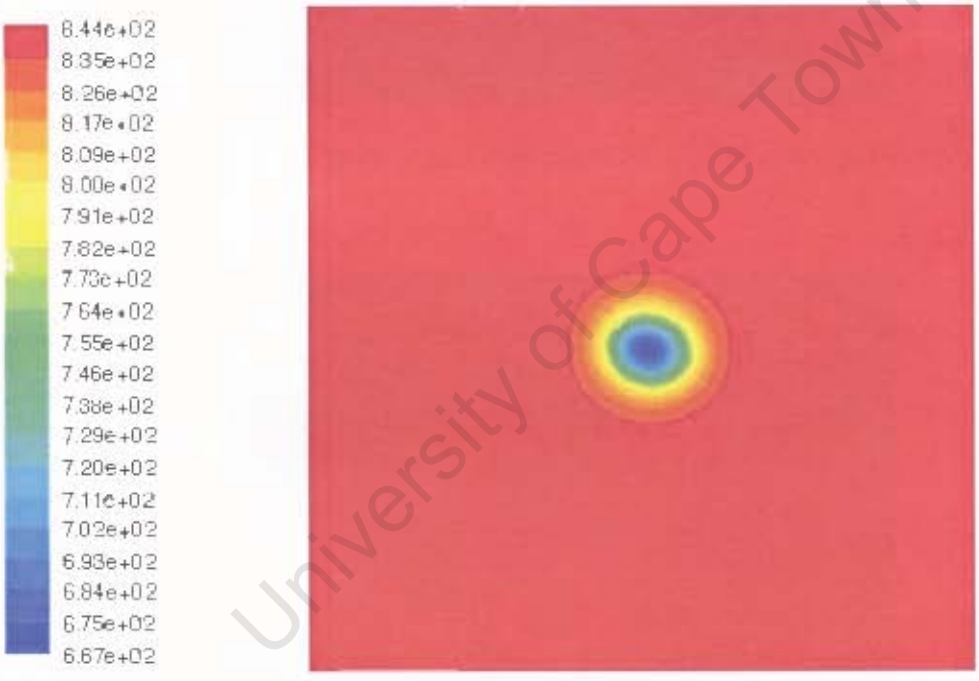

Figure 6-1: Temperature contour plot for a single drop

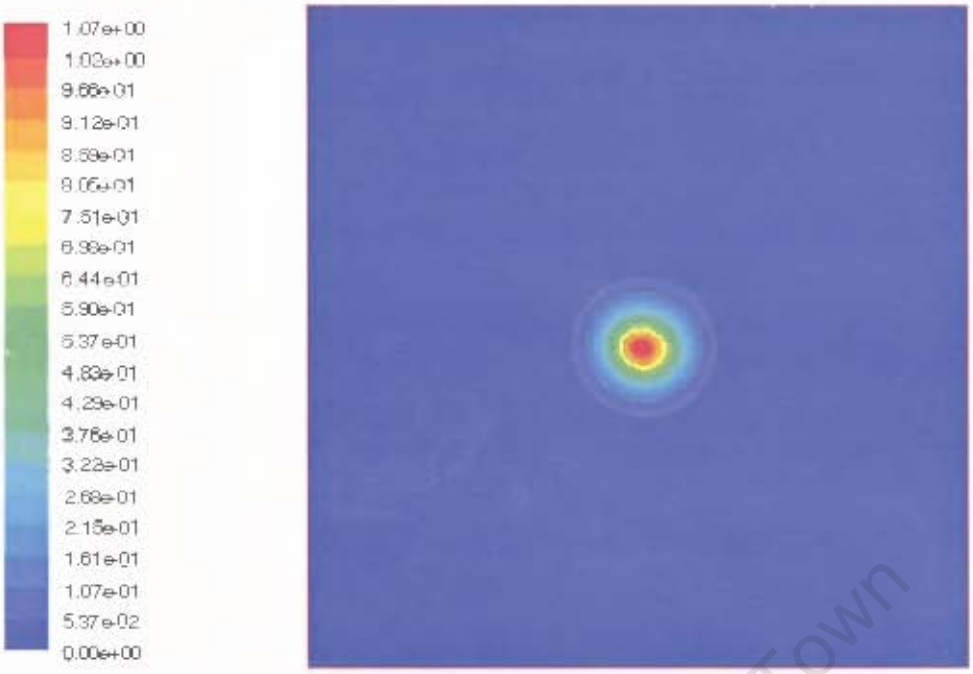

Figure 6-2: Phi contour plot for a single drop

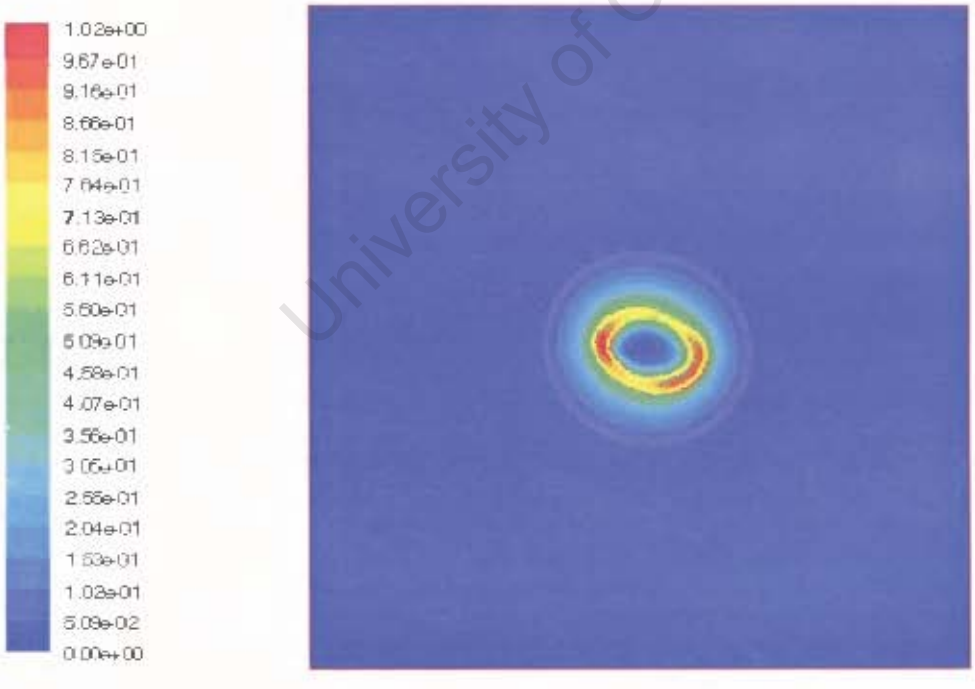

Figure 6-3: Ignition delay integral contour plot for a single drop

The figures above are taken at 3.0 ms, this is a single snap shot of an unsteady event so it does not describe the process entirely. During the injection, the temperature drops to

approximately 400K with a maximum phi value of 5. The assumption that the model would be completely symmetrical did not hold completely true. Looking at figures 6-1 and 6-2 the model appears symmetrical but in figure 6-3 it could be clearly seen that the model is not completely symmetrical. This highlights the sensitivity of the ignition delay to small changes in temperature and phi. Causes for the lack of symmetry could be due to turbulence effect and small numerical residuals which have a greater effect when the velocities are very low, as in this case.

The following plots are of the values of the variables which geometrically fall on the line A-A highlighted below:

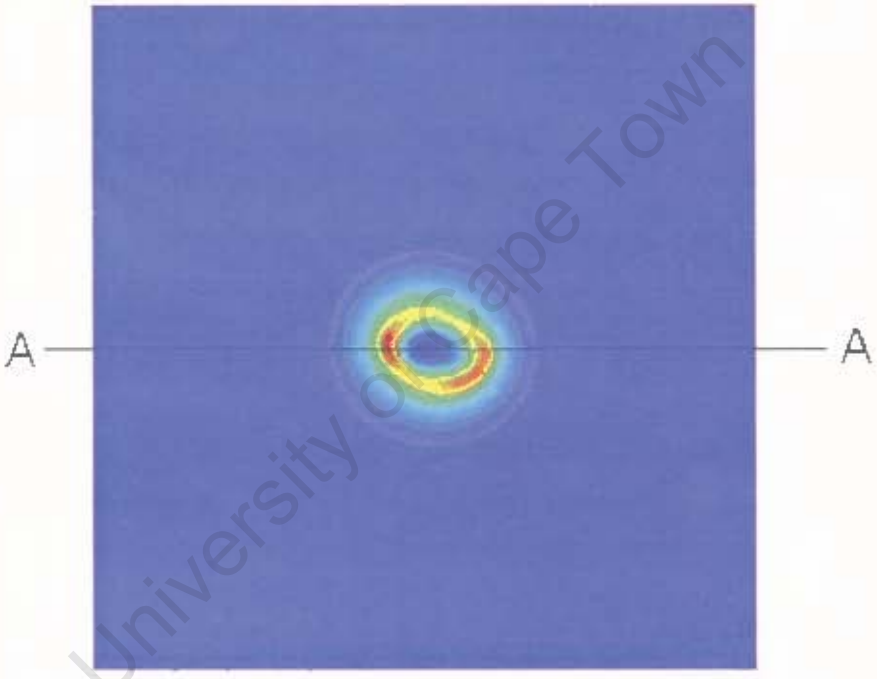

Figure 6-4: Geometrical position of values for plots

In the plot below the actual geometrical position of the point of first ignition can be determined.

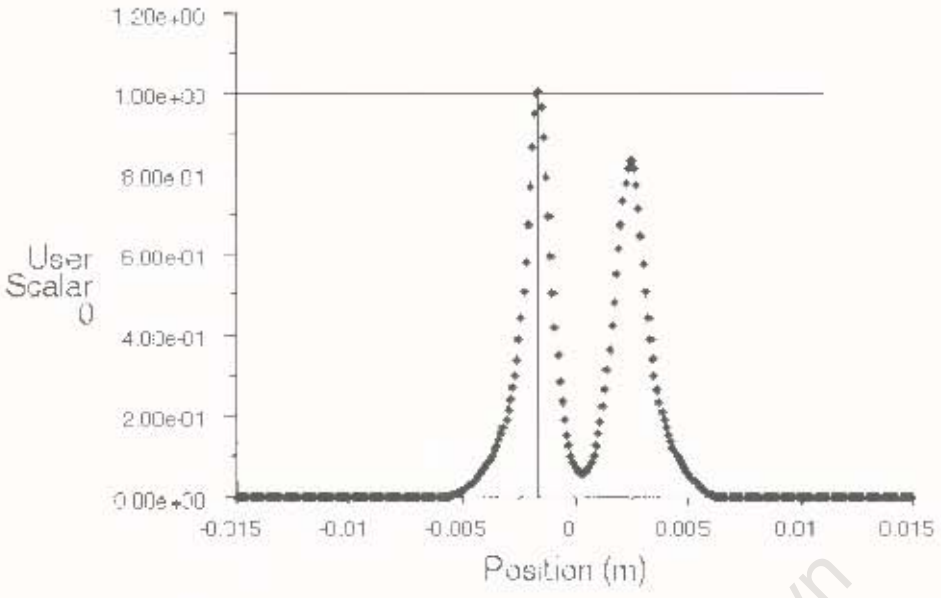

Figure 6-5: Ignition delay plot through the centre of the simulation domain

The point of first autoignition is read of as 1.68 mm from the centre of the domain. this can then be used to determine the temperature and phi at the point of autoignition in the following figures.

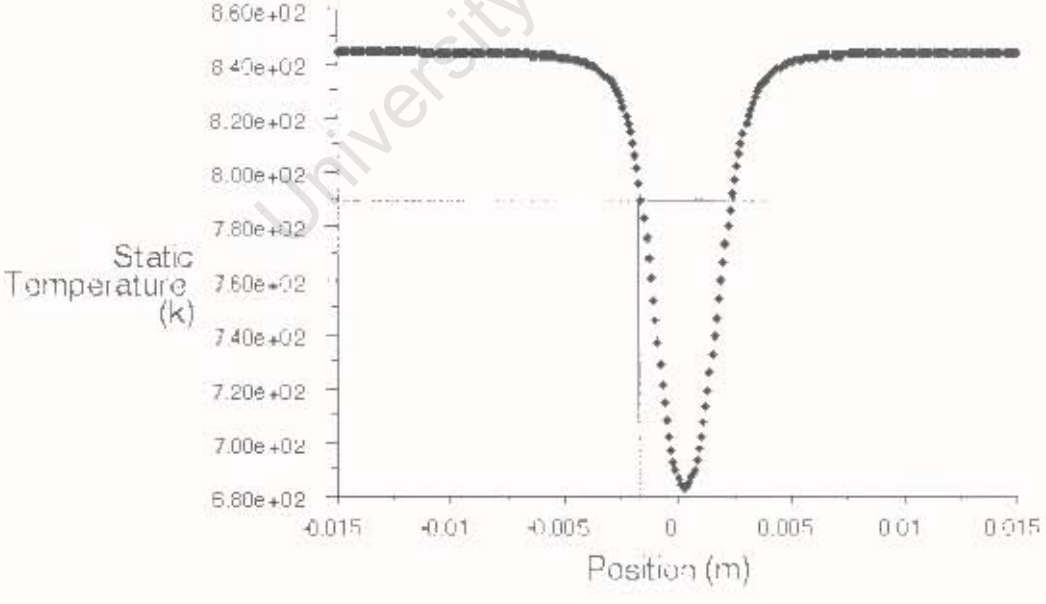

Figure 6-6: Temperature plot through the centre of the simulation domain

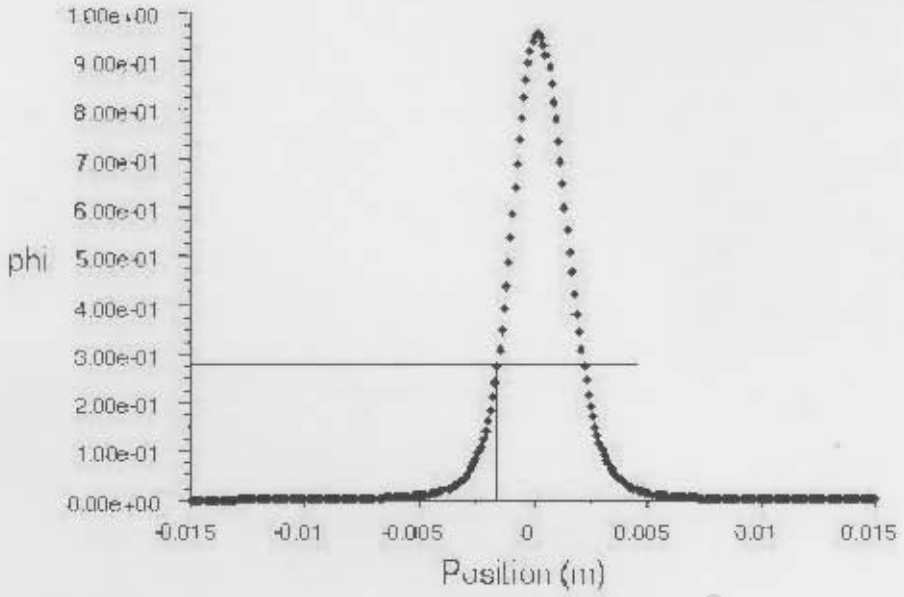

Figure 6-7: Phi plot through the centre of the simulation domain

The temperature determined at the point of ignition was 789.6K and the phi value was 0.26.

The ignition delay calculated in the above simulation was 2.95ms which compares favourably to the experimental result of 2.41ms when considering the simplifications involved. The important simplifications to highlight, is the steady air, the lack of injection velocity and using a "single type" drop, as opposed to using the full injection as in the CFR engine. This model does show the definite possibly of creating a simple model of the CFR engine without requiring the entire combustion chamber and the associated moving meshes. Thus predicting fuel responses to the cetane test, in a simple computationally efficient simulation, becomes a possibility.

In chapter 4-3 a simple homogeneous simulation was created using the same starting conditions as the simulation above, temperature of 844K and a pressure of 32 bar. The fuel/air ratio was defined as 0.83. The ignition delay calculated for the homogeneous model was 7.73ms and if the fuel/air ratio is reduced the ignition delay increases. Thus to resolve a homogeneous model to obtain the correct ignition delay the temperature and the fuel/air ratio have to be calculated (guessed). Globally reducing the temperature affects any further heat transfer models and the temperature would be an average of the temperature experienced by the fuel. In order to create accurate detailed models for diesel engines the homogeneous model is therefore inadequate.

7 Conclusions and Recommendations

From the literature review it was noted that the Arrhenius function which was proposed by Douaud (1978) was an oversimplified model of the fuel's ignition delay. Whilst this simplification was a tolerable representation of fuels having octane values between eighty and a hundred, and specifically in the CFR engine, it was clear that the Douaud (1978) formulation was less appropriate in a more general combustion application.

It was also noted that the Douaud (1978) formulation was specified as one of the autoignition models that was provided in Fluent®. In view of the limited range of applicability of the Douaud (1978) function, the Fluent® user was also provided with two other autoignition models. However, all three of these models were based on the same basic Arrhenius form with different correction factors. It is left up to the Fluent® user to select the most appropriate model for his/her simulation. As has been shown in this report, the Arrhenius function is unable to reflect the auto-ignition character of most hydrocarbon fuels over a wide range of pressures and temperatures. Only the Modified Arrhenius expression proposed for this work is appropriate for all engines because of its intrinsic capability to reflect the ignition delay of the fuel under all the operating normal regimes that would be encountered in real combustion conditions. It is a self-evident requirement of the more sophisticated ignition delay model that one has to have access to the fuel data in order to encode this information into the model.

A continuous flow reactor simulation was used for the project validation and this provided some useful insight into the importance of the in-homogeneities of flows. When the methane gas was injected as the simulated fuel, the variation in temperature and air-fuel ratios extended the predicted time to autoignition and the resulting calculated ignition delays were found to match the experimental data better than those calculated by means of the simple homogeneous models. The importance of having representative data for the fuel model also became clear when the data from the Chemkin® simulation (based on extensive experimental data) indicated ignition delay times that were very close to that of the experiment, while the fuel model based on data from Lefebvre (1986) (limited set of data from one experimental setup) provided poor a correlation.

When modelling a single fuel droplet, the importance of the temperature profile in the vicinity of the evaporating particle was clearly illustrated, as well as the variation in the local air/fuel

ratio. These effects were shown to play off against each other - the centre of the droplet being coldest and hence yielding a longer ignition delay while, at the same time, the high equivalence ratio near the centre had the effect of shortening the ignition delay. Using a simple, two dimensional model and using n-Heptane as the fuel, a realistic prediction of the overall ignition delay was obtained. More importantly, the critical zone of the initial auto-ignition was identified. These simulations show how this model can be used in an environment that exhibits both gradients of temperature and equivalence ratio. It also shows the importance of including such in-homogeneities when creating engine models that include fuel injection. This approach can easily be extended into any type of combustion simulation involving fuel droplets where local temperature and equivalence ratios have a controlling effect on the ignition.

The recommendations for future work are extensive and are dependent on the areas of research which are of interest. The developed CFD code would be equally appropriate and useful for an HCCI engine simulation or a direct injection petrol engine, although these applications would benefit from the inclusion of a heat release model, which is a recommended next step. The heat release model would involve monitoring the ignition delay integral and releasing the combustion energy for each cell when the integral attains the critical value of unity. In this way it would be possible to simulate the rate of pressure rise and the risk of knock. This would have particular relevance in HCCI engine modelling as well as simulating octane rating. Diesel engines could also be modelled with the aid of the droplet simulation which is an area of specific interest to Sasol. The simulation could be applied to the cetane engine and the IQT (Ignition Quality Tester), both of which are used for the certification of diesel fuels.

8 References

Advanced Engine Technology [2002], AET Procedures Manual: IQT™ for Diesel Fuel Cetane Number Evaluation, Canada

Blumenthal, R., et al, [1996], Gas Dynamic Features of Self Ignition of Non Diluted Fuel/Air Mixtures at High Pressure, Combustion Science and Technology, 113-114, 137-166

Brown, C.J., and Thomas,G.O.,[1999], Experimental Studies of Shock-Induced Ignition and Transition to Detonation in Ethylene and Propane Mixtures, Combustion and Flame, 117:861-870

Chemkin® v3.7, [1997], Reaction Design [Computer software]

Clackson, J., et al, [2001], Temperature Fields During the Development of Combustion in a Rapid Compression Machine, Combustion and Flame, 125: 1162 – 1172

Curran, H.J.,Gaffuri, P., Pitz, W.J., Westbrook, C.K., [1998], Cmprehensive Modeling Study of n-Heptane Oxidation, Combustion and Flame, 114: 149 – 177

Douaud, A.M., Eyzat, P., [1978] Four Number Method for Predicting the Anti-Knock Behaviour of Fuels and Engines, SAE 780080

Douaud, A., [1981], Modelling the Knocking Phenomenon in Engine, Int. Symp. On Knocking in Combustion Engines, Wolfsburg

Fish, A., Read, I.A., Affleck, W.S., Haskell, W.N., [1969], The Controlling Role of cool Flames in Two-Stage Ignition, combustion and Flame, Vol 13, pg 39.

Fluent®, [1998], Fluent Inc Release: 6.1.22. [Computer software]

Fluent®, [2005], Fluent Inc Release: 6.2.16. [Computer software]

Golovichev, http://www.tfd.chalmers~valeri/MECH.html [Website]

Hardenburg, H.O., Hase, F.W., [1979], An Empirical Formula for Computing the Pressure Rise Delay of a Fuel from its Cetane Number and from the Relevant Parameters of Direct Injection Diesel Engines, SAE 790493

Held, T. J., Marchese, A.J. and Dryer, F.L. [1997], A semi-empirical reaction mechanism for n-heptane oxidation and pyrolysis, Combustion Science and technology, 123, 107-146

Heywood, John B. 1988. *Internal Combustion Engine Fundamentals.* McGraw-Hill: Singapore.

Hidaka, Y., et al, [1999] Shock-Tube and Modeling Study of Methane Pyrolysis and Oxidation, Combustion and Flame, 118:340-358

Hoskin, D., et al [1992],Ignition delay performance versus combustion of model fuels, SAE 920109

K.J. Hughes, T. Turányi, A. Clague, M.J.Pilling, [2001], Development and testing of a comprehensive chemical mechanism for the oxidation of methane, Int.J.Chem.Kinet., 33: 513-538

Lee, D. and Hochgreb, S. [1998], Rapid Compression Machines; Heat Transfer and Suppression of Corner Vortex, Combustion and Flame, 114:531 – 545

Lefebvre, A.H., Freeman, W. and Cowell, L., 1986, "Spontaneous Ignition Delay Charactoristics of Hydrocarbon Fuel/Air Mixtures", NASA Contractor Report 175064.

Livengood, J.C., Wu, P.C., [1955], Correlation of Autoignition Phenomena in Internal Combustion Engines and Rapid Compression Machines, 5th Symp (Intl) on combustion, p347-356

Ribaucour, M. et al. [2000], Ignition of Isomers of Pentane: An Experimental and Kinetic Modeling Study, Proceedings of the Combustion Institute, Volume 28: 1671 – 1678

Rose, J.W., Cooper, J.R., [1977], Technical Data on Fuel, p281, England, The British National Committee World Energy Conference

Samuelsen, S., et al, 2003, Correlation of Ignition Delay with Fuel Composition and State for Application to Gas Turbine Combustion, Final Report, Contract No. 00-01-SR084Cs, Clemson University Research Foundation.

Siebers, D.L., [1985], Ignition delay characteristics of alternative diesel fuels: implications on cetane number, SAE 852102

Standards South Africa, [2003], SANS 342:2003, SANS Automotive Diesel Fuel

Versteeg, H.K. et al. 1995. *Introduction to Computational Fluid Dynamics: the finite volume method.* New York: Longman Scientific & Technical Wiley

Wang S., et al. [1990], A Flow reactor study of Neopentane Oxidation at 8 atmospheres: experiments and modelling, Combustion and Flame, 118, 415-430

Yates, A.D.B., Swarts, A., Viljoen, C.L., [2005], Correlating Auto-Ignition Delays and Knock-Limited Spark-Advance Data for Different Types of Fuel, SAE 2005-01-2083

Yates, A.D.B., Viljoen, C.L., Swarts, A., [2004], Understanding the Relation between Cetane Number and Combustion Bomb Ignition Delay Measurements, SAE 2004-01-2017

A Appendix

A.1 Tutorial for setting up ignition delay integral

A.1.1 Introduction

This is a simple tutorial indicating the method of including the Arrhenius ignition delay integral. This tutorial does not include an example of mesh development or an injection, since the integral can be used with a multitude of meshes depending on what is being modeled.

A.1.2 Creating the C code

The code shown below is as it is seen in Visual Basic, it can also be created in notepad, but the computer on which fluent will be run needs to have a C code compiler installed.

The code has been given in 2 parts, this is for the sake of simplicity; in the code it will appear as one full text. The first section is to specify all the constants that will be used.

```
#include "udf.h"

/* arrhenius values*/
/* fill in the values of a, n and b according to the values of the specific fuel being used*/

#define A1 1.815e 9
#define n1 1.158
#define B1 11880
#define A2 0.1836
#define n2 1.043
#define B2 2609
#define A3 0.000000E11
#define n3 -0.927
#define B3 16441
#define k -0.7

/* air fuel ratio  data, the stoichiometric value*/
/* for n-heptane stoich = 0.065907 */

#define stoich 0.065907
```

Figure A-1: Variables in the C code

Describing the above code line by line follows:

The "include "udf.h"" includes a library created by fluent for the use of all UDFs, this gives the definitions and associated code to specific terms in the UDF (in the case of this UDF, it will be a "define_source", which will be discussed later).

The next lines are comment lines, comment lines are started with /* and are ended with a */

The lines that follow are to define names of variables and the value. The A, n and B are the constants for the arrhenius function and the k is the exponent for the air/fuel ratio. Finally the last variable is the stoichiometric phi value for the fuel in question.

It is important to change these values for the specific fuel that is being modeled, in this case the values are of n-heptane.

The following part of the code is the calculation and the commands for fluent to input it as a source term

```
DEFINE_SOURCE(ignitionintegral,c,t,dS,eqn)
{
    real p_operating = RP_Get_Real ("operating-pressure");

    real dp = C_P(c,t) + p_operating;


    if (C_YI(c,t,4) <= 0.05)
    {
        real source = 0;
        dS[eqn] = 0.0;

        return source;
    }

    else
    {

    real phi = ((C_YI(c,t,0))/(C_YI(c,t,4)))/stoich;

    real Arr1 = A1 * pow((dp/100000),n1) * exp(B1/C_T(c,t));
    real Arr2 = A2 * pow((dp/100000),n2) * exp(B2/C_T(c,t));
    real Arr3 = A3 * pow((dp/100000),n3) * exp(B3/C_T(c,t));


    real source =  (1000 * C_R(c,t)) / (1/((1/(Arr1+Arr2))+(1/Arr3))*pow(phi,k));
    dS[eqn] = 0.0;
    return source;
    }
}
```

Figure A-2: Calculation in C code

The first line is a line defined by fluent to specify that the variable returned is a source term. the definitions on the variables and how they are used in this tutorial are below:

DEFINE_SOURCE(name, c, t, dS, eqn)

Ignitionintegral	refers to the name of the UDF, as it will be seen in fluent
c	Defines the cell to which the source term will be applied, within the specified thread.
t	Pointer to the cell thread, a common group of cells

dS	The derivative of the source term can be included if the source term includes the dependant variable of the transport equation
eqn	Equation type number

In this case these variables can be ignored because they are specified when the UDF is hooked. When dS is specified as 0.0 the source term is calculated explicitly and therefore the equation type is not required.

The next line creates a new variable which is assigned the value of the operating pressure which has been defined in fluent.
Then this value is added to the static pressure calculated in Fluent. This is to give the absolute pressure, which is required for the Arrhenius function.

The "if" statements that follow defines the situation if the cell contains pure air. This is required because the fuel/air ratio becomes undefined if the mass fraction becomes zero. In this case the value returned is zero.

The else statement that follows is the calculation that occurs if the arrhenius function can be calculated. The first line is the calculation of the phi value, then the three different arrhenius functions are calculated, as can be seen in the equation the density is divided by one hundred thousand. This is to change from pressure in pascals to pressure in bar as required in the arrhenius function. These functions are added in a way that was discussed in the report. As can be seen the function is multiplied by a thousand and density: the thousand is to change the calculation from milliseconds to seconds and the density is to cancel out another density term that is included when the transport equation is solved. This value is then multiplied by the fuel/air ratio.
Then the value is returned to fluent.

The definitions for variables in fluent that are required are as follows:

C_P(c,t)	Pressure, gauge pressure
C_T(c,t)	Temperature
C_R(c,t)	Density
C_YI(c,t,0)	Mass fraction of fuel
C_YI(c,t,4)	Mass fraction of nitrogen (air)

The definitions for the mass fractions need to be carefully checked, depending on the model being run. The number defines which mass fraction is returned. There is no explicit list of which component is given which number. It is found the simplest method to do this is to select the contour plot menu, select species and the species appear in the order of the numbering. Note that the numbering starts at zero.

A.1.3 Compiling the UDF

Select **Define** → Select **User-Defined** → Select **Functions** → Select **Compiled**

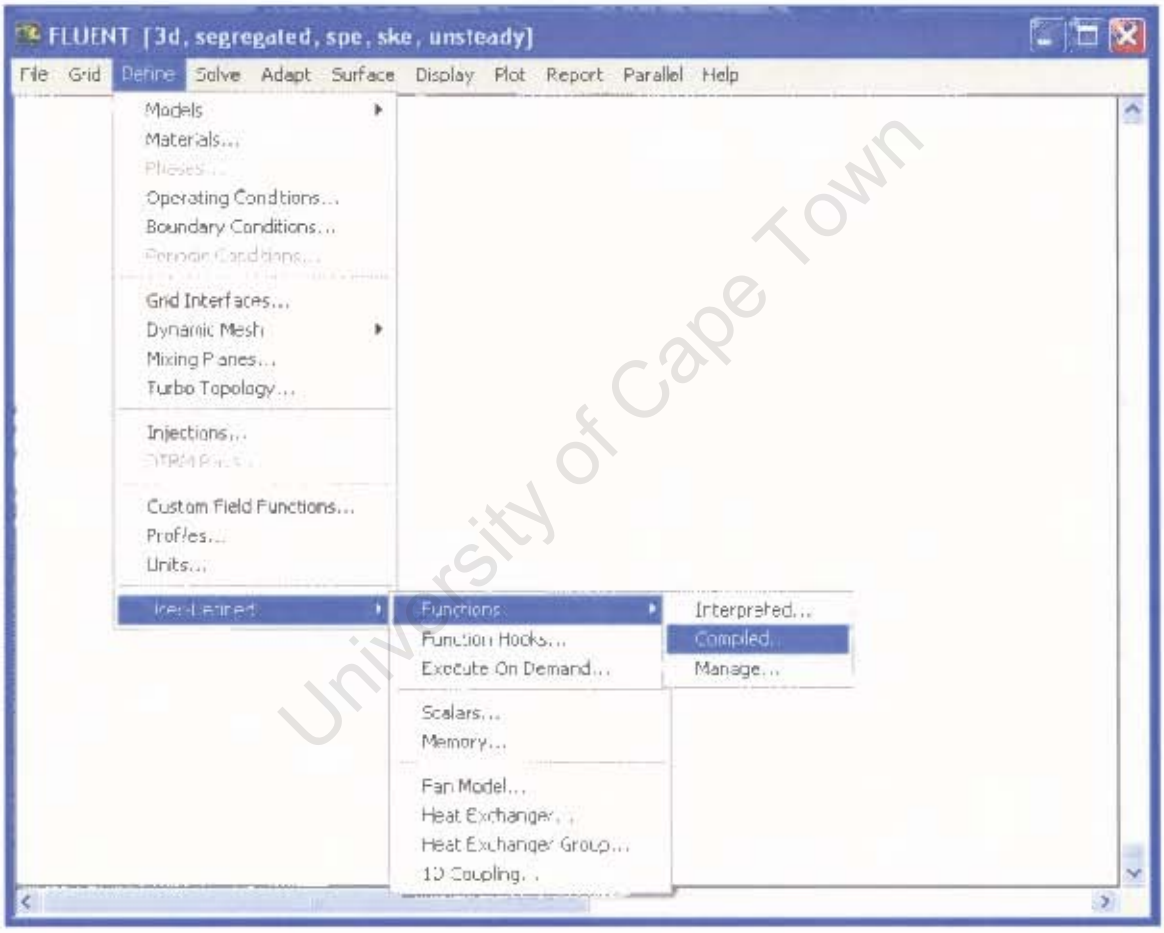

Figure A-3: Compile UDF menu

This opens the window that is shown below.

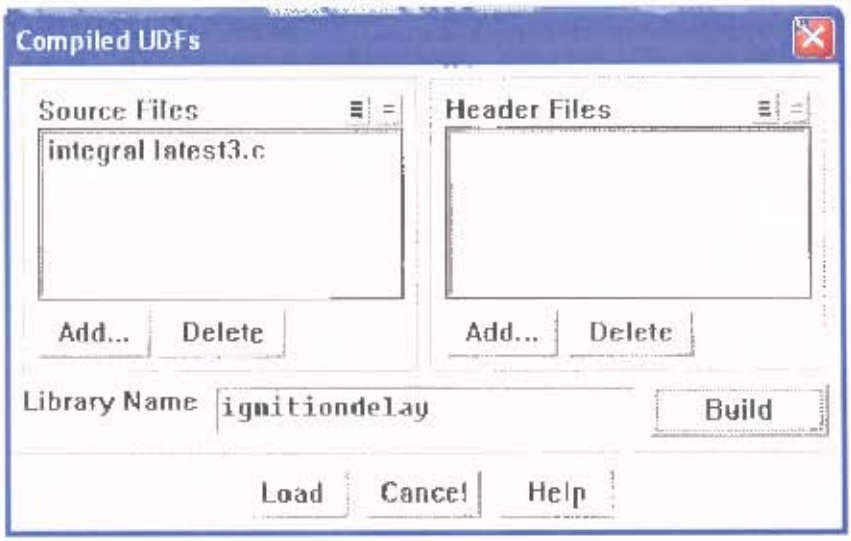

Figure A-4: Selecting the UDF

Add the UDF that you have created, the UDF must be in the same folder as the case in which you wish to use the UDF. Then in the library name window, type in the name that you want to call the UDF, there can be more than one UDF in a library.

Then select build, text will appear in the fluent page, the text can be seen below. If the UDF compiles select load, then the UDF is ready to be used. If there is an error in the code fluent will return an error and specify the line that the error occurs. The user must then return to the code and correct the error, then build the UDF again.

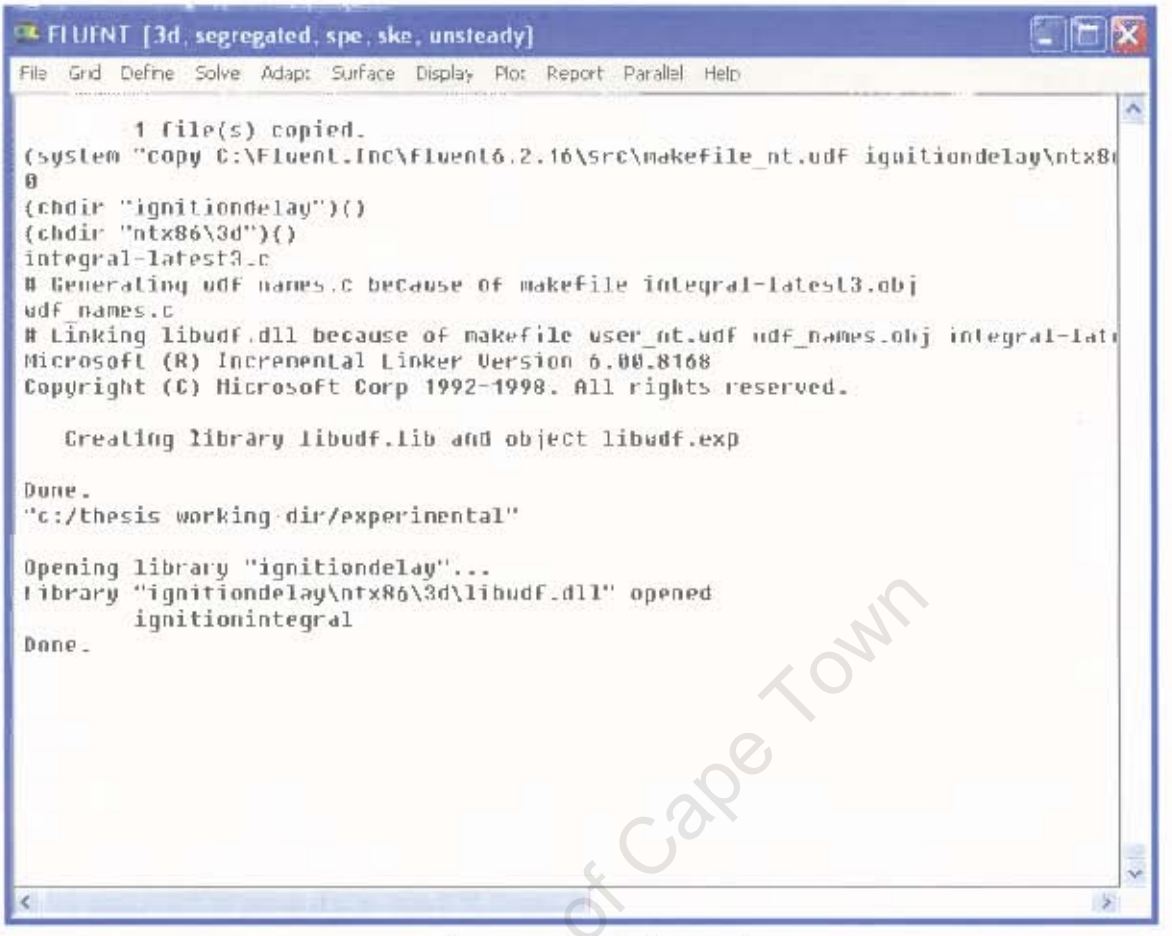

Figure A-5: Building the UDF

A.1.4 Creating the UDS

Until now, only the UDF has been defined, the Scalar transport has not been discussed; the following is the method for creating the UDS and specifying the variables to define the operation of the UDS.

First a UDS must be created, this is done as follows:

Select **Define** → Select **User-Defined** → Select **Scalars**

This opens the following window:

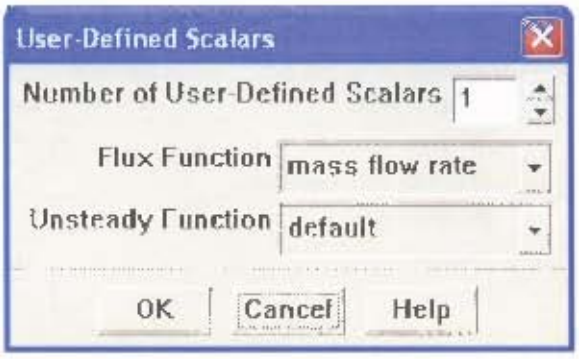

Figure A-6: Creating a UDS

For this tutorial only one UDS is being created, but more can be created, in the case of comparing different arrhenius functions etc.

In the number of User-Defined Scalars window select **1**

In the Flux Function window select **mass flow rate**

In the Unsteady function window select **default**

At this stage there is one UDS and there convection term is defined to be the same as the mass transfer, the unsteady term is defined as default. which specifies that this is a time dependent function.

Below is the method for defining the diffusion term:

Select **Define** → Select **Materials**

The following window opens:

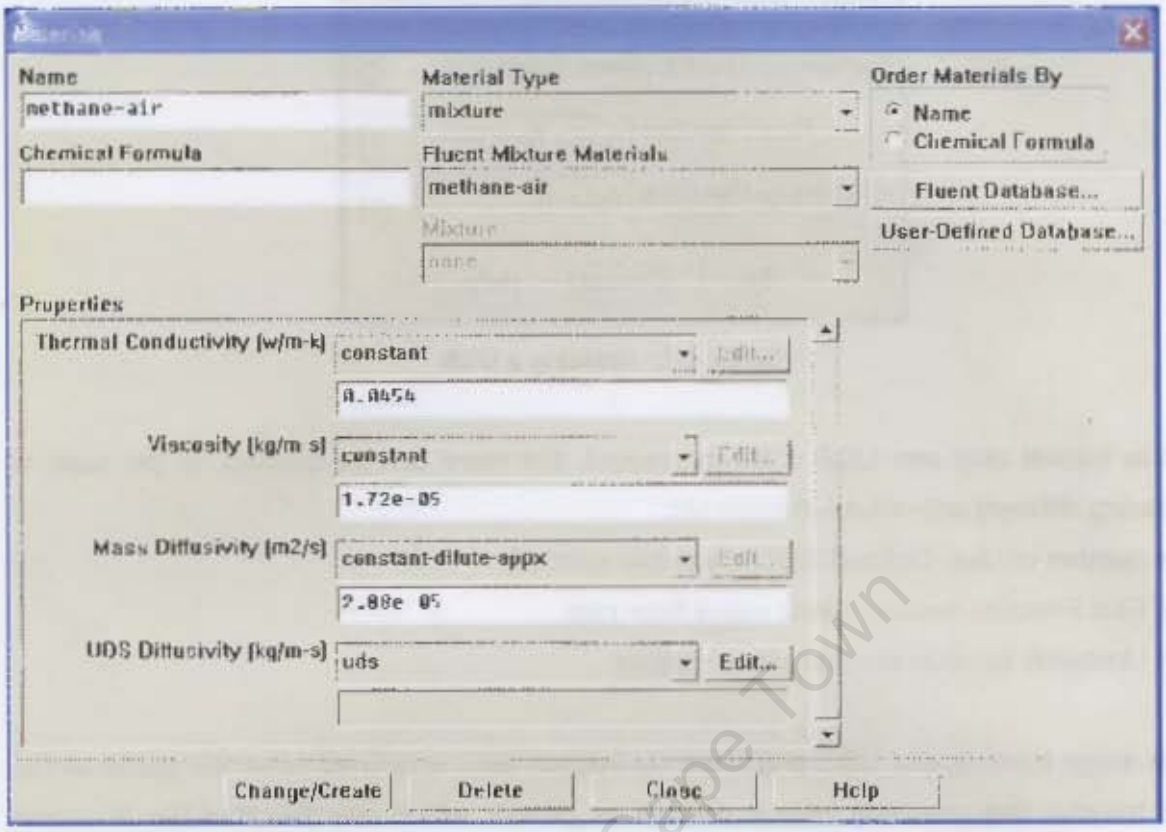

Figure A-7: Define the diffusion term of the UDS

Select **Edit** on the UDS Diffusivity window. the following window opens:

Figure A-8: Specifying the value for diffusion

In this case there is only one UDS. in the User-Defined Scalar Di window select **uds-0,** in the Coefficient window select **constant** and for the value select **0.**
This must be done for the mixture properties and the fluid properties.

The diffusivity has now been defined for the UDS it has been defined as zero so that the Scalar only follows the mass transfer this is since it is only a place holder for the integral which must follow the mixture exactly.

A.1.5 Hooking the UDS

The UDS has now been defined it terms of convection etc but still does not include the source term which is the integral, the following is to connect the UDF to the UDS.

Select **Define** → Select **Boundary Conditions**

The following window opens:

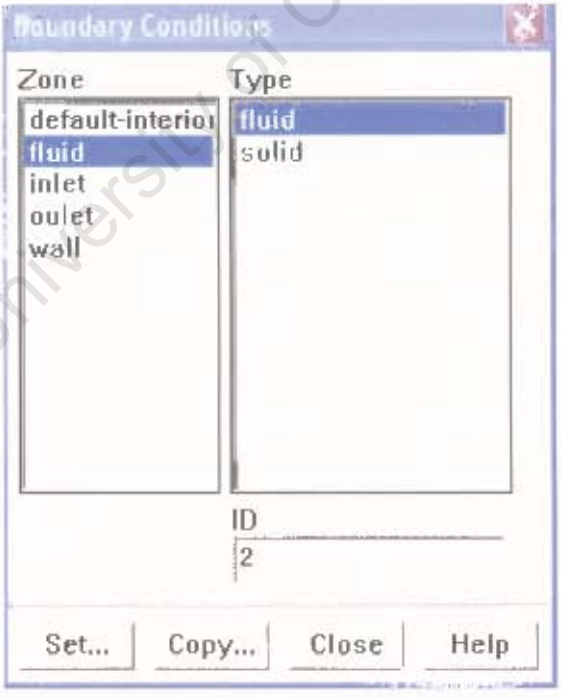

Figure A-9: Boundary Condition Menu

Select zone **fluid** (or any zones specified as fluid if there are more than one or they have been renamed).

The following window opens:

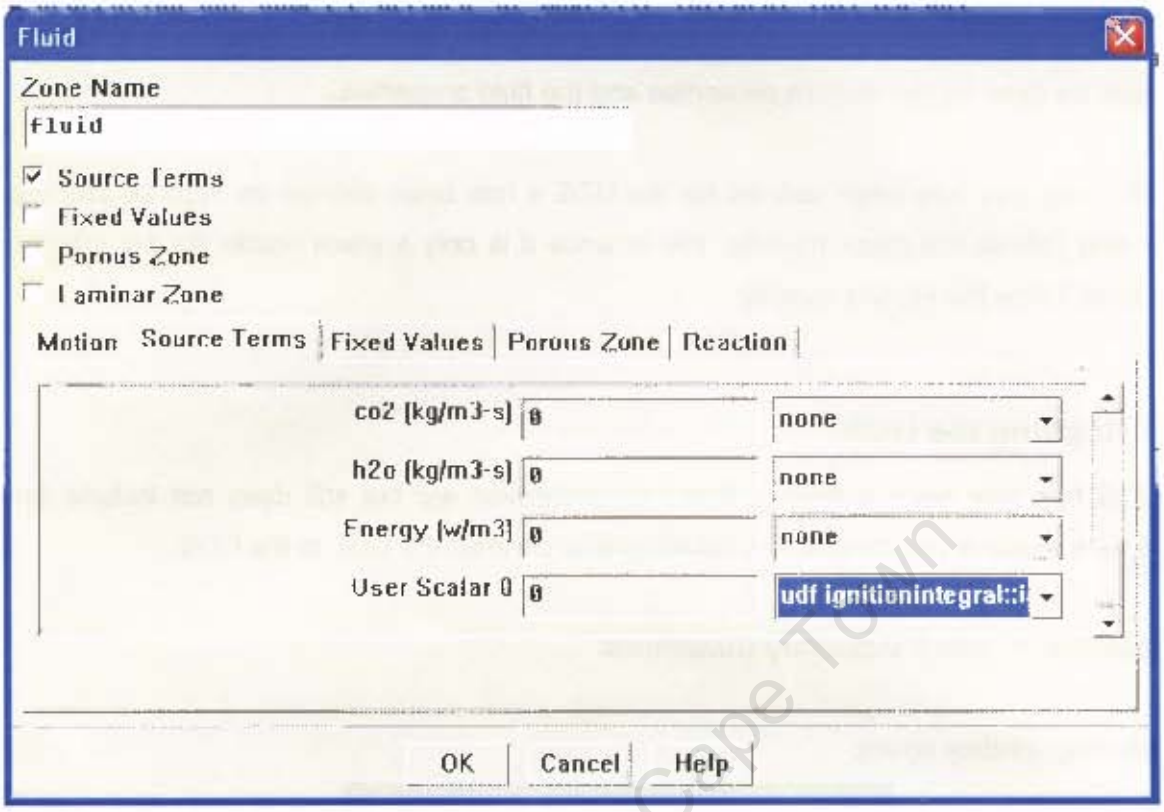

Figure A-10: Hooking the UDF

Check the Source Terms box

Then select the **Source Terms** tab

Then in the drop down menu for User Scalar 0 select the **udf option**, the name of the udf will be the library name, colon, colon, the name specified in the define_source line of the c code. So for this example: **udf ignitionintegral::ignitionintegral**

The UDS is now complete and can be viewed in the contours plot (under user defined scalars). it can also be seen in xy plots or any or the post processing tools avalible.