American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-7, Issue-3, pp-173-199
www.ajer.org
Research Paper Open Access

The Fundamentals of Theoretical Modelling of Gas Explosion– A Review

Joshua Williams[b], Jafar Zanganeh[a,]*, Sazal Kundu[a], Bishnu Lamichhane[b],Behdad Moghtaderi[a]

[a]Priority Research Centre for Frontier Energy Technologies & Utilisation, Discipline of Chemical Engineering, School of Engineering, Faculty of Engineering & Built Environment, University of Newcastle, Callaghan, NSW 2308, Australia

[b]School of Mathematics and Physical Sciences, University of Newcastle, Callaghan NSW 2308, Australia

** Corresponding authorJafar Zanganeh*

Jafar.zanganeh@newcastle.edu.au

ABSTRACT: Thisarticle provides an understanding on the theoretical modelling of gas explosions. Several commercial software are available; however, it is important to know how they work and their limitations. A theoretical modelling of gas explosion is the integration of several sub-models including chemical reaction, fluid dynamics and thermodynamic models. Once these sub-models are known, it is important to determine how accurate is sufficiently accurate. This understanding determines the level of computational expense required to conduct the targeted theoretical modelling of gas explosions. This article provides the readers an understanding on the requirements in the development of computational modelling of gas explosion.

KEYWORDS: Gas explosion, Methane explosion, Natural gas explosion, LPG explosion, Mine explosion, Computational modelling

Date of Submission: 27-02-2018 Date of acceptance: 14-03-2018

Contents

Nomenclature

Symbol	Units	VariableName
a_i^n	m/s	The speed of sound
A	m^2	Area
A	1/s	Pre-exponential factor
CFL	(-)	Courant-Friedrichs-Lewy Number
c	m/s	Sound speed
D	m^2/s	Mass diffusivity
e	J	Internal energy
E	J/m^3	Energy of Control Volume
E_A	J/mol	Activation Energy
E_k	J/m^3	Kinetic Energy
E_{Th}	J/m^3	Thermal Energy
$\mathbf{F}$	N/A	X-Flux
F_T	W/m^2	Heat Transfer
$\boldsymbol{g}$	m/s^2	Gravitational Acceleration Vector
$\mathbf{G}$	N/A	Y-Flux
$\mathbf{H}$	N/A	Z-Flux
H^i	N/A	Higher Order scheme
$\mathbf{I}$	(-)	Identity matrix
K	W/m/K	Thermal Conductivity Coefficient
M	kg/kmol	Molecular weight
M	N/A	Monotone Scheme
$\boldsymbol{n}$	(-)	Outward Normal Vector
P	Pa	Pressure
q	J	Total chemical energy release
Q	J/kg	Enthalpy of Formation
R	J/mol/K	Ideal Gas Coefficient
$\mathbf{S}$	N/A	Source Component
$S^n{}_{max}$	m/s	Approximate Wave Velocity
T	S	Time
T	K	Temperature
T^i	N/A	TVD scheme
U	m/s	X-velocity
u_i	N/A	Average Local State Vector
u_i^n	N/A	Average Local State Vector
$\mathbf{U}$	N/A	State Vector
V	m/s	Y-velocity
V	m^3	Volume
$\boldsymbol{V}$	m/s	Velocity
W	m/s	Z-velocity
$\mathbf{Y}$	(-)	Mass fraction
γ	(-)	Adiabatic Index
ε	(-)	Refinement Indicator
θ	(-)	Local Equivalence Ratio
$\varkappa$	m^2/s	Thermal Diffusivity
$\boldsymbol{\nu}$	Pa s	kinematic shear viscosity
$\boldsymbol{\nu_F}$	(-)	Stoichiometric coefficient of fuel
$\boldsymbol{\nu_O}$	(-)	Stoichiometric coefficient of oxidizer
P	kg/m^3	Density
$\boldsymbol{\tau}$	Pa	Viscous Stress Tensor
φ	N/A	Flux Limiter

Symbol	Units	VariableName
$\hat{\varphi}$	N/A	Localised Flux Limiter
ω	$kg/s/m^3$	Rate of reaction
$\boldsymbol{\Omega}$	(-)	Vorticity
B	N/A	Adiabatic
L	N/A	Laminar
L	N/A	Left
LF	N/A	Lax-Friedrich
LOCAL	N/A	Local
FLIC	N/A	FLIC-approach
FORCE	N/A	FORCE-approach
GLOBAL	N/A	Global
R	N/A	Right
RI	N/A	Richtmyer
0	N/A	Initial
$\dagger$	(-)	Transpose operator
Δ	(-)	Difference Identifier

I. INTRODUCTION

In human history, gas explosionshave taken the lives of a massive number of people and destroyed many infrastructures[1-8]. These destructions include damages in diverse localities, including laboratories and also in industries and mines. Natural gas and liquid petroleum gas (LPG) are commonly used in many parts of the world in household cooking and room heating; however, the application of these flammable gases in household requirementscause explosion hazards, which have led to injuring and killing people and destroying houses[7-10]. In 2016, one house was levelled and 15 people were injured in a natural gas explosion originated from an underground gas-line leak in Stafford Township in New Jersey, USA[9].

Flammable gases are commonly used in research activities and stringent policies are usually followed in their usage; however, a2016hydrogen explosion in a laboratory of the University of Hawaii severely injured a researcher [11].

The death toll due to mine explosion can be more than a thousand where explosion usually initiate due to the presence of methane gas. In 1942, 1,549 people were killed in a coal mine explosion at the Benxihu Colliery in China and this was the deadliest coal mine explosion in human history [2]. The highest killing due to coal mine explosion in Australia was occurred at the Mount Kembla Mine in 1902 and the accident took lives of 96 people [12].Methane can also be present in gold mines and in 1986, an enormous explosion occurred in the Kinross gold mine of South Africa and 177 people lost their lives [13]. The destructive nature of these accidents led to initiate explosion research.

With the aim of addressing explosion hazards, for more than a century, experimental investigations have been carried out employing various flammable gases including methane and hydrogen[14-23]. Those investigations revealed the characteristics of explosionphenomena. Some of those characteristics are presented here.Explosion does not occur at every concentration of the flammable gas. For instance, methane in airis explosive within a concentration range between 4.6±0.3% and 15.8±0.4% [14, 24]. The range of concentrations at which explosion can occur is known as the flammability range of a gas.In addition, explosion phenomena can be characterised as deflagration, detonation and transition from deflagration to detonation (DDT). In a deflagration type of explosion, the flame speed is lower than the speed of sound, where the speed of sound is estimated at the temperature of combustion products[25-27]. When the combustion wave propagates much faster than the speed of sound and the pressure rise can be estimated by Chapman-Jouguet correlations, the explosion phenomena is known as detonation[28-30]. DDT is described as a sudden transition of a low speed deflagrated flame to detonation or quasi detonation stage of explosion[31-33]. Detonation is undoubtedly much more detrimental than deflagration[34-36].

While experimental investigations provided tremendous information on explosion hazards, there are still a lot of unknowns in explosion phenomena. Experimental setups are usually constructed with regular shaped components such as spheres or tubes. This compromises to replicate any real world explosion phenomena, particularly that occur in coal or gold mines. The geometry of explosion an initiation space in coal or gold mines is highly irregular. Therefore, mathematical modelling verified with results from laboratory scale setup can be used in mapping real world explosion phenomena[37-41]. Besides the simulation of explosion in an irregular geometry, mathematical modelling can aid in exploring science of explosion phenomena.

Typically, commercial software such as Fluent and FLACS were applied to investigate explosion phenomena and found to reproduce the results well for deflagrations[42-47]. However, these numerical models were unable to completely capture the experimental results for DDT and detonation because of theirinsufficiently developed algorithms[48]. It is most likely because both models use a $k - \varepsilon$ turbulence model and a β flame model, there is ultimately an accuracy limit due to the smearing of turbulence effects[48]. This smearing acts to reduce the gradients in most variables such as temperature, pressure, reactants and products concentrations and velocity, and therefore, impacts on reproducing the reproducibility of explosion results of DDT and detonation[49]. Therefore, understanding the fundamentals of these numerical models is important, as this can help in better use of the commercial software and also assist to develop an algorithm and a computer code to suit the purpose.

This article describes the building blocks of theoretical modelling of gaseous explosions. It elaborates the various components of a theoretical model, the choices a researcher can make during the modelling and their implication on the result.

II. GOVERNING EQUATIONS AND NUMERICAL PHILOSOPHY

Applying a mass balance to a given three dimensional control volume produces the following integral equation (1) where ρ is the density of a fluid, V is the control volume, A is the surface area of the control volume, t is the time,$\boldsymbol{V}$ is the velocity vector and $\boldsymbol{n}$is the outward pointing unit vector normal to the surface A [50].

$$\frac{\partial \iiint \rho dV}{\partial t} + \iint \boldsymbol{n} \cdot (\rho \boldsymbol{V}) dA = 0 \qquad \textbf{(1)}$$

The first term is representative of how the total mass is changing, and the second term describes how the mass is leaving and entering the system as a flux across the control volume.

Similarly, a momentum balance can give the following integral equation, (2); where, importantly, each term in the equation is a vector and therefore is more aptly described as a set of scalar equations (the number of equations being the number of important spatial dimensions). Note that P symbolises the pressure of the system, $\boldsymbol{\tau}$is the viscous stress tensor given by equation (3), $\boldsymbol{g}$is the gravity vector, ν is the viscous stress tensor,**I** is the unit tensor and † is the transpose operator[48, 50].

$$\frac{\partial \iiint \rho \boldsymbol{V} dV}{\partial t} = -\iint [\boldsymbol{V}(\boldsymbol{n} \cdot \rho \boldsymbol{V}) + P\boldsymbol{n} - \boldsymbol{n}.\boldsymbol{\tau}] dA + \iiint \rho \boldsymbol{g} dV \textbf{(2)}$$

$$\boldsymbol{\tau} = \rho \nu \left(\frac{2}{3}(\nabla \cdot \mathbf{V})\mathbf{I} - \nabla\mathbf{V} + (\nabla\mathbf{V})^{\dagger}\right) \qquad \textbf{(3)}$$

An energy balance can also be applied to the control volume that generates the following integral equation (4), where E is the internal energy of the system, k is the thermal conductivity coefficient,T is the temperature vector,q is the heat produced by the reaction and ω is the rate of the reaction[50].

$$\frac{\partial \iiint E dV}{\partial t} = -\iint [\boldsymbol{n} \cdot (E\boldsymbol{V} + P\boldsymbol{V} + k\nabla T) - \boldsymbol{V} \cdot (\mathbf{n} \cdot \boldsymbol{\tau})] dA + \iiint \rho q \omega dV \qquad \textbf{(4)}$$

Lastly, a chemical species balance can be applied which generates the following integral equation, (5).This is a vector equation and is therefore representative of multiple scalar equations. $\boldsymbol{Y}$ is the mass fraction vector (where each element of the vector is representative of the mass fraction of each individual chemical species) and D is the mass diffusion coefficient[48].

$$\frac{\partial \iiint \rho \boldsymbol{Y} dV}{\partial t} = -\iint [\boldsymbol{n} \cdot (\rho \boldsymbol{Y}\boldsymbol{V} + \rho D \nabla \boldsymbol{Y})] dA + \iiint \rho \omega dV \qquad \textbf{(5)}$$

Models, in general, will use thermodynamics to identify or build relationships to evaluate the physical and chemical coefficients such as the mass diffusion coefficient, the temperature and pressure of the system along with other properties that are required for the chemical model or an understanding of the physical system such as the speed of sound and the viscosity of the system[50].

The rate of the reaction and the heat released by the reaction are determined by the reaction model, by use of the fluid dynamics properties, and the thermochemical properties of the fluid.

The rest of the variables, such as density, velocity and energy, along with the consequences of the thermodynamics and the reaction model are calculated using the CFD component of the model. The biggest issue for the CFD section of this model is dealing with discontinuities and turbulence. The turbulence is modelled by use of the viscous stress tensor which allows the second order mixed partial derivatives to affect the state variables.

By constructing a state vector, U, and functions of the state vector, F,G,H,S one can describe equations

(1)to (5), by the use of Green's theorem to form one vector equation as given below.

$$\frac{\partial \iiint \boldsymbol{U} dV}{\partial t} + \oiiint \frac{dF(\boldsymbol{U})}{dx} + \frac{dG(\boldsymbol{U})}{dy} + \frac{dH(\boldsymbol{U})}{dz} dV = \iiint S(\boldsymbol{U}) dV \qquad \textbf{(6)}$$

In cases where the distributions of these variables are sufficiently smooth these equations can be expressed in a differential form as shown below[48, 51]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0\textbf{(7)}$$
$$\frac{\partial \rho \boldsymbol{V}}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}\boldsymbol{V}) + \nabla P + \nabla . \boldsymbol{\tau} = 0\textbf{(8)}$$
$$\frac{\partial E}{\partial t} + \nabla \cdot \big((E + P)\boldsymbol{V}\big) + \nabla \cdot (\boldsymbol{V} . \boldsymbol{\tau}) + \nabla \cdot (k \nabla T) + \rho q \omega = 0\textbf{(9)}$$
$$\frac{\partial (\rho \boldsymbol{Y})}{\partial t} + \nabla \cdot (\rho \boldsymbol{Y}\boldsymbol{V}) - \nabla \cdot (\rho D \nabla \boldsymbol{Y}) - \rho \omega = 0\textbf{(10)}$$

where $\boldsymbol{\tau}$ is the viscous stress tensor in the vector form and † is the transpose operator.

Again, these four equations can be combined into one state vector equation[52]:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial F(\boldsymbol{U})}{\partial x} + \frac{\partial G(\boldsymbol{U})}{\partial y} + \frac{\partial H(\boldsymbol{U})}{\partial z} = S(\boldsymbol{U}) \qquad \textbf{(11)}$$

One should note that the functions *F*, *G* and *H*, may have inside them another spatial derivative causing secondary derivative terms (due to thermal conduction, diffusion and viscous forces). Considering that this problem has both integral forms and differential forms, from a mathematical perspective, the choice of using a finite differencing method or a finite volume method can appear arbitrary. Below in Figure 1, the two methods are visually compared. The way that thermodynamics is described lends itself towards a finite volume method approach (as thermodynamics is centred on control volumes and surface areas of control volumes)[53]. Further finite volume methods allow for a direct control on boundary values and boundary fluxes – which are the most common description of boundaries.

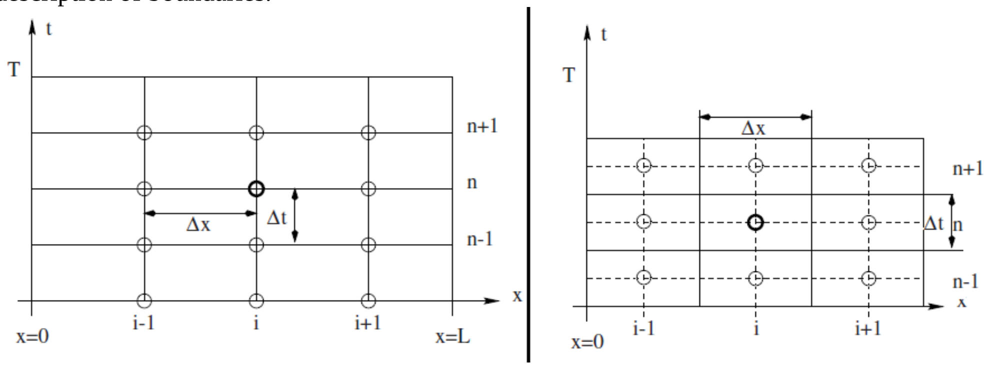


Figure 1: A depiction of the typical manner in which finite differencing methods and finite volume methods are visualised. Specifically, the finite differencing approach (left) looks at particular points (generally visualised as the nodes on a graph) whereas the finite volume method approach (right) looks at the average of a control volume (visualised as the centre of a cell). This figure is augmented from the original produced by[50].

The finite volume method approaches this problem by use of the integral forms of the problem, meaning that it does not require as much smoothness in the spatial distributions (although smoothness in the temporal variable is still required) as the differential form[50]. This is important as combustion can cause incredibly rapid changes in pressure, temperature, velocity, density and the fuel mass fraction which are all components of the state vector. This can generate apparent discontinuities in the spatial distributions of the state functions and therefore removes the smoothness of the distributions causing problems for finite differencing approach. For these reasons, the finite volume paradigm was selected.

By defining an integral average of each control volume we can define the following sequence where u_i^n is the integral average of the ith cell, and ΔV_i is the size of the volume of the i^{th} control volume.

$$u_i = \frac{1}{\Delta V_i} \iiint U dV \qquad \textbf{(12)}$$

From here, equation (6)can be described as below.

$$\frac{\partial u_i}{\partial t} = \frac{1}{\Delta V_i} \oiiint \frac{dF(\boldsymbol{U})}{dx} + \frac{dG(\boldsymbol{U})}{dy} + \frac{dH(\boldsymbol{U})}{dz} dV + \frac{1}{\Delta V_i} \iiint S(\boldsymbol{U}) dV \quad \textbf{(13)}$$

By the use of the integral average, the general problem can be visualised as the following partition of the system (depicted in one dimension for simplicity) as seen below in Figure 2.

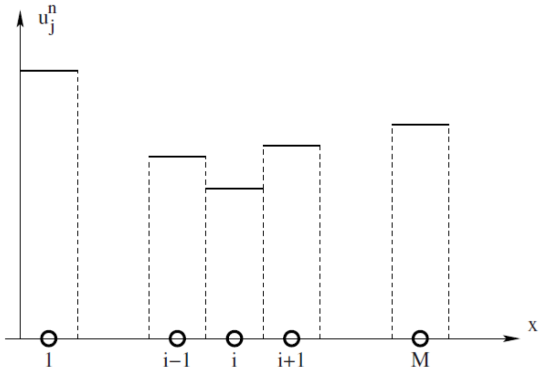


Figure 2: Visualisation of constant cell value equal to the integral average. From this depiction it can be seen that the very manner in which the system is being solved implies discretised data (as opposed to the conventional continuous data produced by many theoretical models). Further, the model should always be processing on discontinuous (or constant) values, meaning that the model must be able to deal with discontinuous (and therefore non-differentiable) approximations of distributions. This figure is taken from[50].

By use of the fractional step method, this high order equation can be split into the following equations, by considering only changes in each spatial variable separately and then solving the equations iteratively. From here the integral term in the first three equations can be evaluated by the fundamental theorem of calculus (by assuming the grid is well built to allow for smoothness inside the control volumes):

$$\frac{\partial u_i}{\partial t} + \frac{1}{\Delta x_i} \int \frac{dF(\boldsymbol{U})}{dx} dx = 0 \quad \textbf{(14)}$$

$$\frac{\partial u_i}{\partial t} + \frac{1}{\Delta y_i} \int \frac{dG(\boldsymbol{U})}{dy} dy = 0 \quad \textbf{(15)}$$

$$\frac{\partial u_i}{\partial t} + \frac{1}{\Delta z_i} \int \frac{dH(\boldsymbol{U})}{dz} dz = 0 \quad \textbf{(16)}$$

$$\frac{\partial u_i}{\partial t} = \frac{1}{\Delta V_i} \iiint S(\boldsymbol{U}) dV \quad \textbf{(17)}$$

Over one time step, a typical solution for equation (13)might be found by solving equation (14), which would then be used as the initial conditions to solve equation (15), the solution of which is then used as the initial conditions for (16)and so forth until the solution of equation (17)is found, finishing the cycle for one time step. It is important to note that this cycle is only illustrative of a typical cycle. In order to eliminate any variable bias and maintain the order of accuracy for whichever numerical scheme is applied to solve these equations, the order in which the spatial variables are solved must be reversed every other time step[54]. Further the source term solution must be split in half and done once at the beginning and once at the end of each loop to recover the order of accuracy in time[55].

Finding how to approximate the functions F, G, H, and S evaluated at the limits of the control volume (approximated by the sequence of u_i^n) and the implementation of this on a computer is discussed in the later part of the article.

III. REACTION MODELLING

The mathematical treatment of the chemical reaction is one of the most difficult components in the theoretical modelling of gas explosion. Nonlinearity causes significant problems to many generalised theorems and techniques in mathematical models. The majority of models of reactions are, at least in some part, described by exponentials due to an Arrhenius relationship[56-58], which can be difficult to linearise. The complexity of the reaction model can differ from using the mass fraction of the products as a reaction-completion indicator describing what percentage of the reaction has been completedto attempting to model intermediary steps of the reaction and how the intermediary radicals influence the reaction rates[59, 60]. In the case where reaction-completion indication is used, the heat of formation is employed to calculate the amount of energy released and is incorporated as potential chemical energy. This model is useful as it removes the nonlinearity in the total system's energy; however, it does still require an exponential term in the modelling of the mass fractions of the reactants and products[59]. Whilst this model has been useful historically for its apparent linearity, it has been criticised for experimental conditions that have a large amount of mass fraction displacement that acts back into the combusted region. This is because if the concentration of the reactants/products is caused by displacement rather than reaction their concentration is no longer indicative of the 'reaction completion.

One of the more common ways to simulate reactions is to model the reaction as if it was a one-step unimolecular first order reaction[48]. This means that the numerical scheme for the reaction term still needs to deal with non-linearity in both the released energy as well as the mass fraction of the chemical species. This model is still linear outside of the thermal effects which does allow for some interesting numerical techniques that are discussed in 3.2. In 3.1, the real chemical reaction is discussed briefly, and importantly, the reaction is neither first-order, one-component nor one-step. This means that models that rely on this simplification require some way to deal with the stark differences in the reality of the system and the simplifications used. Usually, this is done by limiting the versatility of the model, specifying experimental conditions for which the model can be used. Interestingly, Oran et al. introduced a more versatile manner in which this can be extended by using a combination of theoretical and phenomenologically derived relationships between the type of reaction model used and the reacting fluid dynamic results found in combustion experiments[48].

In general, reaction models are made to reflect the reality of the system by improving on the simplifications discussed above, such as increasing the number of steps of the reaction or increasing the number of species in the reaction (both of which requires modelling more chemical species and reactions which causes a larger demand on computing power), or increasing the order of the reaction which increases the complexity of the required numerical schemes. Increasing the complexity of numerical schemes often causes either a decrease in the formal order of accuracy (meaning more cells are required) or an increase in produced rounding errors (meaning a limit to how accurate a scheme can become by reducing the limit to how small the control volumes can become).

3.1. Methane combustion reaction

Whilst the theory presented in this article uses a second order bimolecular reaction approximation, the actual reaction of methane is significantly more convoluted and complex in subtle ways. Methane combustion occurs through at least nineteen intermediary major reactions[61] and due to the ephemeral nature of some of the intermediary reactions, there can be more intermediary reactions. If the intermediary reactions are so rapid that their consequences are difficult to observe experimentally, it is unlikely to affect the model to a large degree – meaning evanescent species have a small effect on the model.

Whilst the exact description of each subsidiary reaction is beyond the scope of this article, a generalised description of the categories of reactions is included. Initially, the reactants decompose into more fundamental chemical species. This decomposition breaks down chemical bonds and is therefore unsurprisingly endothermic. This creates a sort-of thermal threshold (energy required to initialise the endothermic reactions) [62]. After this category of intermediary reactions, there are subsequent reactions that combust the intermediary species into products. These reactions are usually highly exothermic. The over-all chemical reaction is expressed as[63, 64]:

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O \qquad (\Delta H = -891\text{kJ/mol})$$

As is often the case with multi-step reactions, the overall reaction rate is a function of the concentration of all the intermediary chemical species' concentrations, along with the reaction rate coefficient of all the subsidiary reactions (both forward and backward rates)[60]. It is important to note that the reaction rate is dependent on both the products and the reactants of the intermediary reactions as the reactions are both forward and backward reactions. Temperature then becomes rather critical in the determination of the reaction rate. This is because the intermediary reaction rates have an inverse hyperbolic exponential relationship to temperature via the Arrhenius equation (see equation (21)), which correspondingly causes a drastic change in the concentrations of the intermediary species.

The overall reaction rate is hard to calculate exactly as it is the solution of competing non-linear coupled partial differentials. Classically this is resolved by assuming that the intermediary reactions have already reached steady-state, and then solving the consequences analytically. When this becomes laborious, ephemeral species are often neglected. Even applying this method (which is hardly consistent with the system of unstable combustions), producing an analytical solution of this can still be difficult (the steady-state equations are normally multiplicatively non-linear)[65].

Contrasting the analytical approaches, a numerical system of approximations can be solved in a vectorised manner that is closer to the physical problem. Whether such a model should be solved explicitly, implicitly or semi-implicitly is up to the design of a particular model. However, that if an implicit method is selected, the non-linear manner of the system does not allow for a simple matrix inversion approach as is typical for implicit schemes. Avectorised numerical model can be executed in a highly efficient manner with small computational processing requirement.

3.2. How reactions are modelled

To model the reaction term, the source function is approximated (as shown below, derived from equation (17)) by using a grid that is sufficiently refined such that the way that a chemical reaction produces heat and consumes a chemical species is invariant across the control volume:

$$\frac{\partial u_i}{\partial t} = S(U) \qquad \textbf{(18)}$$

Using a forward differencing equation for the temporal derivative operator in equation (18) the following numerical scheme is produced.

$$u_i^{n+1} = u_i^n + \Delta t^n S(u_i) \qquad \textbf{(19)}$$

The only thing still left to decipher in equation (19) is whether the argument of the reaction-source function should be at time step n or time step $n + 1$. If the scheme uses step n, the overall scheme becomes explicit, which is the fastest style of scheme to calculate answers for; however, it does produce issues of convergence – particularly when used around low concentrations or at large temperatures[66]. Indeed, because the source function to be approximated is often a function of exponentials, the ordinary differential system is often stiff (leading to exceptionally small time steps to gain even reasonable accuracy regardless of the convergence requirements of the scheme).

The other conventional choice is to consider time step $n + 1$, which does not incur the issues of stability. As previously discussed, the source function is often non-linear and so the implicit model cannot be solved by mere matrix inversion (which is already a time-consuming option). Indeed, the only immediate way that this model can solved implicitly is to use an iterative approach[67].

Both of these approaches can lead to misleading results. If solved explicitly, then due to stiffness, it is likely that the results produced are not truly reflective of the physics of the system (if the only requirement of the resolution of the grid and the size of the time step is the CFL condition). When solved implicitly, this stiffness is removed and no requirement is necessary to produce reasonable results – this often spuriously lends a degree of verisimilitude to the numerical results, independent on the coarseness of the grid or size of the time step[68]. However, the order of accuracy of implicit and explicit methods is very similar. This means that even though the implicit method would remove the stiffness, the method can still produce inaccurate results for a grid or time step that is inappropriate.

3.3. Augmented one step unimolecular reaction

In a one-step unimolecular combustion reaction, fuel is converted directly into reactants, as shown below in equation (20). For such a reaction, the reaction rate can be expressed by equation (21), where A is the pre-exponential factor, E_A is the activation energy, T is the temperature of fluid, R is ideal gas constant and $[F]$ is the mass fraction of the fuel[48].

$$F \rightarrow P \qquad \textbf{(20)}$$

$$\omega = A\rho[F]e^{-\frac{E_A}{RT}} \qquad \textbf{(21)}$$

It is assumed that the complex reaction can be accurately modelled by drastically reducing the complexity of the reaction by approximating the reaction rate, ω, with a one-step unimolecular reaction. However, as this model is reduced from a more complex reality, more versatility is gained by augmenting the model using a phenomenological expression for the chemical properties parametrised by the volume percentage of fuel.

$$\omega = A(\theta)\rho[F]e^{-\frac{E_A(\theta)}{RT}} \qquad \textbf{(22)}$$

3.4. Augmented one step bimolecular reaction

In the unimolecular reaction discussed in the previous section, the effect of the concentration of oxygen only affects the parametrisation of the chemical properties of the fluid by affecting the volume percentage of the fuel concentration. In reality, the effects of the concentration of oxygen is significantly more than this, as can be seen by the inaccuracies of previous unimolecular reaction models.

In a one-step bimolecular combustion reaction, fuel and oxygen are both converted into products, as shown below in equation (23). Similar to the unimolecular reaction, the reaction rate can be expressed by equation (24), where [O] is the concentration of the oxidiser. Note that the stoichiometry of the reaction is irrelevant to modelling the rate of the reaction as the stoichiometric constants are factored into the CFD component of the model.

$$F + O \rightarrow P \qquad \textbf{(23)}$$

$$\omega = A\rho[F][O]e^{-\frac{E_A}{RT}} \qquad \textbf{(24)}$$

Similarly to the unimolecular reaction model, to allow for a more complete picture of the complex reality and to aid with more versatility, the model is augmented using a phenomenological expression for the chemical properties parametrised by a local-equivalence ratio.

$$\omega = A(\theta)\rho[F][O]e^{-\frac{E_A(\theta)}{RT}} \qquad \textbf{(25)}$$

Additionally, by parametrising the chemical properties of the system to the local equivalence ratio, this model can simulate more complex relationships of combustion behaviour such as the species lower and upper flammability limits (which are themselves functions of the local equivalence ratios).

3.5. Application of implicit and explicit methods in reaction modelling

3.5.1. Unimolecular model

It is important that the developed model could simulate unimolecular reactions as they are often used as a comparison to other models, as well as a benchmark for contemporary models. As explained in section 3.3, unimolecular first order reactions have a reaction rate given by equation (21). Knowing that the reaction rate can be approximated by equation (21), the source term then becomes:

$$u_i^{n+1} = u_i^n + \Delta t^n A\rho[F]e^{-\frac{E_A}{RT}}\begin{pmatrix}0\\0\\0\\0\\q\\1\end{pmatrix} \qquad \textbf{(26)}$$

Noting that the vector on the right implies that the reaction only affects the internal energy and mass fraction of the fuel (which then has a secondary effect on the other variables).

Explicit method

In the explicit method, this scheme is calculated using only values from the previous time step, where the superscript n indicates that the value is derived from the previous time step.

$$u_i^{n+1} = u_i^n + \Delta t^n A\rho^n[F]^n e^{-\frac{E_A}{RT^n}}\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix} \qquad \textbf{(27)}$$

Semi-implicit method

In the semi-implicit method, the scheme uses a combination of previous time step values, as well as values that are solved simultaneously. To do this, one takes advantage of the Taylor series of the exponential function:

$$e^{-\frac{E_A}{RT}} = \sum_{k=0}^{\infty}\frac{(-\frac{E_A}{RT})^k}{k!} = 1 + \sum_{k=1}^{\infty}\frac{(-\frac{E_A}{RT})^k}{k!} \qquad \textbf{(28)}$$

For large values of temperature, the first term (the one) becomes increasingly more dominant, which is the linear term (in regards to temperature). This means, that as temperature increases, the linear term dominates, and as temperature decreases the non-linear terms dominate. One can state equation (27), guided by equation (28), as:

$$u_i^{n+1} = u_i^n + \Delta t^n A\rho[F]\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix} + \Delta t^n A\rho[F](e^{-\frac{E_A}{RT}} - 1)\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix} \qquad \textbf{(29)}$$

where the second term is linear and dominates when temperature is large, and the third term is non-linear and dominates when temperature is insufficiently large. Choosing to solve the linear term implicitly and the non-linear term explicitly (solving non-linear terms implicitly produces iterative methods which are not desirable) produces the following:

$$u_i^{n+1} = u_i^n + \Delta t^n A\rho^{n+1}[F]^{n+1}\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix} + \Delta t^n A\rho^n[F]^n(e^{-\frac{E_A}{RT^n}} - 1)\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix} \qquad \textbf{(30)}$$

This reduces to:

$$u_i^{n+1} = \left(I - \Delta t^n A\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 1\end{pmatrix}\right)^{-1}\left(u_i^n + \Delta t^n A\rho^n[F]^n(e^{-\frac{E_A}{RT^n}} - 1)\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix}\right) \qquad \textbf{(31)}$$

Now, one can leave Matlab (or any other matrix inversion software) to handle the inversion of this simple six by six matrix. However, whilst it is easy to verify by hand that none of the eigenvalues of this matrix are 0 (so the matrix inverse exists) the eigenvalues are sometimes very close to zero (in comparison to the largest eigenvalue of the matrix). This means that subtle rounding errors in the matrix can lead to significantly different values in the matrix multiplication. Indeed, if the largest eigenvalue is sufficiently large, when the eigenvalues are normalised (a step used to produce the inverse matrix) the smaller eigenvalues are sometimes within rounding error of zero (leading to a singular matrix) which is a fatal error. This can be easily rectified by doing the matrix inversion directly (meaning less rounding errors) which leads to:

$$u_i^{n+1} = \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 1 & 0 & 0 & 0 & 0\\0 & 0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 1 & 0\\0 & 0 & 0 & 0 & -\frac{-qA\Delta t^n}{1+A\Delta t^n} & \frac{1}{1+A\Delta t^n}\end{pmatrix}\left(u_i^n + \Delta t^n A\rho^n[F]^n\left(e^{-\frac{E_A}{RT^n}} - 1\right)\begin{pmatrix}0\\0\\0\\0\\q\\-1\end{pmatrix}\right) \qquad \textbf{(32)}$$

3.5.2. Bimolecular model

It is important that the developed model could simulate bimolecular reactions as they are used in conjunction with empirical correlations to model methane combustion. As explained in section 3.4, bimolecular first order reactions have the reaction rate given by equation (24). Knowing that the reaction rate can be approximated by equation (24), the source term then becomes:

$$u_i^{n+1} = u_i^n + \Delta t^n A\rho[F][O]e^{-\frac{E_A}{RT}}\begin{pmatrix}0\\0\\0\\0\\q\\-\nu_F\\-\nu_O\end{pmatrix} \qquad \textbf{(33)}$$

One should note that the state vector now has one more dimension than the unimolecular model. This is because it now contains both the concentration of the fuel, as well as the oxidiser.

Explicit method

In the explicit method, this scheme is calculated using only values from the previous time step.

$$u_i^{n+1} = u_i^n + \Delta t^n A \rho^n [F]^n [O]^n e^{-\frac{E_A}{RT^n}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ -\nu_F \\ -\nu_O \end{pmatrix} \quad \textbf{(34)}$$

Semi-implicit switch method

The following can be established for the semi-implicit method, where the second term dominates for large temperatures and the third term dominates for small temperatures.

$$u_i^{n+1} = u_i^n + \Delta t^n A \rho [F][O] \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ -\nu_F \\ -\nu_O \end{pmatrix} + \Delta t^n A \rho [F][O] (e^{-\frac{E_A}{RT}} - 1) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ -\nu_F \\ -\nu_O \end{pmatrix} \quad \textbf{(35)}$$

It is perhaps surprising to note that the linearity is not the same as the unimolecular model as now the first term involves both the fuel and oxidiser concentration. To alleviate this concern the switch method is used, whereby the second term is solved implicitly for the fuel concentration the first time, and the oxidiser concentration the second time. This works well in conjunction with the fractional step method, as discussed in section 2, as it must already split the source term into two. To remove any variable bias, the order in which the source term is calculated (for the fuel or oxidiser concentration) is alternated. The same concern of matrix inversion accuracy for the unimolecular reaction is relevant in this case as well; therefore, the matrix is inverted by hand. The Semi-implicit scheme is therefore:

$$u_i^{n+1}{}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{qA\Delta t^n [O]^n}{1-\nu_F A\Delta t^n [O]^n} & \frac{1}{1-\nu_F A\Delta t^n [O]^n} & \frac{-\nu_O A\Delta t^n [O]^n}{1-\nu_F A\Delta t^n [O]^n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \left(u_i^n + \Delta t^n A \rho^n [F]^n [O]^n \left(e^{-\frac{E_A}{RT^n}} - 1\right) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ -\nu_F \\ -\nu_O \end{pmatrix} \right) \textbf{(36)}$$

$$u_i^{n+1}{}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{qA\Delta t^n [F]^n}{1-\nu_O A\Delta t^n [F]^n} & \frac{-\nu_F A\Delta t^n [F]^n}{1-\nu_O A\Delta t^n [F]^n} & \frac{1}{1-\nu_O A\Delta t^n [F]^n} \end{pmatrix} \left(u_i^n + \Delta t^n A \rho^n [F]^n [O]^n \left(e^{-\frac{E_A}{RT^n}} - 1\right) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ q \\ -\nu_F \\ -\nu_O \end{pmatrix} \right) \quad \textbf{(37)}$$

IV. THERMODYNAMICS

4.1. Thermochemical and thermophysical properties

The total energy of an ideal gas can be said to be a composition of kinetic energy and thermal energy. The thermal energy can be then described as a function of pressure. This can be stated as below[48]:

$$E = E_k + E_{Th} = \frac{\rho}{2} v^2 + \frac{P}{(\gamma - 1)} \textbf{(38)}$$

where v is the velocity, ρ is the density, γ is the ratio of specific heats and P is the pressure.

Equation (38) means that the pressure can be calculated if the velocity and the relative energy are known. Lastly, if the temperature can be described through the state vector, equation (6) would be closed (all the variables parametrised to the state variable) except for the physical characteristics of the gas. By assuming that system behaves as an ideal gas, the temperature must be identified by equation (39)[48].

$$T = \frac{PM}{\rho R} \textbf{(39)}$$

The speed of sound is necessary to enforce the convergence requirement of the CFD schemes. The speed of sound can be fundamentally described as the first equality in equation (40), if the system behaves ideally then the second equality is also valid[48].

$$a = \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_s} = \sqrt{\frac{\gamma P}{\rho}} \textbf{(40)}$$

In some models, physical characteristics of the gas can be considered to be constant, and therefore, merely additional input parameters of the model. In this case, the model is to be designed in a manner that can handle all ranges of deflagration, detonation and DDT. This means that at any time, the temperature and pressure ranges within the system can differ by an order of magnitude or more. This diverse range in temperature causes a large range in the properties of the gas. To handle this concern, the viscosity, the diffusivity, and thermal conductivity are described by the following empirical relationships where the subscript 0 denotes an initial condition and $\varkappa$ is the thermal diffusivity [48]. For methane, which has a Lewis number around unity, we can note the equality between D_0 and $\varkappa_0$, and thus only require information of ν_0, and D_0 to parametrise the physical properties of the fluid.

$$\nu = \nu_0 \frac{T^{0.7}}{\rho} \quad \mathbf{(41)}$$

$$D = D_0 \frac{T^{0.7}}{\rho} \mathbf{(42)}$$

$$K = \varkappa_0 \frac{T^{0.7}}{\rho} c_P = D_0 \frac{T^{0.7}}{\rho} \frac{\gamma R}{\gamma - 1} \mathbf{(43)}$$

4.2. Resolution of the numerical method

The convergence of the model and how rapidly the scheme converges to the analytical solution is an important issue. In all numerical methods, the problem then becomes determining how accurate is sufficiently accurate. This problem is sometimes tackled by numerical optimisations (running the simulation at different levels of refinement to optimise the computational costs and the numerical improvements). In general, one wants the schemes to produce results that are realistic, at least qualitatively. Due to discontinuities, non-smooth distributions, turbulence and the non-linear reaction model, the differential-integral equations are stiff (require step sizes that are smaller than merely ensuring convergence, to ensure important qualities of the solution are described). The issues of discontinuities, lack of smoothness, and turbulence are in general well resolved by use of monotonic schemes and total variation diminishing (TVD) schemes. TVD scheme is described in later part of the article. However, the added complexity of the reaction causes additional concerns in the CFD component of the model.

The stiffness in the reaction model itself, for this project, is dealt with by use of a semi-implicit scheme created for this project and is described in the methodology section of this report. However, the effects that this reaction imposed onto the CFD component can be resolved by considering the physical system. For some models, this can be simplified using empirical correlations in either the deflagration regime or the detonation regime. The model is to be designed such that it can handle both deflagration and detonation regimes concurrently in the same system. For that reason, the solution to the stiffness must correctly handle both regimes as well as come from the perspective of the fundamental physics of the system (to handle their transitional regime as well as to reduce the required empirical data). Indeed, to handle the stiffness, one sets the maximum refinement (minimum characteristic length of a control volume) to be one tenth of the minimum of the laminar flame thickness and the half reaction thickness [48].

This then leaves the question of calculating the laminar flame thickness and the half reaction thickness. Laminar flame thickness is defined under a one dimensional system for purely laminar flow conditions with smooth distribution when the system is in steady-state. In the steady-state condition, the governing equation becomes a coupled ordinary differential equation in regards to the temperature profile as shown below in equation (44) [48]. It should be noted that the answer to this differential equation is again, a function of the choice of the reaction model.

$$\frac{dF_t}{dx} = \rho\left(UC_P \frac{dT}{dx} - q\omega\right);\ F_t = K \frac{dT}{dx} \mathbf{(44)}$$

From this temperature distribution the laminar flame thickness and the laminar flame velocity can be calculated by the following expression in equation (45), where T_b is the adiabatic flame temperature, x_l is the laminar flame thickness, and S_l is laminar velocity[48].

$$x_l = \frac{T_b - T_0}{\max\left(\frac{\partial T}{\partial x}\right)}; S_l = \frac{U\rho}{\rho_0} \mathbf{(45)}$$

In the case of the half reaction thickness, the definition is considered for the turbulent flow regime, with purely detonation conditions; the governing differential equation then becomes the following coupled ordinary differential equations[48].

$$\frac{d\rho}{dt} = \frac{q\omega\rho\,(\gamma-1)}{(U^2 - c^2)}; \frac{de}{dt} = \frac{P}{\rho^2}\frac{d\rho}{dt} - q\omega;\ \frac{dx^*}{dt} = U\mathbf{(46)}$$

From this temperature distribution the half reaction thickness and the turbulent flame velocity can be determined. Fortunately, the half reaction thickness and the laminar flame thickness can be experimentally found, so that the source model can be empirically parametrised.

4.3. Estimation of Laminar flame thickness and characteristic time

Equation (44) displays the coupled governing equation for the temperature profile. Writing the equation in terms of vector addition gives the following expression:

$$\begin{pmatrix} T_{i+1} \\ F_{t_{i+1}} \end{pmatrix} = \begin{pmatrix} T_i \\ F_{t_i} \end{pmatrix} + \Delta x_i \begin{pmatrix} F_{t_i} \\ \rho\left(UC_P \frac{F_{t_i}}{K} - q\omega(T_i)\right) \end{pmatrix} \qquad \textbf{(47)}$$

This can be solved iteratively on a uniform grid directly. Refining the uniform grid and comparing the result to the coarser grid allows for a measure of convergence. Continuing to refine the grid whilst comparing the new distribution to the previous distribution allows for a continued measure of convergence. Once the convergence ratio is less than some specified error threshold the distribution is considered to be stable and the laminar thickness and velocity is calculated by equation (45).

4.4. Estimation of Characteristic lengths and times for turbulent flames

Equation (46) displays the coupled governing equation for the temperature profile. Writing the equation in terms of vector addition gives:

$$\begin{pmatrix} \rho_{n+1} \\ e_{n+1} \end{pmatrix} = \begin{pmatrix} \rho_n \\ e_n \end{pmatrix} + \Delta t_n \begin{pmatrix} \frac{q\omega_n \rho_n (\gamma-1)}{U^2 - {c_n}^2} \\ \frac{P_n}{\rho_n^2}\left(\frac{\rho_{n+1}-\rho_n}{\Delta t_n}\right) - q\omega_n \end{pmatrix} \qquad \textbf{(48)}$$

Clearly, the first component must be solved first as the second component requires information of the current first component. Like in the laminar case, this is solved on a uniform grid iteratively, with each iteration solved on a finer resolution until convergence is found. Once the distribution is created the detonation thickness and characteristic time can be calculated.

V. COMPUTATIONAL FLUID DYNAMICS

5.1. Convergence

The majority of fluid dynamics models use an explicit numerical scheme. The reasons for this include the advantage in processing times, decrease in developmental complexity and the chaotic nature of the problem (non-linear systems are traditionally difficult to solve implicitly without some form of iteration which always introduces more processing time and questions of how to know when to stop iterating – what local error threshold is required to produce an adequate global error?). However, explicit schemes introduce another concern to the model: stability.

The stability of the model refers to the necessary conditions which will enforce the model to converge to a valid solution. It is important to remember that stability and accuracy are different: stability deals with the production of valid (non-infinite) solutions; whereas accuracy deals with how well produced solutions align to the correct solution of the theoretical equations (and therefore, if given adequate assumptions, the experimental results). The discussion of erroneous artefacts that cause unnecessary oscillations produced by a particular numerical scheme or solution methodology can often be considered under either accuracy or stability (as it can, and often does, affect both). In this article, this will be addressed under the concerns of convergence.

The stability of an explicit numerical scheme is almost always a function of the ratio of the characteristic length of the control volumes the model is using (the spatial resolution) to the size of the time step (temporal resolution) [50]. This ratio is often called the resolution velocity (due to its dimensions). Further, the majority of such numerical schemes applied to fluid dynamics ends up having a stability condition dependent on a dimensionless number called the Courant-Friedrichs-Lewy (CFL) number[69-71]. The CFL number is the ratio of the wave velocity, ω, in a control volume, to the resolution velocity of the control volume as shown below in equation (49)[50]. Physically, this condition enforces the situation that the model will be sufficiently resolved such that no component of the fluid will have a sufficient wave velocity to affect control volumes that are not directly adjacent.

$$\mathrm{CFL} = \frac{\boldsymbol{\omega}}{\frac{\Delta x}{\Delta t}} = \frac{\boldsymbol{\omega}\Delta t}{\Delta x} \qquad \textbf{(49)}$$

This is depicted below in Figure 3 where the differences between explicit and implicit schemes are shown, as well as the physical representation of the CFL condition, all for the same flow field. Subfigures (a) and (b) both exemplify the explicit scheme, where the current derivative is extrapolated to approximate the new position of the fluid. In subfigure (a) the flow vector has sufficient magnitude to imply that the flow of the fluid goes past the adjacent cells. However, a problem exists in that the model assumes all of the flow moves only into adjacent cells, meaning that instead of some amount of the fluid moving across two cells, twice as much fluid is modelled to be moved into the adjacent cell. Subfigure (b) shows the enforcement of the CFL condition by reducing the time step, such that the flow can be modelled to first go into the adjacent cell, and then the next cell. The overall fluid flow remains the same as subfigure (a) would predict, as the flow field is sufficiently similar. It is noteworthy that if the direction of the flow field of the adjacent cells is drastically different to the original cell, then the flow predicted by a time step that violates the CFL condition could be significantly different than any that enforces the CFL condition. Subfigure (c) depicts the implicit scheme, where a cell is found such that the flow field vector would “back-predict” the original cell. Whilst the implicit scheme is not used in the CFD component of this report, the concept of implicit schemes is used in the reaction component of the model. In the implicit scheme, the CFL condition is not required for stability although accuracy still depends on the spatial and temporal resolution. It is clear that if the explicit step was used on the cell for subfigure (c) then the result would be significantly different. More so, if the time step was not reduced enough to enforce the CFL condition, such that the difference in direction of the flow field (in the adjacent cell) was not identified. It is occasionally the case that the manner in which the flow field is produced is sufficiently non-linear such that enforcing the CFL condition is insufficient to ensure that this difference in flow field is fully identified by the scheme. If a differential system implies such a flow field, the system is described as being stiff. The implicit scheme resolves such stiffness well. The stiffness that is seen in the CFD component is due to the effects of turbulence. The manner in which this stiffness is identified is discussed below (without the use of implicit schemes). The stiffness in the reaction component of the model is due to the highly non-linear representation of reaction rates (caused by the exponential terms in the Arrhenius equation).

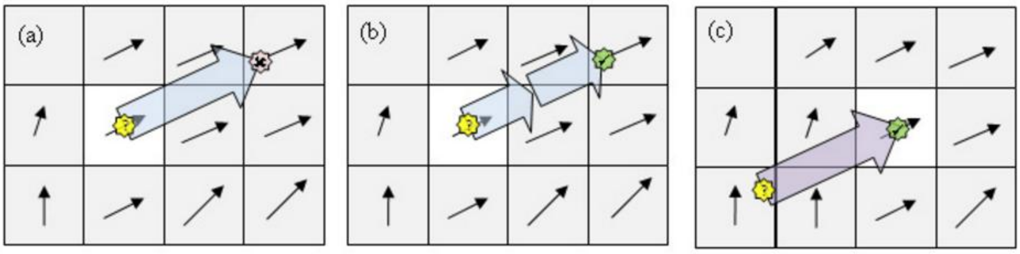


Figure 3: A depiction of the difference between the implicit and explicit calculation of the intermediary flow vectors based on the flow field. Both (a) and (b) depict an explicit calculation that extrapolates the direction of the derivative at the current point. The difference between (a) and (b) is the time step used to calculate the flow vector. In the case of (a) the CFL condition fails; whereas, in the case of (b) the CFL condition is enforced by reducing the time step. In the case of (c), the solution is found implicitly. Importantly, the implicit method identifies a location whose derivative “back tracks” to the original cell, and so the CFL condition is not required for stability. This figure was produced by[72] .

It is often the case that explicit finite volume methods produce a stability requirement of the CFL number being less than unity. Indeed, all of the numerical schemes discussed in this article require a CFL number of less than unity[73]. This means that one can calculate the largest value for the next time step if the wave velocity in each control volume is known. It is common to then multiply this ‘maximum’ time step by a number between zero to one called the ‘CFL number’ as there is often errors correctly identifying the wave velocity. This is expressed in equation (50). Generally speaking, the CFL number can be selected to be between 0.7, during stages of instability, and 0.9, during stages of stability.

$$\Delta t = CFL\frac{\Delta x}{\boldsymbol{\omega}} < \frac{\Delta x}{\boldsymbol{\omega}} \tag{50}$$

The wave velocities are often difficult (and computationally expensive) to find exactly, so instead it is often approximated for compressible fluids with a new sequence S_{max}^n which is defined in equation (51) where a_i^n is the speed of sound in the *i*th control volume after the *n*th time step, and $\boldsymbol{U}_i^n$ is the velocity in the i^{th} control volume after the *n*th time step[50].

$$\Delta t = cfl\frac{\Delta x}{\boldsymbol{\omega}} \sim cfl\frac{\Delta x}{S_{max}^n} \tag{51}$$

$$S_{max}^n = \max_i(|\boldsymbol{U}_i^n| + a_i^n) \tag{52}$$

Important to the identification of stable and accurate numerical schemes is the notion of the property of variation diminishing. A numerical scheme is said to be total variation diminishing (TVD) if it has the property that the L^1 norm of its derivative is monotonically decreasing in respect to time[74-76]. This is expressed below in equation (53) in terms of a continuous function and (54) in terms of a discrete sequence where $\boldsymbol{x}$is the spatial vector, ε is some positive value, and where the inequality means inequality for every element of the vector.

$$\int \left|\frac{\partial \boldsymbol{U}(\boldsymbol{x},t)}{\partial \boldsymbol{x}}\right| d\boldsymbol{x} \leq \int \left\|\frac{\partial \boldsymbol{U}(\boldsymbol{x},t+\varepsilon)}{\partial \boldsymbol{x}}\right\| d\boldsymbol{x}\,, \forall \varepsilon \geq 0 \qquad \textbf{(53)}$$

$$\sum_j \left|\boldsymbol{U}_{j+1}^{n} - \boldsymbol{U}_{j}^{n}\right| \leq \sum_j \left|\boldsymbol{U}_{j+1}^{n+1} - \boldsymbol{U}_{j}^{n+1}\right| \qquad \textbf{(54)}$$

The property of TVD is important in schemes that attempt to describe discontinuous distributions without inducing spurious oscillations. In general, most well designed numerical schemes can predict discontinuities without the production of incorrect oscillations given a sufficient amount of spatial resolution[73]. The advantage of TVD schemes is that they still remain accurate with coarse resolution. This means that TVD schemes are consequentially more advantageous for explosive modelling due to the inherently discontinuous nature of state variables – as these discontinuities can be well described in an economic manner, and the schemes don't cause large compounding errors in the model. Therefore, TVD schemes save computational processing time (by allowing for a less refined grid).

Another important definition for numerical sequences in CFD is monotone sequences. A (differentiable) numerical scheme, M, is considered monotone if and only if it satisfies the following equation[50]:

$$\frac{\partial M}{\partial u_k^n} \geq 0, \forall k \qquad \textbf{(55)}$$

Monotone sequences form the foundation for a considerable amount of modern CFD, and takes a pivotal role in the CFD component of this developed model, for reasons that are explained in the following sections. A sequence is monotone everywhere if it satisfies equation (55) for all CFL conditions. If a sequence is only monotone for conditional CFL numbers, then it is said to be monotone in that region. Fortunately, a monotone sequence must also be TVD. Further a scheme that has the property of being TVD preserves the monotone property (but will not necessarily cause it). The importance of this relationship between TVD and monotone schemes is discussed more completely in section 5.3.

5.2. Turbulence

Turbulence causes many issues in attempting to approximate fluid dynamics. It adds chaos to the system causing small errors to compound into large errors in a rapid fashion. It is, in general, possible to completely resolve and identify the effects of turbulence through modelling viscous terms so long as a sufficiently fine grid is used such that the control volumes are down to the 'viscous scale'. However, this is often impractical as this viscous scale is typically very small in comparison to the size of the domain that is attempting to be modelled, meaning that a very large number of control volumes would be required if the domain is divided uniformly [48]. This leads to a high demand on computational processing time. Models that would attempt to solve turbulence this way are impractical for most engineering purposes.

Since the advent of adaptive mesh refinement (AMR) it has been possible to use a non-uniform gridding in order to have smaller sized control volumes in areas of high turbulence without the need for over refinement in non-critical areas. AMR is further discussed in section 6.3. This partially solves this issue; however, it is still often too costly to refine down to the viscous scale.

The majority of modern models will use a coarse grid (the level of coarseness that the model will employ will depend on the geometry of the system as well as whether or not the model takes advantage of AMR techniques) and then attempt to impose the predicted effects of turbulence that would occur at a smaller scale. One of the most common ways is to apply a Reynolds average version of equations (1) through to (4) [73]. The exact manner that the 'subscale' effects are predicted and imposed on to the coarser computational grid depends on the turbulence model chosen. The choice of which turbulence model is used is strongly dependent on the conditions of the system as these models are built partially from averaged physics as well as empirical correlations. In some cases, these correlations will make use of parameters that can be approximated from the interim properties of the simulation itself, where others require some ad hoc parameter [48].

Some models leave the choice of turbulence model to the user – implicitly meaning that the user of the model must be able to predict some properties of what will happen, and these properties must not change. Other models have the ability to identify when a turbulence model should be used inherently. The majority of models use some form of turbulence resolution and therefore have large issues dealing with both regimes of deflagration and detonation due to their significantly different properties [73]. These issues become more drastic as these models attempt to predict what will occur during DDT when both the experimental properties of deflagration and detonation are simultaneously present.

It has been noted that when CFD is solved using a numerical scheme that has the property of being monotone, it will smear viscous effects in a similar way to turbulence does physically, meaning that neither resolving down to the viscous scale nor the implementation of a turbulence model (other than the fundamental viscous tensor) is necessary. Not only does this simplify in the model development, as it removes the requirement of turbulence models, it also improves the limit of the accuracy of the model. This is because the monotone schemes allow for correctly resolving the important scales of turbulence; whereas, turbulence models will always average out the grid, even if the computational grid is fine enough to capture the required information leading to a loss of accuracy.

The idea of using monotone schemes without the use of a turbulence model is referred to as monotone integrated large eddy simulation (MILES). These methods are successful at modelling turbulence in part due to their TVD nature as well as their monotony which has been shown to dissipate energy, at the control volume scale, in a way that is isomorphic to natural viscous dissipation. Essentially, this means that MILES schemes are not only appropriate numerical schemes to solve a physically derived integral-differential equation but also solve the problem in a very physical way[48].

This means that monotone schemes and monotone preserving schemes are powerful in CFD models that are used for combustion modelling as they are able to handle the drastic discontinuities that can occur, along with being able to cope with a coarser grid when handling turbulence.

5.3. The rate of convergence

In section 5.1, the categorisation and requirements of a numerical method's convergence was discussed in terms of if it will converge to a solution, and if this solution is the intended distribution. For practical applications, it is necessary that the used numerical methods converge; however, convergence of a numerical method is insufficient. Indeed, it is necessary that the model must produce results within a reasonable time. For this reason, a discussion on the rate of the convergence is important.

This is done mathematically by discussing the asymptotic behaviour of the error of the numerical scheme as the number of cells approaches infinity. To ease calculations, as well as to ease comparison to other schemes, the behaviour of the error is described in terms of big O notation[77-79], which is often referred to as the order of the scheme or order of accuracy. It is standard to describe the error of a scheme in terms of orders of polynomials, such as $O(h)$ or $O(h^2)$. The importance of the previous two statements is that if a scheme is in $O(h)$ then its error decreases approximately linearly (after a sufficient number of cells) for an increase in the number of cells. This differs from the second example of $O(h^2)$ which describes a scheme whose error decreases quadratically with an increase in the number of cells. The practical implications of this is that a scheme with a higher order accuracy is far more efficient with the cells it uses, and can obtain the same measure of error with significantly fewer cells. In general, if the number of cells decrease then the speed at which a model can produce meaningful results increases; however, as the complexity of the method increases so does the computational time on each cell. Further, higher order schemes often lead to more significant rounding errors.

Clearly, a desirable scheme must have the properties of being TVD and preferentially monotonic with a large order of accuracy. It is unfortunate, however, that all schemes that are everywhere monotone can only be first order accurate [50]. This means that monotone schemes, compared to other numerical schemes, require a large number of cells to give the same level of error. It is often the case that the powerful resolution of discontinuities and turbulence that monotone schemes endow to a model is infrequently required. It is therefore inefficient for a model to use a monotone scheme everywhere, when the advantages of a monotone scheme are often only required in specific locations in the computational grid.

To overcome this difficulty, flux limiters were defined in such a way to identify when the monotone properties where required[80]. This allowed numerical schemes to be developed which are monotone in locations that require it, but remain higher order accurate when it is not required, allowing a model to reap the benefits of both options. Equation (56) shows a generalised manner in which this can be adopted, where T is a combined scheme, M is a monotone scheme, H is a higher order scheme and φ is a flux limiter. A direct example of this is shown in the following section.

$$T^i = M^i + \varphi^i (H^i - M^i) \qquad \textbf{(56)}$$

5.4. Discussion of numerical schemes

Navier-Stokes equations have been traditionally employed in the attempt to solve CFD. The schemes introduced in this section hope to solve equations (14) through to equation (16). Important to the theoretical modelling of explosion are the following schemes that approximate the flux leaving a cell and flowing into the adjacent cell. The first two schemes are in some way the fundamental schemes, upon which the next two schemes are built.

The Lax-Friedrich scheme is shown below[73]. It is interesting to note that this scheme is comprised of a term that averages the fluxes centred at each cell, as well as a numerical diffusive term. Without the numerical diffusion this flux would be unconditionally unstable. Counter-intuitively, this scheme will also solve, directly, the differential equation (58) which is the same as equation (14) – the equation it is designed to approximate – except for the added diffusion term (which is scaled by some grid dependent non-zero value α) which asymptotically approaches zero as the grid becomes more refined. By ‘solve directly’ one means that it is mathematically consistent with. It is in fact possible for one numerical scheme to be consistent with a different differential system that is “sufficiently” close to the intended differential system in order for it to be a good approximation[81].

$$F^{LF} = \frac{1}{2}(F(U_i{}^n) + F(U_{i+1}{}^n)) + \frac{\Delta t}{2\Delta x}(U_{i-1}{}^n - U_i{}^n)\textbf{(57)}$$

$$U + \frac{\partial F(U)}{\partial x} = \left(\frac{(\Delta x)^2}{2\Delta t}\right)\frac{\partial^2 U}{\partial x^2}(\to 0 \text{ asthe } \Delta x \to 0)\textbf{(58)}$$

It should be noted that this added artificial diffusion term causes the predictions of this model to appear less sharp, but also means that this system is monotonic everywhere, TVD, and formally first order in space.

Didactically opposite to the Lax-Friedrich scheme is the Richtmyer scheme which is shown below[73]. This scheme attempts to identify what the state vector of the fluid is in between the two cells and then evaluate the flux at that point. This scheme is formally second order in space. However, it is neither TVD nor monotonic.

$$U^n_{i+\frac{1}{2}} = \frac{u^n_{i+1}+u^n_i}{2} + \frac{\Delta t}{2\Delta x}\big(F(U_i) - F(U_{i+1})\big); \quad F^{RI}{}_{i+1} = F\left(U^n_{i+\frac{1}{2}}\right)\textbf{(59)}$$

As a sort of compromise to these two methods is the FORCE scheme. This scheme combines the two as an average. Unlike the Lax-Friedrich scheme, this scheme is not monotonic everywhere. However, it is monotonic in the region $CFL < 1$, which is a necessary requirement for convergence and thus is no more restrictive. This means that the FORCE scheme is monotone whenever it is convergent [73]. Further, this scheme is formally second order when the state vector distribution is smooth and first order otherwise.

$$F^{FORCE}_{i+\frac{1}{2}} = \frac{F^{LF}_{i+\frac{1}{2}}+F^{RI}_{i+\frac{1}{2}}}{2}\textbf{(60)}$$

Improving upon this method is the FLIC scheme which is a weighted average of the FORCE scheme and the Richtmyer scheme[73]. This scheme remains monotonic around discontinuities and is formally first order accurate around discontinuities and second order elsewhere.

$$F^{FLIC}_{i+\frac{1}{2}} = F^{FORCE}_{i+\frac{1}{2}} + \varphi_{i+\frac{1}{2}}\left(F^{RI}_{i+\frac{1}{2}} - F^{FORCE}_{i+\frac{1}{2}}\right) \qquad \textbf{(61)}$$

The exact choice of flux limiter is somewhat arbitrary. It has been found that the MC-limiter used in conjunction with the FLIC method produces effective results for combustion reaction modelling of fluid dynamics[73]. The MC-limiter is defined as the minimum of the adjacent internal flux limiters.

$$\varphi_{i+\frac{1}{2}} = \min\left(\varphi\left(r^L_{i+\frac{1}{2}}\right), \varphi\left(r^R_{i+\frac{1}{2}}\right)\right)\textbf{(62)}$$

Where the internal flux limiter is defined as follows for any argument r:

$$\varphi = \begin{cases} 0\ if r < 0 \\ 2r if\ 0 \le r < \frac{1}{3} \\ \min\left(2{,}0.5 + \frac{r}{2}\right) if r \ge \frac{1}{3} \end{cases} \quad \textbf{(63)}$$

The argument of each internal flux limiter is defined as below, where the δ operator acts as a numerical difference and the overall arguments depict a type of numerical derivative[73].

$$r^L_{1+\frac{1}{2}} = \frac{\delta U_{i-\frac{1}{2}}}{\delta U_{i+\frac{1}{2}}};\ r^R_{1+\frac{1}{2}} = \frac{\delta U_{i+\frac{3}{2}}}{\delta U_{i+\frac{1}{2}}};\ \delta U_{i-\frac{1}{2}} = U_i - U_{i-1}\textbf{(64)}$$

The four most common flux limiters are the SUPERBEE, the MINBEE, the MC-limiter and the VanLeer. The SUPERBEE has the largest flux limiter value for a particular argument, and the MINBEE has the smallest; the MC-limiter takes on a good middle ground with a slight tendency to lean more towards the

SUPERBEE as shown below in Figure 4.

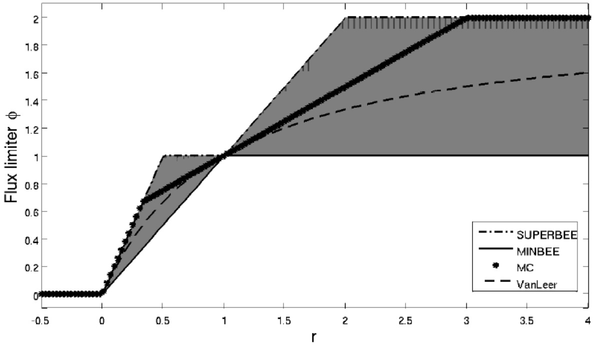


Figure 4: Flux limiter comparison for varied arguments of ‘r’. It is important that all flux limiters have a range of zero to two and take the value one when their argument is one. It is for this reason that there is no surprise that all the limiters agree when r is one, and have a similar range. This Figure was produced by[73].

VI. COMPUTATIONAL PROCESSING

A gas explosion model can be applied onto a computational grid where the grid is solved iteratively by a computer. The use of the computer introduces additional theoretical and practical concerns. The main computationalprocessing topics that are relevant to this article are parallel processing, rounding errors and adaptive gridding, which are individually discussed below.

6.1. Parallel processing

Parallel processing is a computation technique whereby a large number of sequential operations can become a small number of simultaneous operations. This technique requires that the tasks are independent (the numerical schemes are independent of current control volumes’ values except the central control volume)[82]. This works by a client service splitting the task up into subtasks and assigning the subtasks to different workers. Whilst this method often leads to a rapid increase in performance (reduced processing time) it can sometimes cause a worse performance due to large over-heads in the allocation of resources (the client informing the workers the details of their jobs)[83].

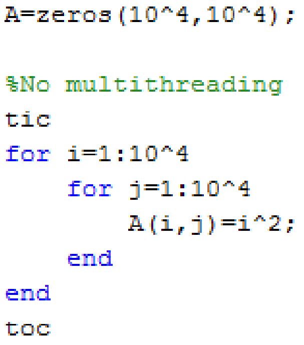
Matlab can be used in the modelling of gaseous explosion. In this article, parallelisation will be discussed only in reference to their applications in Matlab. It is typical to categorise parallelisation into implicit and explicit where implicit parallelisation is done in a way that Matlab inherently converts the code into C, and explicit parallelisation is where the code explicitly tells Matlab to handle the jobs in parallel.

6.1.1. Intrinsic Matlab parallelisation

Matlab will automatically multithread (a technique that for this application works the same, or similarly to, parallelisation) matrix and vector operations such as matrix (or vector) addition, multiplication, exponentiation and maximum/minimum functions[84]. This is particularly evident when comparing the following ways of calculating a didactic matrix whose entries are defined mathematically below:

$$A_{i,j} = i^2 \qquad \textbf{(65)}$$

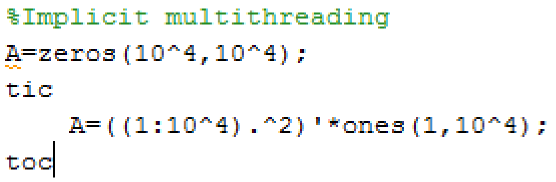
This simple mathematical expression can be programmed into Matlab in a few different ways. Below is a code that creates the matrix A.



```
A=zeros(10^4,10^4);

%No multithreading
tic
for i=1:10^4
    for j=1:10^4
        A(i,j)=i^2;
    end
end
toc
```

Alternatively, below is another code that creates the exact same matrix. The difference between the two codes is that the first script calculates each component of the matrix individually; whereas, the second script is calculated using vector operations and therefore inherits the advantage of multithreading. The first script, producing the same result, takes 27 times as long as the second script.



```
%Implicit multithreading
A=zeros(10^4,10^4);
tic
    A=((1:10^4).^2)'*ones(1,10^4);
toc
```

Once the size of the problem becomes significantly large, this difference becomes increasingly noticeable. Indeed, it can often be the difference between a practical solution and an impractical one.It is for this reason that the choice of converting the equations into a vector form (and corresponding vector functions) allows for a quicker calculation for the problem.

6.1.2. Extrinsic Matlab parallelisation

Whilst intrinsic Matlab parallelisation is simple and effective, it can only be used in cases where the code can be represented in terms of vector operations and elementary vector construction. The major components of this model, such as the CFD and the reaction modelling, cannot be described in terms of vector operations alone due to the increased complexity in the operations. This leads to extrinsic parallelisation, whereby the code makes use of “parfor” loop structures with use of the “parpool” command. The requirements of this structure are more complex but the technique can be applied more generally[84].

Explicitly, the most important requirements are: 1) the iterations of the loop must be completely independent; 2) the ordering of the loop must be irrelevant; 3) variable classification must be clear; 4) the inputs must be transparent (clearly seen by the compiler); and, 5) the variables must be sliceable. These five requirements can be summarised (when applied to this model) as 1) the numerical scheme must only require information about neighbouring cells of the previous time step and the current cell’s value (previous time step or current time step); 2) the rest of the code cannot rely on the final values of any loop/scheme structure; 3) all variables must be used unambiguously and only in a prescribed manner; 4) the required information of each cell must be clearly declared so subprograms cannot be used that rely on global variables, instead functions (that have global variables as inputs) must be used; 5) the variables should be readily separable.

Whilst the requirements of the explicit parallelisation can lead to code that is bulky and convoluted, the significant increase in processing time is paramount to a numerical model’s ability to practically compute sufficiently resolved computational grids.

6.2. Rounding error

Matlab holds numbers in terms of floating point numbers which are analogous to a binary form of the standard scientific notation. Necessary for a computer to hold real numbers, the number must be rounded to the closest floating point number. Unfortunately, this must be done every time a new operation is conducted, meaning that in a single equation or variable-assignment it is likely that Matlab must round multiple times.

A great deal of research and effort has gone into optimising computer calculation methods/algorithms such that their rounding error is sufficiently small so that most applications are not affected. However, in the case of the numerical techniques used for the model of gaseous explosion (and indeed, when derivatives are approximated in general) the difference between two similar values is required, along with the ability to divide by small numbers. This can be seen by observing the defining ratio of the derivative:

$$f'(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \Rightarrow f'(x) \approx \frac{f(x+h)-f(x)}{h} \qquad \textbf{(66)}$$

Clearly if the derivative is to be closely approximated, then smaller and smaller values of h are desirable. As the function is continuous, $f(x+h)$ becomes closer and closer to $f(x)$ as h becomes closer and closer to zero. This means that the model will routinely require subtracting nearly the same value, and dividing by a value that would be desirably arbitrarily small. However, if $f(x+h)$ becomes within rounding error of $f(x)$, their difference becomes zero causing significant problems in the approximation. Even when their difference does not round to precisely zero, rounding errors are still often large enough to cause significant errors. Indeed, it is sometimes the case that the subtraction of similar values will result in a number whose figures are all meaningless(that is, a number that has a relative error of one hundred percent or more)[85]. Further, if h becomes within rounding error of 0, a division of zero occurs causing a fatal error in the model. Even if h is not within rounding error of precisely 0, dividing by a small number escalates the numerical errors already introduced from the difference of two similar values.

In general, the sensitivity of a scheme to rounding error increases with the complexity of the scheme. This is because the higher order schemes often involve higher powers of *h* in the division, and the subtraction and addition of more terms that are nearly equivalent.

This leads to the rather contradictive concept that higher order schemes are mathematically more accurate (in general) but are computationally less accurate due to rounding errors for small *h*. Further, smaller values of *h* will lead to a mathematically more accurate approximation but sometimes a computationally less accurate one. If the derivatives of the solution are known and the machine epsilon (the accuracy to which the computer is recording numbers) is known, then the optimal trade-off between the improved accuracy of decreasing the step size and increasing the size of the rounding errors can be approximated.

In general it has been noticed that the second order accurate schemes have a good trade-off between improved accuracy of the scheme, and rounding errors produced due to the scheme, when using a language that uses double precision floating point numbers (the order of accuracy of Matlab)[84, 85].

A necessary consequence of rounding errors is that the more simplified an expression is put mathematically, the more accurate the computation becomes. This is particularly important when it comes to matrix inversion when some of the matrix's eigenvalues are close to zero (compared to the largest eigenvalue).

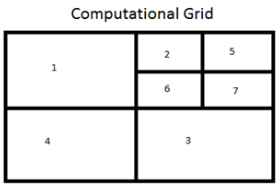
6.3. Adaptive grid mesh

It is often the case in combustion modelling that a large number of control volumes are required in localised areas due to the chaotic role of critical sections in the system, along with the stiffness in the governing differential equations. On a simple uniform grid, this means that a large number of processes are done while calculating highly precise information in unimportant areas. This causes an unnecessarily high requirement of computational resources for no real increase in accuracy. Adaptive Grid Mesh (AGM) is a technique that comprises of three steps: creating a grid over the system initially, iteratively identifying the critical areas inside of the system (such as shock waves, reactions fronts, high temperatures, low densities, etc.) and deciding whether or not particular areas are sufficiently refined or over-refined by use of refinement indicators, and lastly, refining or un-refining the grid[86-89].

6.3.1. Grid initialisation and internal locating

Different types of AGM exist, all of which have their own unique structure (and therefore ordering and locating methods). The style that is relevant to the method used in this model is the Fully Threaded Tree (FTT) structure[90-93]. To completely explain the importance and methodology of FTT, the two conventional ways of developing AGM must be discussed. These two methods were to either use an adjacency matrix, or a tree structure[55].

The adjacency matrix paradigm has its origins in topological graph theory and, as the name suggests, works by storing a matrix with rows and columns that are indicative of each cell. The matrix is then filled with a one in the position of a certain column and row if the cell that the column represents is adjacent to the cell the row represents (or the other way around depending on convention)[86]. This matrix is therefore a square matrix of a large size due to the desire and requirement to use a large number of control volumes. As the matrix is usually sparse, there are methods to store the matrix for a lot less computational resources then a normal matrix. Every time the grid is then adapted, the matrix has to be recalculated, and the matrix must be adjusted in a manner that is not independent of the refinement of a particular cell (meaning that this approach is not parallelisable). In general, for modern methods, this paradigm is insufficient as it requires a significantly large amount of storage for the large number of control volumes desired, as well as requires a large amount of processing time to consistently adjust the matrix. An example of a computational grid and its corresponding adjacency matrix is shown below in Figure 5.



$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Figure 5: An example of the adjacency matrix approach to AGM. Each cell in the computational domain is given a particular number from one to seven to uniquely identify the cell. It is clear that the matrix is equal to its transpose, as every adjacent cell is adjacent to the reference cell[86].

The tree structure is ordered with a set root cell that is the entire system. The cell is then subdivided in some manner to produce children cells which are stored in the tree as child nodes of the root cell. This is performed iteratively to hold all of the cells in a tree-structure. Below in Figure 6, is an example of a tree structure representation of a domain. The adjacency of a cell is then found by using a tree-searching algorithm. The advantage of this method is that holding the cells in a tree structure requires considerably less computational resources than the adjacency matrix approach and can be adjusted as required as the grid is refined or unrefined. Further it is independent of the refinements of other cells meaning that this approach is parallelisable. The disadvantage is that this approach spends more time finding neighbouring cells using tree searching algorithms.

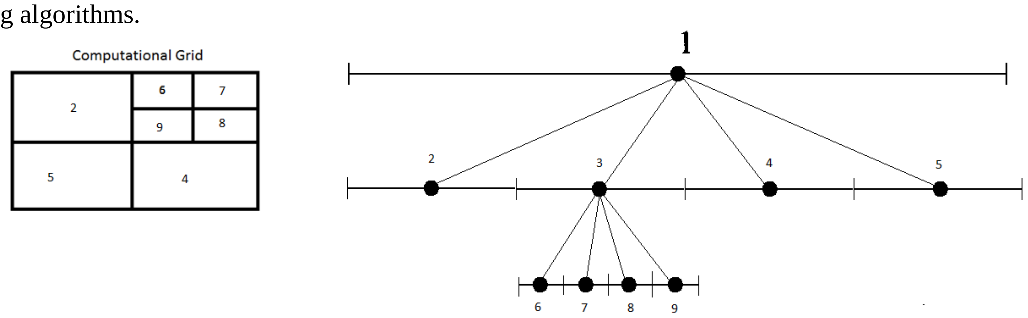


Figure 6: An example of a tree structure for a particular computational grid. In the case presented, the adjacency matrix structure is far superior to the tree structure. However, in this example there are only three levels of refinement. Most computational grids have a level of refinement larger than eight, leading to a much more complex grid[86].

The FTT structure is a sort of intermediary of the two structures. It is a tree structure where each node also holds location information about adjacent siblings and its parent cell (and sometimes the neighbouring cells of the parent depending on convention and approach). This is shown below in Figure 7, where the smaller directed arrows represent the internal location information. This method requires slightly more storage resources than the traditional tree structure but still significantly less than the adjacency matrix paradigm. Further, this method allows for significantly faster tree searching algorithms[55]. Many FTT systems will enforce that an external node cannot have adjacent external nodes which differ by more than one level. The application of this restriction is shown in Figure 8.

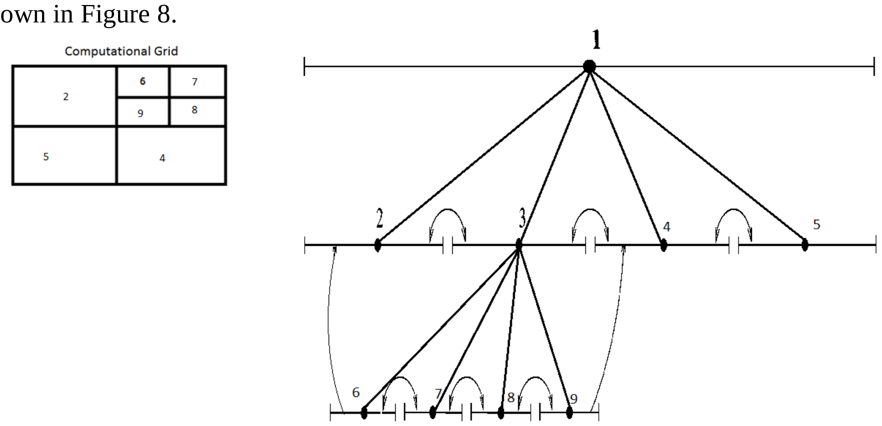


Figure 7: A representation of the FTT structure for a simple computational domain. The strong black lines represent the tree structure, and the directed arrows indicate the internal locative information. Each neighbouring cell knows the location of the adjacent siblings and adjacent parental siblings. This additional structure allows for a significantly more rapid searching algorithm[55].

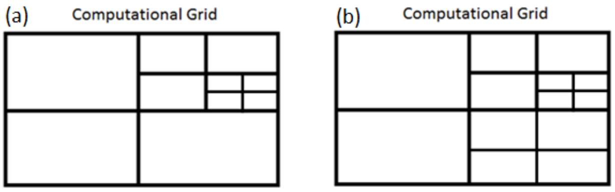


Figure 8: An example of the restriction of adjacent cells generation. Subfigure (a) shows a gridding that disregards the generational requirement. This can be seen because the bottom right cell has a refinement of 2 but is adjacent to a cell of refinement 4. Subfigure (b) shows how this is conventionally fixed by refining the bottom

right cell to a refinement level of 3[86].

One of the principles of FTT, as discussed previously, is that adjacent cells must not differ in the tree by more than one generation. This is required in the original structure to give an upper bound on the number of search operations required to find adjacent cells in the tree. In other words, the benefit of this requirement is that less time is spent locating the position of adjacent cells, at the cost of requiring additional cells.

In the case of combustion modelling, each additional cell causes the use of additional computational resources in terms of both storage and processing time. Therefore, this requirement – once the impact on the other components of the model has been determined – has the benefit of decreased location times, at the cost of increased iteration times and increased storage requirements.

Whilst the increase in storage requirements has little impact, the increase in iteration time is significant. The relaxation of this requirement leads to grids that have only the refinement that is necessary, not a refinement that minimises location time. This report investigated the impact of this relaxation.

6.3.2. Grid refinement

AGM works by refining and un-refining the computational grid in an iterative manner which is reflective of the numerical solution. In the FTT structure, when the grid is refined the selected nodes are amended with children cells. For a uniform splitting, the number of children cells is always two to the power of the dimension to be modelled. The state vector at the children cells can be defined in a few different ways depending on the structure of the problem to be solved. Because the variables in the governing equations should be preserved, many interpolation techniques cannot be used. For simplicity, the children cells are taken to be precisely the value of the parental cell.

When the grid is un-refined the selected nodes are removed and the parental cell is set to be equal to the average of the children cells, to preserve the overall sum of the state vector quantities. It should be noted that at any one time step, the computational grid is likely to be simultaneously refining and un-refining across the system.

6.3.3. Refinement indicators

Knowing how the grid is stored, and how the grid is refined/un-refined, is important, but is not complete without understanding when the grid should be refined or unrefined. This is done through the use of refinement indicators. These indicators are strongly dependent on the problem that is to be solved. In the case of combustion modelling, there are three widely considered areas of importance: 1) areas with large differences to adjacent cells (suggesting the grid is insufficiently refined); 2) areas with contact discontinuities; 3) areas with shock waves. It should be noted that if the first concern is sufficiently resolved the other two concerns are enforced; however, resolving the general case of the first concern often leads to an over resolution in non-critical areas. Historically, only the second and third considerations were deemed critical; but, the majority of contemporary research engages a lot of effort in the first concern as a more general indicator of resolution[55].

The first indicator, the general indicator ε^g, is expressed mathematically below for some cell n, with the set of adjacent cells A_n (where $A_n(j)$ represents the j^{th} neighbour of cell n), and a state vector dimension of V, where the argument of the state vector denotes the component of the state vector[55]:

$$\varepsilon_n^g = \max_{j \leq \|A_n\|}\left(\max_{k=1\ldots V}\left(\frac{\left|U_{A(j)}(k) - U_n(k)\right|}{\max\left(\left|U_{A(j)}(k)\right|, \left|U_n(k)\right|\right)}\right)\right) \textbf{(67)}$$

One should note that this definition produces the largest percentage difference in any of the state vector's components of the current cells and that of any of the neighbouring cells.

The second indicator, the discontinuity indicator ε^d, is expressed below where P_k is the pressure at cell k, ρ_k is the density at cell k, and both ε_d and ε_s are threshold values (both of which are generally selected to be 0.2)[55].

$$\varepsilon_n^d = \max_{j \leq \|A_n\|}\left(\begin{cases} 1, \text{ if } \dfrac{\left|P_{A(j)} - P_n\right|}{\min\left(\left|P_{A(j)}\right|, \left|P_n\right|\right)} > \varepsilon_s; \dfrac{\left|\rho_{A(j)} - \rho_n\right|}{\min\left(\left|\rho_{A(j)}\right|, \left|\rho_n\right|\right)} > \varepsilon_d \\ 0, \text{ otherwise} \end{cases}\right) \textbf{(68)}$$

The tertiary indicator, the shock indicator ε^s, is expressed below [55]:

$$\varepsilon_n^s = \max_{j \leq \|A_n\|}\left(\begin{cases} 1, \text{ if } \dfrac{\left|P_{A(j)} - P_n\right|}{\min\left(\left|P_{A(j)}\right|, \left|P_n\right|\right)} > \varepsilon_s; \delta_j \cdot (u_n - u_k) \\ 0, \text{ otherwise} \end{cases}\right) \textbf{(69)}$$

where δ_j and k_j are defined as[55]:

$$\delta_j = 1 - 2\text{mod}(j, 2);\ k_j = \text{floor}\left(\frac{j+1}{2}\right) \textbf{(70)}$$

The total refinement indicator is then defined as below[55]:

$$\varepsilon_n^T = \max_q\left(\varepsilon_n^q\right) \qquad \textbf{(71)}$$

It is sometimes the case that these indicator values lie around refinement and un-refinement thresholds, which leads to the mesh cycling between refining and un-refining which can spasmodically produce errors. To alleviate this, the total refinement indicator is then smoothed. The method of smoothing depends on the type of problem to be solved and in general should reflect the physical problem in some simple manner. In the case of CFD and generalised combustion modelling, this is usually done by analogy to the steady-state differential equation (whose temporal version is shown below) of a simple diffusion-reaction model (note the lack of advection terms)[55]:

$$\frac{\partial \varepsilon^T}{\partial t} = K\nabla^2\varepsilon^T + Q\textbf{(72)}$$

where K is defined to be a constant rate of diffusion dependent on the level of refinement, and the reaction term, Q, is defined the same way ε_n will be, once ε^T is smoothed – given for this model in equation (75). Naturally, it is desirable to directly calculate the steady-state solution. This is shown below[55]:

$$-S_{\varepsilon^T}\frac{\partial \varepsilon^T}{\partial x} = \frac{\partial^2 \varepsilon^T}{\partial x^2} + Q\textbf{(73)}$$

where S is the constant speed of diffusion. This solution represents a reaction-diffusion front that expands, and smooths out isolated values of ε^T which are arbitrarily oscillating near threshold values.

The refinement indicator is then defined as below, where ε_r and ε_u are refinement and un-refinement thresholds generally taken to be 0.7 and 0.3 respectively. If the indicator is 1, the cell is refined, if the indicator of the cell is zero the cell is left alone, and if all the children of a parent node are leaves and have an indicator of -1 then the children cells are unrefined.

$$\varepsilon_n = \begin{cases} 1 \text{ if} \varepsilon_n^t > \varepsilon_r \\ 0 \text{ otherwise } \textbf{(74)} \\ -1 \text{ if} \varepsilon_n^t < \varepsilon_u \end{cases}$$

6.3.4. Iterations across the grid

On a uniform grid, the theoretical concerns for stability have already been discussed in section 0 and led to the expression (51). In the case of a non-uniform grid, the theoretical concerns of stability still apply locally. This means that locally, for a specified cell, *i*, at time step *n*, that has been refined *l* times, with a system length *L*, the stability requirement becomes[55]:

$$\Delta t_i^n = \mathrm{cfl}\frac{2^{-l}L}{s_i^n} \qquad \textbf{(75)}$$

Attempting to allow each cell to have a different time step becomes troublesome. At this point, two different paradigms occur: 1) take each time step globally to be the smallest time step or 2) allow each cell to have a different time step depending on the level requirement (meaning each cell on the same level has the same time step). The first paradigm allows for a simple model at the cost of requiring an increased number of calculations (and therefore processing time). The second paradigm allows for the following expression which ensures that the scheme is both locally stable (for an ε neighbourhood of a control volume including at least one neighbouring control volume) and globally stable (across the entire computational grid)[55].

$$\Delta t^n(l) = 2^{l_{min}-l}\Delta t_{\mathrm{Global}}^n \qquad \textbf{(76)}$$

where l_{min} is the level of refinement of the least refined cell, and $\Delta t_{\mathrm{Global}}^n$, is the global *n*th time step. This is defined by:

$$\Delta t_{\mathrm{Global}}^n = \mathrm{cfl}\frac{2^{-l_{min}}L}{s_i^n} \qquad \textbf{(77)}$$

After one successive iteration of the scheme, $\Delta t_{\mathrm{Global}}^n$ time has passed. If *A(l)* is the advancement procedure of a cell (for example the FLIC scheme discussed in section 5.4 applied only to the set of cells that have a level of refinement *l*, by half a time step (note the implied use of the fractional time step method) then the total procedure is given recursively by:

$$S(l) = \begin{cases} R(l+1)A(l)S(l+1)A^{\dagger}(l)S(l+1)\text{if} l < l_{\max} \\ A(l)A^{\dagger}(l)\text{ifif} l = l_{\max} \end{cases} \qquad \textbf{(78)}$$

The global procedure to advance the entire system by the global time step is then called by *S(l*$_{\min}$*)*, where $l_{\max}$ is the maximum level of refinement currently on the grid, and *R(l)* is the refinement procedure for the cells with a level of refinement of *l*. Note, the transpose of an operator is taken to be spatial and temporal reconfiguration as required by the fractional step method.

One should note the similarity to (and significant inspiration from) the procedure given in[55];

however, the subtle difference in the location and argument of the refinement procedure is critical due to the changes in the refinement structure.

6.4. Turbulence indicator

Turbulence is one of the critical components to CFD, and is therefore a large concern for this combustion model. This is particularly true as the model must be able to simultaneously handle high and low turbulence areas, as well as areas that change from flows where viscous effects dominate the regime to areas where the effects of viscosity are negligible. It is therefore important for the model to be able to identify critical areas of turbulence. To do this, the model calculates the magnitude of the vorticity vector at each cell. Cells that have a significantly different vorticity vector from nearby cells must be insufficiently refined. Therefore, the following vorticity indicator is defined, to be able to locate and identify areas where turbulence is not resolved. Equation (79) below, describes a discretised version of the vorticity vector (the curl of the velocity vector) using a forward differencing approximation. Equation (80) depicts the vorticity indicator. Whilst the mathematical description is quite cumbersome the intuitive concept is that it is equal to the largest percentage difference of the vorticity field in the cell *n* by considering the difference in the forward and backward first order differencing schemes. Note that the first “max” in equation (80) has a column argument structure to format it into one line.

$$\Omega(n,1,2,3) = \left(\frac{((w_3-w_n)\Delta z-(v_2-v_n)\Delta y)}{\Delta z\Delta y}, \quad \frac{((u_1-u_n)\Delta x-(w_3-w_n)\Delta z)}{\Delta z\Delta x}, \quad \frac{((v_2-v_n)\Delta y-(u_1-u_n)\Delta x)}{\Delta x\Delta y}\right) \tag{79}$$

$$\varepsilon_n^V = \max_{k=0\ldots 2^V-1}\begin{pmatrix} \left(\dfrac{\left\|\Omega\left(n,(-1)^k x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,z\right)-\Omega\left(n,(-1)^k x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,-z\right)\right\|}{\max\left(\left\|\Omega\left(n,(-1)^k x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,z\right)\right\|,\left\|\Omega\left(n,(-1)^k x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,-z\right)\right\|\right)}\right) \\ \left(\dfrac{\left\|\Omega\left(n,(-1)^k x,y,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}z\right)-\Omega\left(n,(-1)^k x,-y,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}z\right)\right\|}{\max\left(\left\|\Omega\left(n,(-1)^k x,y,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}z\right)\right\|,\left\|\Omega\left(n,(-1)^k x,-y,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}z\right)\right\|\right)}\right) \\ \left(\dfrac{\left\|\Omega\left(n,x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,(-1)^k z\right)-\Omega\left(n,-x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,(-1)^k z\right)\right\|}{\max\left(\left\|\Omega\left(n,x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,(-1)^k z\right)\right\|,\left\|\Omega\left(n,-x,(-1)^{\left\lfloor\frac{k}{2}\right\rfloor}y,-(-1)^k z\right)\right\|\right)}\right) \end{pmatrix} \tag{80}$$

where x is the adjacent cell in the x direction, y is the adjacent cell in the y direction and z is the adjacent cell in the z direction. For some cell *n* that the indicator is designated for, *V* is the number of spatial dimensions and ε_n^V represents the vorticity indicator.

VII. CONCLUDING REMARKS

This article details the process of the construction methodology of a theoretical model of a gaseous explosion. A theoretical model on gas explosion can be developed from the iterative application of concurrent mass balances, momentum balances, energy balances and species balances.

The theoretical modelling of gas explosion includes several sub-models including reaction modelling, thermodynamic modelling and fluid dynamic modelling. Each of these models raises a number of questions and a researcher needs to decide the optimum conditions for these sub-models. These optimum conditions finally determine the computational expense required for the theoretical modelling of the gaseous explosion. These have been discussed in detail in the article and are expected to assist researchers in their development of theoretical modelling.

ACKNOWLEDGEMENT

This work was supported by the Department of Resources Energy and Tourism, Australia (grant number – G1201029) and ACA Low Emissions Technologies Ltd., Australia (grant number – G1400523).

REFERENCES

[1]. Moore M, The mine explosion at Johnstown, J. Franklin Inst. 158 (1904) 83-96.
[2]. Dhillon BS, Mine safety: a modern approach, Springer Science & Business Media, 2010.
[3]. Dingsdag D, The Bulli Mining Disaster 1887: Lessons from the Past, St Louis Press, Darling Heights, 1993.
[4]. Eckhoff R, Explosion Hazards in the Process Industries, Gulf Publishing Company, Houston, Texax, 2005.
[5]. Markham CW, Mines Rescue Bill,
[6]. https://www.parliament.nsw.gov.au/prod/parlment/hansart.nsf/V3Key/LA19940412038 (Accessed 27 January 2016).Organ MK,

Bulli Mine disaster centenary, Illawarra Historical Society Bulletin September (1987) 54-56.

[7]. Dolan J, 2 homes leveled, 15 damaged in Paterson gas explosion, ABC Inc., WABC-TV New York, http://abc7ny.com/news/2-homes-leveled-15-damaged-in-paterson-gas-explosion/1538285/, Accessed: 19 Oct 2016.

[8]. Metcalfe J, How a Simple Gas Leak Could Destroy Two Entire Buildings, Citylab website, http://www.citylab.com/work/2014/03/how-gas-leak-disasters-happen/8627/, Accessed: 19 Oct 2016.

[9]. Einiger J, Stafford Township house explosion blamed on underground gas leak, ABC Inc., WABC-TV New York, http://abc7ny.com/news/stafford-township-house-explosion-blamed-on-underground-gas-leak/532311/, Accessed: 19 Oct 2016.

[10]. Newcomb T, How a Gas Leak Can Destroy a Building, Popular Mechanics website, http://www.popularmechanics.com/technology/infrastructure/a10244/how-a-gas-leak-can-destroy-a-building-16585780/, Accessed: 19 Oct 2016.

[11]. Kemsley J, Spark from pressure gauge caused University of Hawaii explosion, fire department says, Chemical and Engineering News, American Chemical Society, http://cen.acs.org/articles/94/web/2016/04/Spark-pressure-gauge-caused-University.html, Accessed: 05 Apr 2017.

[12]. Radford J, "What happened tragic day of Mt Kembla Mine disaster" (Illawarra Mercury website, http://www.illawarramercury.com.au/story/2441786/what-happened-tragic-day-of-mt-kembla-mine-disaster/), Accessed: 14 August 2015.

[13]. Block R, "Locomotive crushes 105 gold miners" (website, http://www.independent.co.uk/news/world/locomotive-crushes-105-gold-miners-1619145.html) (Accessed: 05 April 2017).

[14]. Kundu S, Zanganeh J, Moghtaderi B, A review on understanding explosions from methane–air mixture, J. Loss Prevent. Process Ind. 40 (2016) 507-23.

[15]. Schröder V, Holtappels K. Explosion characteristics of hydrogen-air and hydrogen-oxygen mixtures at elevated pressures. In. Explosion characteristics of hydrogen-air and hydrogen-oxygen mixtures at elevated pressures. 2005.

[16]. Lin B, Ye Q, Zhai C, Jian C, The propagation rule of methane explosion in bifurcation duct, J. China Coal Soc. 33 (2008) 136-39.

[17]. Yi Y, Xueqiu H, Geng L, Huia W, Effect of meshy obstacle on methane gas explosion, Procedia Eng. 26 (2011) 70-74.

[18]. Salzano E, Cammarota F, Di Benedetto A, Di Sarli V, Explosion behavior of hydrogen–methane/air mixtures, J. Loss Prevent. Process Ind. 25 (2012) 443-47.

[19]. Bone WA, Drugman J, LXXI.—The explosive combustion of hydrocarbons, J. Chem. Soc., Trans. 89 (1906) 660-82.

[20]. Cooper C, Wiezevich P, Effects of temperature and pressure on the upper explosive limit of methane-oxygen mixtures, Ind. Eng. Chem. 21 (1929) 1210-14.

[21]. Bowden F, Yoffe A. Hot spots and the initiation of explosion. In. Hot spots and the initiation of explosion. Elsevier, 1948, pp. 551-60.

[22]. Babkin V, Kozachenko L, Study of normal burning velocity in methane-air mixtures at high pressures, Combust. Explo. Shock Wave. 2 (1966) 46-52.

[23]. Hjertager BH, Bjørkhaug M, Fuhre K, Explosion propagation of non-homogeneous methane—air clouds inside an obstructed 50 m^3 vented vessel, J. Hazard. Mater. 19 (1988) 139-53.

[24]. Gieras M, Klemens R, Rarata G, Wolański P, Determination of explosion parameters of methane-air mixtures in the chamber of 40dm^3 at normal and elevated temperature, J. Loss Prevent. Process Ind. 19 (2006) 263-70.

[25]. Suzuki Y, Sarachik M, Chudnovsky E, McHugh S, Gonzalez-Rubio R, Avraham N, Myasoedov Y, Zeldov E, Shtrikman H, Chakov N, Propagation of Avalanches in Mn$_{12}$-Acetate: Magnetic Deflagration, Phys. Rev. Lett. 95 (2005) 147201.

[26]. Kuznetsov M, Alekseev V, Yankin Y, Dorofeev S, Slow and fast deflagrations in hydrocarbon-air mixtures, Combust. Sci. Technol. 174 (2002) 157-72.

[27]. Kessler D, Gamezo V, Oran E, Simulations of flame acceleration and deflagration-to-detonation transitions in methane–air systems, Combust. Flame 157 (2010) 2063-77.

[28]. Chapman DL, On the rate of explosion in gases, London Edinburgh Dublin Philos. Mag. J. Sci. 47 (1899) 90-104.

[29]. Jouguet JCE, Sur la propagation des réactions chimiques dans les gaz, Journal des Mathématiques Pures et Appliquées 1 (1905) 347–425.

[30]. Lee JH, The detonation phenomenon, Cambridge University Press Cambridge, 2008.

[31]. Chan C, Dewit W. Deflagration-to-detonation transition in end gases. In. Deflagration-to-detonation transition in end gases. Elsevier, 1996, pp. 2679-84.

[32]. Hasson A, Avinor M, Burcat A, Transition from deflagration to detonation, spark ignition, and detonation characteristics of ethylene-oxygen mixtures in a tube, Combust. Flame 49 (1983) 13-26.

[33]. Sorin R, Zitoun R, Desbordes D, Optimization of the deflagration to detonation transition: reduction of length and time of transition, Shock Waves 15 (2006) 137-45.

[34]. Sharma R, Gurjar B, Wate S, Ghuge S, Agrawal R, Assessment of an accidental vapour cloud explosion: Lessons from the Indian Oil Corporation Ltd. accident at Jaipur, India, J. Loss Prevent. Process Ind. 26 (2013) 82-90.

[35]. Brinkley S, Lewis B. On the transition from deflagration to detonation. In. On the transition from deflagration to detonation. Elsevier, 1958, pp. 807-11.

[36]. Liberman M, Kiverin A, Ivanov M, On detonation initiation by a temperature gradient for a detailed chemical reaction models, Phys. Lett. A 375 (2011) 1803-08.

[37]. Hjertager BH, Computer modelling of turbulent gas explosions in complex 2D and 3D geometries, J. Hazard. Mater. 34 (1993) 173-97.

[38]. Arntzen BJ. Modelling of turbulence and combustion for simulation of gas explosions in complex geometries. 1998.

[39]. Venetsanos A, Baraldi D, Adams P, Heggem P, Wilkening H, CFD modelling of hydrogen release, dispersion and combustion for automotive scenarios, J. Loss Prevent. Process Ind. 21 (2008) 162-84.

[40]. Makarov D, Verbecke F, Molkov V, Roe O, Skotenne M, Kotchourko A, Lelyakin A, Yanez J, Hansen O, Middha P, An inter-comparison exercise on CFD model capabilities to predict a hydrogen explosion in a simulated vehicle refuelling environment, Int. J. Hydrogen Energ. 34 (2009) 2800-14.

[41]. Park DJ, Lee YS, A comparison on predictive models of gas explosions, Korean J. Chem. Eng. 26 (2009) 313.

[42]. Bi M, Dong C, Zhou Y, Numerical simulation of premixed methane–air deflagration in large L/D closed pipes, Appl. Therm. Eng. 40 (2012) 337-42.

[43]. Li Z, Yao T, Simulation Study on Leakage and Dispersion of High Sulfur Natural Gas Pipeline with FLUENT CFD-based

Software, Oil Gas Stor. Transport. 5 (2008) 013.

[44]. Staša P, Fuchsíková P, Kebo V, Kubáč L. Modelling potential action for building protection against flow of methane from the underground using the Fluent software. In. Modelling potential action for building protection against flow of methane from the underground using the Fluent software. IEEE, 2012, pp. 677-83.

[45]. Hansen O, Renoult J, Sherman M, Tieszen S. Validation of FLACS-hydrogen CFD consequence prediction model against large scale H_2 explosion experiments in the flame facility. In. Validation of FLACS-hydrogen CFD consequence prediction model against large scale H_2 explosion experiments in the flame facility. 2005.

[46]. Yet-Pole I, Chiu Y-L, Wu S-J, The simulation of air recirculation and fire/explosion phenomena within a semiconductor factory, J. Hazard. Mater. 163 (2009) 1040-51.

[47]. Li J, Ma G, Abdel-jawad M, Hao H, Evaluation of gas explosion overpressures at configurations with irregularly arranged obstacles, J. Perform. Constr. Fac. 29 (2014) B4014003.

[48]. Oran ES, Gamezo VN, Kessler DA. Deflagrations, Detonations, and the Deflagration-to-Detonation Transition in Methane-Air Mixtures. In. Deflagrations, Detonations, and the Deflagration-to-Detonation Transition in Methane-Air Mixtures. DTIC Document, 2011.

[49]. Launder BE, Spalding DB, The numerical computation of turbulent flows, Comput. Method. Appl. Mech. Eng. 3 (1974) 269-89.

[50]. Toro EF, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.

[51]. Marra FS, Continillo G. Numerical study of premixed laminar flame propagation in a closed tube with a full Navier-Stokes approach. In. Numerical study of premixed laminar flame propagation in a closed tube with a full Navier-Stokes approach. Elsevier, 1996, pp. 907-13.

[52]. Anderson JD, Wendt J, Computational fluid dynamics, Springer, 1995.

[53]. Eu BC, Generalized Thermodynamics: Thermodynamics of Irreversible Processes and Generalized Hydrodynamics, Springer Science & Business Media, 2002.

[54]. LeVeque RJ, Finite volume methods for hyperbolic problems, Cambridge university press, 2002.

[55]. Khokhlov AM, Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations, J. Comput. Phys. 143 (1998) 519-43.

[56]. Hänggi P, Talkner P, Borkovec M, Reaction-rate theory: fifty years after Kramers, Rev. Mod. Phy. 62 (1990) 251.

[57]. Bauer T, Lunkenheimer P, Loidl A, Cooperativity and the Freezing of Molecular Motion at the Glass Transition, Phys. Rev. Lett. 111 (2013) 225702.

[58]. Steinfeld JI, Francisco JS, Hase WL, Chemical kinetics and dynamics, Prentice Hall Englewood Cliffs (New Jersey), 1989.

[59]. Fickett W, Wood WW, Flow Calculations for Pulsating One-Dimensional Detonations, Phys. Fluids 9 (1966) 903-16.

[60]. Arima T, Numerical methods for chemically reacting fluid flow computation under low-Mach number approximation, Tokyo J. Math. 29 (2006) 167-98.

[61]. Drysdale D. Physics and chemistry of fire. In: Cote AE editor. Physics and chemistry of fire. Quincy, MA: National Fire Protection Association, 2008, pp. 2-18.

[62]. Williams FA, Combustion Theory: the fundamental theory of chemical reacting flow systems, Perseus Book, Massachusetts, 1985.

[63]. Spencer LF, Gallimore AD, Efficiency of CO_2 dissociation in a radio-frequency discharge, Plasma Chem. Plasma Process. 31 (2011) 79-89.

[64]. Lysenko D, Solomatnikov A, Numerical Modeling of Turbulent Heat Exchange in the Combustion Chambers of Gas-Turbine Plants with the Use of the Fluent Package, J. Eng. Phys. Thermophys. 76 (2003) 888-91.

[65]. Connors KA, Chemical kinetics: the study of reaction rates in solution, John Wiley & Sons, 1990.

[66]. Ebert KH, Deuflhard P, Jäger W, Modelling of Chemical Reaction Systems, Springer Science & Business Media, 2012.

[67]. Miranker A, Numerical Methods for Stiff Equations and Singular Perturbation Problems: and Singular Perturbation Problems, Springer Science & Business Media, 2001.

[68]. Helzel C. Numerical Approximation of Conversation Laws with Stiff Source Term for the Modelling of Detonation Waves, PhD Thesis. 2000.

[69]. Smith GD, Numerical solution of partial differential equations: finite difference methods, Oxford university press, 1985.

[70]. Strang G, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506-17.

[71]. Sanz-Serna JM, Spijker MN, Regions of stability, equivalence theorems and the Courant-Friedrichs-Lewy condition, Numerische Mathematik 49 (1986) 319-29.

[72]. Gourlay MJ, Fluid Simulation for Video Games (part 2), Intel online articles, website: https://software.intel.com/en-us/articles/fluid-simulation-for-video-games-part-2 , Accessed: 01 May 2017.

[73]. Vågsæther K. Modelling of gas explosions. PhD Thesis. Faculty of Engineering Science & Technology, Norwegian University of Science and Technology, 2010.

[74]. Yee H, Warming R, Harten A, Implicit total variation diminishing (TVD) schemes for steady-state calculations, J. Comput. Phys. 57 (1985) 327-60.

[75]. Shu C-W, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073-84.

[76]. Gottlieb S, Shu C-W, Total variation diminishing Runge-Kutta schemes, Math. Comput. - Am. Math. Soc. 67 (1998) 73-85.

[77]. Knuth DE, Big omicron and big omega and big theta, ACM Sigact News 8 (1976) 18-24.

[78]. Rosen KH, Discrete mathematics and its applications, McGraw-Hill Higher Education, 2007.

[79]. Papadimitriou CH, Computational complexity, John Wiley and Sons Ltd., 2003.

[80]. Kuzmin D, Turek S, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter, J. Comput. Phys. 198 (2004) 131-58.

[81]. Bertoluzza S, Falletta S, Russo G, Shu CW, Numerical Solutions of Partial Differential Equations, Birkhäuser Basel, 2009.

[82]. Fox GC, Williams RD, Messina GC, Parallel computing works!, Morgan Kaufmann, San Francisco, USA, 2014.

[83]. Rünger G. Parallel programming models for irregular algorithms. In. Parallel programming models for irregular algorithms. Springer, 2006, pp. 3-23.

[84]. "User's Guide, Parallel Computing Toolbox" (Mathworks website, http://www.mathworks.com/help/pdf_doc/distcomp/distcomp.pdf) (Accessed: 09 May 2017).

[85]. Goldberg D, What every computer scientist should know about floating-point arithmetic, ACM Computing Surveys (CSUR) 23 (1991) 5-48.

[86]. Plewa T, Linde T, Weirs VG. Adaptive mesh refinement-theory and applications. In. Adaptive mesh refinement-theory and applications. Springer, 2003.

[87]. Berger MJ, Oliger J, Adaptive mesh refinement for hyperbolic partial differential equations, J. Computat. Phys. 53 (1984) 484-512.

[88]. Bell J, Berger M, Saltzman J, Welcome M, Three-dimensional adaptive mesh refinement for hyperbolic conservation laws, SIAM J. Sci. Comput. 15 (1994) 127-38.

[89]. Berger MJ, Colella P, Local adaptive mesh refinement for shock hydrodynamics, J. Computat. Phys. 82 (1989) 64-84.

[90]. Ishida T, Takahashi S, Nakahashi K, Efficient and robust cartesian mesh generation for building-cube method, J. Comput. Sci. Technol. 2 (2008) 435-46.

[91]. Yajun Y, Jianxin Z, Dunming L, Shengyong P, Xu S, Phase-field simulation of dendritic solidiication using a full threaded tree with adaptive meshing, China Foundry 11 (2014)

[92]. Ishida T, Takahashi S, Nakahashi K. Fast cartesian mesh generation for building-cube method using multi-core pc. In. Fast cartesian mesh generation for building-cube method using multi-core pc. 2008, pp. 919.

[93]. Ji H, Lien F-S, Zhang F, A GPU-accelerated adaptive mesh refinement for immersed boundary methods, Comput. Fluids 118 (2015) 131-47.

Joshua Williams.The Fundamentals of Theoretical Modelling Of Gas Explosion – A Review." American Journal of Engineering Research (AJER), vol. 7, no. 3, 2018, pp.173-199.