International Conference of Mechanical Engineering Research (ICMER) 2011
5-7 Dec, Malaysia, Paper ID: ICMER2011-151

Implementation of Conditional Moment Closure using the Taylor Expansion and Finite Differences

M.M.Noor[1,3], A.Aziz Hairuddin[1,4], Andrew P.Wandel[1] and T.F.Yusaf[2,3]
[1]Computational Engineering and Science Research Centre, Department of Mechanical and Mechatronic, Engineering University of Southern Queensland (USQ),
[2]National Centre for Engineering in Agriculture, University of Southern Queensland (USQ), Australia
[3]Faculty of Mechanical Engineering, Universiti Malaysia Pahang (UMP), Malaysia.
[4]Faculty of Engineering, Universiti Putra Malaysia (UPM), Malaysia.
E-mail: Muhamad.MatNoor@usq.edu.au / muhamad@ump.edu.my

Abstract. Current stringent rules for pollution release to the atmosphere have led to researchers worldwide developing methods to increase the efficiency of combustion. In order to reduce experimental cost, accurate modelling and simulation is a very critical step. One combustion model is Conditional Moment Closure (CMC). This paper reports a preliminary result implementing the CMC model using a finite difference method. The Taylor series expansion was utilised to determine the error term for the discretization. FORTRAN code was used to simulate the discretized partial differential equation. From the simulation result, it is found that the explicit method is simple but less accurate than the implicit method.

1. Introduction

The desire to improve combustion efficiency with lower emissions has led to an increased interest in combustion modelling and research. Turbulent combustion is divided into two classes depending on when the fuel is mixed with the oxidizer: premixed and non-premixed. For non-premixed combustion, there are types of modelling methods for the mixing process: probability density function (PDF) based model [1-3], flamelet model [4-7], fast chemistry limit model [8], and mapping closure model [9, 10]. Besides those models, there are a few other developments such as conditional moment closure (CMC) by Klimenko and Bilger [11-13] and multiple mapping conditioning (MMC) by Klimenko and Pope [14], with further developments [15-18] for the mixing process.

The CMC concept is a model for turbulent combustion processes describing the transport of reactive scalars in conserved scalar spaces. The key innovation in the introduction of CMC was to take into account conditional averages in the combustion modelling because the chemical source term is not a function of unconditional averages, which are used in conventional modeling. Ternat et al [19] computed stable solutions using two finite difference methods, namely the Euler method and the Crank–Nicolson method, to advance the solution of the heat equation in time. Clarke et al [20] used Direct Numerical Simulation (DNS) results to model the parameters of CMC for combustion systems with droplets. This is a complicated process because of the interaction of the evaporating liquid with the gaseous phase; the usage of a spark to evaporate the fuel and ignite the mixture exacerbates this complexity in comparison with auto ignition.

A mixing model is required to close the molecular diffusion term in the PDF transport equation [1], which is the last term in equation (1) and contains the molecular diffusion flux vector (J_{ik}):

$$\frac{\partial \rho P}{\partial t}+\frac{\partial \rho u_i P}{\partial x_i}+\frac{\partial \rho S_k P}{\partial \psi_k}=-\frac{\partial}{\partial x_i}\left[\rho\langle u_i^{"}|\psi\rangle P\right]+\frac{\partial}{\partial \psi_k}\left[\frac{1}{\rho}\langle\frac{\partial J_{i,k}}{\partial x_i}|\psi\rangle P\right] \quad (1)$$

Mixing plays a crucial role in the non-premixed combustion process; a number of mixing models have been developed. The Coalescence and Dispersion process was modelled by Curl in 1963 [21] and Levenspiel and Spielman in 1965 [22] and is often called "Curl's model". The governing equation for Curl's model is shown in equation (2). The left part of the equation is the particle composition before mixing. This particle then Coalescences with another particle and mixing occurs. The mixing process can be referred to at the middle part of equation (2) and then this mixed particle will be dispersed as shown in the last part of equation (2).

$$\begin{pmatrix}(\phi_{A1},\phi_{B1})_1\\(\phi_{A2},\phi_{B2})_2\end{pmatrix}\rightarrow\begin{pmatrix}\phi_A^*=(\phi_{A1}+\phi_{A2})/2\\\phi_A^*=(\phi_{B1}+\phi_{B2})/2\end{pmatrix}\rightarrow\begin{pmatrix}(\phi_A^*,\phi_B^*)_1\\(\phi_A^*,\phi_B^*)_2\end{pmatrix} \quad (2)$$

Here ϕ_{A1} is the composition of species *A* for particle 1, ϕ_A^* is the new composition of species *A* and (ϕ_A,ϕ_B) is a particle which consists of species *A* and *B*. The Curl's model was modified by Janicka et. al [23] and Dopazo [24] in 1979. This model was called modified Curl's (MC) model (equations (3) and (4)) where β can take any value from 0 to 1 and can be a random variable. If $\beta = 0$, then no mixing occurs, whilst $\beta = 1$ reproduces Curl's model.

$$\phi_i^{MC}(t+\delta t)=(1-\beta t)\phi_i^{MC}(t)+\frac{1}{2}\beta\left(\phi_j^{MC}(t)+\phi_i^{MC}(t)\right) \quad (3)$$

$$\phi_j^{MC}(t+\delta t)=(1-\beta t)\phi_j^{MC}(t)+\frac{1}{2}\beta\left(\phi_j^{MC}(t)+\phi_i^{MC}(t)\right) \quad (4)$$

Here ϕ_i and ϕ_j is the composition of particles *i* and *j*, *t* is time and δt is the time interval.

The weakness of the MC model is that this mixing process does not enforce the requirement that only particles that are close to each other are allowed to mix and interact with each other. This issue was solved by the Euclidean Minimum Spanning Tree (EMST) model [25] whereby particles that mix are close together in composition space as shown in equations (5) and (6). In these equations, there are two new constants introduced where d is determined so that the desired amount of mixing is obtained and P_b is the position of particle in the EMST branch. Particles near to the centre will have higher P_b values.

$$\phi_i^{EMST}(t+\delta t)=\phi_i^{EMST}(t)+dP_b\delta t\left(\phi_j^{EMST}(t)+\phi_i^{EMST}(t)\right) \quad (5)$$

$$\phi_j^{EMST}(t+\delta t)=\phi_j^{EMST}(t)+dP_b\delta t\left(\phi_i^{EMST}(t)+\phi_j^{EMST}(t)\right) \quad (6)$$

Another mixing model is the interaction by exchange with the mean (IEM) model [26] where the composition of all particles in a cell are moved a small distance toward the mean composition using a characteristic mixing timescale. The IEM equation is shown in equation (7) where $\tilde{\phi}$ is the Favre mean-composition vector at the particle's location and τ_t is the turbulence time scale. The scalar mixing time scale τ_ϕ in equation (7) is often modelled as proportional to τ_t as in equation (8).

$$\frac{\partial \phi}{\partial t}=-\frac{\phi-\tilde{\phi}}{2\tau_\phi} \quad (7)$$

$$\tau_\phi\equiv\frac{\tau_t}{c_\phi} \quad (8)$$

The mixing models just described are either non-local [21-24] or over-local [25], thereby producing imperfect combustion modelling processes. Recently Wandel [27] has proposed a new mixing model which randomizes the particle interaction in a local manner. The proposed model is the Stochastic Particle Diffusion Length (SPDL) [27] model, which is based upon the practical localness of the random inter-particle distance [28].

Combustion processes are very complex especially because the chemical reactions between chemical species involved in burning fuel (gas, liquid or solid) are highly non-linear functions of temperature and species concentration. Significant errors are produced when a computational fluid dynamics (CFD) code only solves the Reynolds Averaged Navier-Stokes (RANS) model as written in equation (9), which involves averaging the source terms. All terms are averaged and the models work reasonably well when solving $\bar{\phi}$ for velocity but not for chemical source term.

$$\frac{\partial \bar{\rho}\bar{\phi}}{\partial t} + \frac{\partial \bar{\rho}\bar{\phi}\,\overline{u_i}}{\partial x_i} = \frac{\partial}{\partial x_i}\left[D\frac{\partial \bar{\phi}}{\partial x_i}\right] + \bar{W} + \bar{T} \tag{9}$$

These models are inaccurate and produce errors by ignoring the correlation between the source term variables, which are generally significant, compared to the corresponding averages. These errors were overcome by the CMC model using the conditional averages method. It takes the average of the variables for a specified value of the mixture fraction and effectively averages over a smaller region of space. Compare to the RANS method, CMC is more accurate at the cost of more complex simulations that require more computational time. The chemical source term $\langle W|Z\rangle$ can be calculated using equation (10) and the reaction rate is obtained by using equation (11). The Arrhenius equation [16] in equation (12) is used to determine the reaction rate constant. The chemical source term, reaction rate and reaction rate constant can be written as:

$$W_I = \frac{w_I}{\rho}\sum_j v_{Ij}\omega_j \tag{10}$$

$$\omega_j = \left[k_{fj}\prod_{I=1}^{n_s}[X_I]^{v_{Ij}^{'}} - k_{bj}\prod_{I=1}^{n_s}[X_I]^{v_{Ij}^{''}}\right] \tag{11}$$

$$k = AT^{\beta}exp\left(-\frac{E_a}{RT}\right) \tag{12}$$

with the rate of progress variable ω_j the net strength of reaction j in the forward direction and w_I is the mean molecular weight, ρ is density, E_a is the activation energy, A and β are Arrhenius constants, R is the ideal gas constant, and T is temperature. The CMC equation for two-phase homogenous systems is shown in equation (13):

$$\frac{\partial\langle Y|Z\rangle}{\partial t} = \langle N|Z\rangle\frac{\partial^2\langle Y|Z\rangle}{\partial Z^2} + \langle W|Z\rangle + \langle S|Z\rangle \tag{13}$$

In this paper we discretized the CMC equation using the Taylor series expansion method. FORTRAN code was used to run a simulation and produce results.

2.0 Taylor Expansion

Implicit finite difference relations have been derived by many mathematicians and physicists with various methods [29-41]. Most of them claim that all the implicit formulas can be derived from a Taylor series expansion. The Taylor series expansion is a good basis for studying numerical methods since it provides a means to predict a function's value at one point in terms of the function's value and derivatives at another point. In particular, the theorem states that any smooth function can be approximated as a polynomial [41]. There are many different types of numerical differentiation formulations, depending on the number of points, direction of the formula and the required derivative order [42]. The Taylor expansion is a useful method to discretize partial differential equations to minimize and accurately predict the value of the error term. The expansions for values for y_1 and y_{-1} which are to the right and to the left of y_0 respectively are shown below up to the 7th-order derivatives as equations (14) and (15).

$$y_1 = y_0 + (\Delta x_1)\frac{dy}{dx}\Big|_{x_0} + \frac{(\Delta x_1)^2}{2!}\frac{d^2y}{dx^2}\Big|_{x_0} + \frac{(\Delta x_1)^3}{3!}\frac{d^3y}{dx^3}\Big|_{x_0} + \frac{(\Delta x_1)^4}{4!}\frac{d^4y}{dx^4}\Big|_{x_0} + \frac{(\Delta x_1)^5}{5!}\frac{d^5y}{dx^5}\Big|_{x_0} + \frac{(\Delta x_1)^6}{6!}\frac{d^6y}{dx^6}\Big|_{x_0} + \frac{(\Delta x_1)^7}{7!}\frac{d^7y}{dx^7}\Big|_{x_0} \quad (14)$$

$$y_{-1} = y_0 + (\Delta x_{-1})\frac{dy}{dx}\Big|_{x_0} + \frac{(\Delta x_{-1})^2}{2!}\frac{d^2y}{dx^2}\Big|_{x_0} + \frac{(\Delta x_{-1})^3}{3!}\frac{d^3y}{dx^3}\Big|_{x_0} + \frac{(\Delta x_{-1})^4}{4!}\frac{d^4y}{dx^4}\Big|_{x_0} + \frac{(\Delta x_{-1})^5}{5!}\frac{d^5y}{dx^5}\Big|_{x_0} + \frac{(\Delta x_{-1})^6}{6!}\frac{d^6y}{dx^6}\Big|_{x_0} + \frac{(\Delta x_{-1})^7}{7!}\frac{d^7y}{dx^7}\Big|_{x_0} \quad (15)$$

We assume Δ_i is constant, then y_1 and y_0 become

$$y_1 = y_0 + (\Delta x)\frac{dy}{dx}\Big|_{x_0} + \frac{(\Delta x)^2}{2!}\frac{d^2y}{dx^2}\Big|_{x_0} + \frac{(\Delta x)^3}{3!}\frac{d^3y}{dx^3}\Big|_{x_0} + \frac{(\Delta x)^4}{4!}\frac{d^4y}{dx^4}\Big|_{x_0} + \frac{(\Delta x)^5}{5!}\frac{d^5y}{dx^5}\Big|_{x_0} + \frac{(\Delta x)^6}{6!}\frac{d^6y}{dx^6}\Big|_{x_0} + \frac{(\Delta x)^7}{7!}\frac{d^7y}{dx^7}\Big|_{x_0} \quad (16)$$

$$y_{-1} = y_0 + (\Delta x)\frac{dy}{dx}\Big|_{x_0} + \frac{(\Delta x)^2}{2!}\frac{d^2y}{dx^2}\Big|_{x_0} + \frac{(\Delta x)^3}{3!}\frac{d^3y}{dx^3}\Big|_{x_0} + \frac{(\Delta x)^4}{4!}\frac{d^4y}{dx^4}\Big|_{x_0} + \frac{(\Delta x)^5}{5!}\frac{d^5y}{dx^5}\Big|_{x_0} + \frac{(\Delta x)^6}{6!}\frac{d^6y}{dx^6}\Big|_{x_0} + \frac{(\Delta x)^7}{7!}\frac{d^7y}{dx^7}\Big|_{x_0} \quad (17)$$

Rearrangement of equations (16) and (17) becomes the first order derivatives at x_0 as below (equations (18), (19) and (20)). Equation (16) is used to obtain the forward difference method (which calculates $\frac{dy}{dx}\Big|_{x_0}$ based on forward movement from y_0 to y_1):

$$\frac{dy}{dx} = \frac{(y_1 - y_0)}{\Delta x} - \frac{(\Delta x)}{2!}\frac{d^2y}{dx^2} - \frac{(\Delta x)^2}{3!}\frac{d^3y}{dx^3} - \frac{(\Delta x)^3}{4!}\frac{d^4y}{dx^4} - \frac{(\Delta x)^4}{5!}\frac{d^5y}{dx^5} - \frac{(\Delta x)^5}{6!}\frac{d^6y}{dx^6} - \frac{(\Delta x)^6}{7!}\frac{d^7y}{dx^7} \quad (18)$$

Equation (17) is used to obtain the backward difference method, which calculates $\frac{dy}{dx}\Big|_{x_0}$ based on backward movement from y_0 to y_{-1}.

$$\frac{dy}{dx} = \frac{(y_0 - y_{-1})}{\Delta x} - \frac{(\Delta x)}{2!}\frac{d^2y}{dx^2} - \frac{(\Delta x)^2}{3!}\frac{d^3y}{dx^3} - \frac{(\Delta x)^3}{4!}\frac{d^4y}{dx^4} - \frac{(\Delta x)^4}{5!}\frac{d^5y}{dx^5} - \frac{(\Delta x)^5}{6!}\frac{d^6y}{dx^6} - \frac{(\Delta x)^6}{7!}\frac{d^7y}{dx^7} \quad (19)$$

By taking the difference between equations (16) and (17), the central difference method is derived which calculates $\frac{dy}{dx}\Big|_{x_0}$ based on the domain between y_1 and y_{-1}

$$\frac{dy}{dx} = \frac{(y_1 - y_{-1})}{2\Delta x} - \frac{(\Delta x)^2}{3!}\frac{d^3y}{dx^3} - \frac{(\Delta x)^4}{5!}\frac{d^5y}{dx^5} - \frac{(\Delta x)^6}{7!}\frac{d^7y}{dx^7} \quad (20)$$

with leading error term of $\frac{(\Delta x)^2}{6!}\frac{d^3y}{dx^3}$. This equation (20) is a second-order accurate method: $O(\Delta x^2)$. Since the differences actually evaluate the derivative at the midpoint of the finite difference, equation (20) estimates the derivative at x_0, while equation (18) and (19) estimate the derivative either side of x_0. Using central difference derivative, we can obtain $\frac{d^2y}{dx^2}$, as shown in equation (21). The first term in equation (21) can be re-arranged as equation (22) to show that this is simply a central difference of the first derivative.

$$\frac{d^2y}{dx^2} = \frac{(y_1 - 2y_0 + y_{-1})}{(\Delta x)^2} - \frac{2!(\Delta x)^2}{4!}\frac{d^4y}{dx^4} - \frac{2!(\Delta x)^4}{6!}\frac{d^6y}{dx^6} \quad (21)$$

$$= \frac{1}{\Delta x}\left(\frac{y_1 - y_0}{\Delta x} - \frac{y_1 - y_{-1}}{\Delta x}\right) \quad (22)$$

The leading error term is $\frac{(\Delta x)^2}{12!}\frac{d^4y}{dx^4}$, so this is a second-order accurate method: $O(\Delta x^2)$. The first-order derivative using the fourth-order Taylor expansion scheme is below, for the difference between y_2 and y_{-2}:

$$\frac{dy}{dx} = \frac{(y_{-2} - 8y_{-1} + 8y_1 - y_2)}{12\Delta x} + \frac{4(\Delta x)^4}{5!}\frac{d^5y}{dx^5} \tag{23}$$

with leading error term of $\frac{(\Delta x)^4}{30}\frac{d^5y}{dx^5}$. This is a fourth-order accurate method: O(Δx^4) and can be summarized as equation (24) to show that it is a weighted average of the "near" and "far" central differences:

$$\frac{dy}{dx} = \left(\frac{4}{3}\right)\frac{(y_1 - y_{-1})}{(2\Delta x)} - \left(\frac{1}{3}\right)\frac{(y_2 - y_{-2})}{(4\Delta x)} \tag{24}$$

The second-order derivative for the fourth-order Taylor expansion scheme is equation (25):

$$\frac{d^2y}{dx^2} = \frac{(-y_{-2} + 16_{y-1} - 30_{y_0} + 16_{y_1} - y_2)}{12(\Delta x)^2} + \frac{8(\Delta x^4)}{6!}\frac{d^6y}{dx^6} \tag{25}$$

with leading error term of $\frac{(\Delta x^4)}{90}\frac{d^6y}{dx^6}$, so this is a fourth-order accurate method: O(Δx^4). This can be summarized as equation (26):

$$\frac{d^2y}{dx^2} = \left(\frac{4}{3}\right)\frac{(y_{-1} - 2y_0 + y_1)}{(\Delta x)^2} - \left(\frac{1}{3}\right)\frac{(y_{-2} - 2y_0 + y_2)}{(2\Delta x)^2} \tag{26}$$

The third-order derivative for the second-order Taylor expansion scheme is shown as equation (27):

$$\frac{d^3y}{dx^3} = \frac{(-y_{-2} - 2y_{-1} - 2y_1 + y_2)}{2(\Delta x)^3} - \frac{(\Delta x)^2}{4}\frac{d^5y}{dx^5} \tag{27}$$

with leading error term of $\frac{(\Delta x)^2}{4}\frac{d^5y}{dx^5}$, so this is a second-order accurate method: O(Δx^2).

Assuming that there is a uniform spacing of Δx, using notation that $y^{(k)} = \frac{d^k y}{dx^k}$, for the Taylor series expansion, central difference derivatives can be summarized as equation (28):

$$y^{(k)} = C\sum_{i=\frac{(n-1)}{2}}^{\frac{(n-1)}{2}} z_i y_i + ET \tag{28}$$

Forward difference derivatives can be written as equation (29):

$$y^{(k)} = C\sum_{i=0}^{(n-1)} z_i y_i + ET \tag{29}$$

Backward difference derivatives can be written as equation (30):

$$y^{(k)} = C\sum_{i=-(n-1)}^{0} z_i y_i + ET \tag{30}$$

where *n* is the number of points (y_{-2},y_{-1},y_0,y_1,y_2 is equal to five points), *ET* is the leading error term and *z* is the coefficient of *y* for each point *i*.

3.0 Numerical Method

The finite difference schemes, as agreed by most of the scientific community, were first used by Euler (1707-1783) [43] to find an approximate solution of a differential equation. It was invented prior to boundary element methods (BEM), finite element methods (FEM), spectral methods, and discontinuous spectral element methods [44]. FDM is still relevant and remain competitive as a discretization method for use in many applications and can be used to solve problems with simple and complex geometry, such as fluid flows and gas reaction [45,46]. The Finite difference method (FDM) is a numerical method for approximating the solutions to partial differential equations by using finite difference equations to approximate derivatives based on the properties of Taylor expansions and on

the straightforward application of the definition of derivatives [47]. The objectives are to transform the calculus problem to algebra as from a continuous equation to a discrete equation. The discretization process is a mathematical process that divides the continuous physical domain into a discrete finite difference grid and then approximates each individual partial derivative in the partial differential equation. Using the Taylor expansion method, a partial differential equation was discretized in order to transform it to FORTRAN code. From the CMC equation, to study the discretization and code it in FORTRAN, simplifications of the CMC equation were used: the conditional chemical source term ($\langle W|Z\rangle$) and conditional generation due to droplet evaporation term ($\langle S|Z\rangle$) were not considered. So the homogeneous and passive CMC is equation (31):

$$\frac{\partial\langle Y|Z\rangle}{\partial t} = \langle N|Z\rangle \frac{\partial^2\langle Y|Z\rangle}{\partial z^2} \tag{31}$$

In this equation, the conditional mass fraction quantity $\langle Y|Z\rangle$ can be considered as "*Y* is a function of *Z*" (written as *y*(Z)) and conditional scalar dissipation $\langle N|Z\rangle$ is expressed as "*N* is a function of *Z*" (written as *N*(Z)). After summarizing all the assumption, the CMC equation becomes equation (32):

$$\frac{dy}{dt} = N\frac{d^2y}{dz^2} \tag{32}$$

The Taylor expansion equations (18) and (21) can be expressed in this nomenclature as:

$$\frac{dy}{\partial t} = \frac{y(z,t+\Delta t)-y(z,t)}{(\Delta t)} \tag{33}$$

$$\frac{d^2y}{\partial z^2} = \frac{y(z+\Delta z,t)-2y(z,t)+y(z-\Delta z,t)}{(\Delta z)^2} \tag{34}$$

The final form of the CMC equation after discretization is equation (35) for the explicit method and equation (36) for implicit.

$$YH(i,j+1) = \Big(\big(1-(2B)\big)*YH(i,j)\Big) + \big(B*YH(i+1,j)\big) + \big(B*YH(i-1,j)\big) \tag{35}$$

$$(1+2B)*YH(i,j+1) - \big(B*YH(i+1,j+1)\big) - \big(B*YH(i-1,j+1)\big) = YH(i,j) \tag{36}$$

where $B = N\frac{\Delta t}{(\Delta z)^2}$., *YH* is the array in the computer code for the variable mass fraction of fuel, *i* is the index for mixture fraction and *j* is the index of the time step. Equations (35) and (36) were coded in FORTRAN to simulate the CMC modelling.

4.0 Program Code using FORTRAN

FORTRAN is a high level of programming language developed by team of IBM programmers led by John Backus in 1954. The name of FORTRAN was derived from the words "Formula Translation", started from 1957 when the first FORTRAN compiler was used. It has evolved through FORTRAN II, FORTRAN 66, until now FORTRAN 2008. In this study, FORTRAN 90 was used to code the CMC equation that was discretised as in equation (35) and (36). Parts of the FORTRAN code for the explicit and implicit methods are listed in appendixes A and B.

There are three parameters as input to the code: *dt* is time step size, *dz* is step size in space and *TT* is total time for the simulation. These three parameters will determine the accuracy and total time taken to run the code. The more steps taken, the longer it will take the computer to complete the simulation. The result of the simulation must be check to ensure its convergence meets expectations. These parameter must also comply with the Courant-Friedrichs-Lewy (CFL) condition [48,49] to ensure the stability of the solution and that the result acceptably reaches a converged solution. The stability of the solution is very important because an unstable condition will create large errors in the

solution and wrong predictions of the result. The time-step must satisfy the condition shown in equation (37) otherwise the simulation will produce incorrect results.

$$CFL = N\frac{dt}{dz^2} \leq \frac{1}{2} \tag{37}$$

In order to achieve this condition, the time step must be small enough for the flow conditions.

5.0 Comparison of Explicit and Implicit Methods

Simulation is very important to reduce the experimental cost. After simulation results are obtained and ready to be validated, then experimentation can take place to confirm the findings and results from the simulation. In this study, the value for the conditional scalar dissipation N in equation (32) was assumed to be the constant 0.5. Results from the simulation are plotted in Figures 1, 2 and 3. The explicit method was found to converge faster than the implicit method, reaching the steady-state condition after 190 time steps whereas implicit required 247 time steps. ("Steady-state" is defined as no variation in 5 significant figures.) In addition, the time required for the computer to calculate one time step was significantly shorter for the explicit method. However, the explicit method reaches the steady-state too quickly due to greater errors in this method for the same time-step. The implicit method can have a bigger time-step (δt) for the same accuracy as the explicit method.

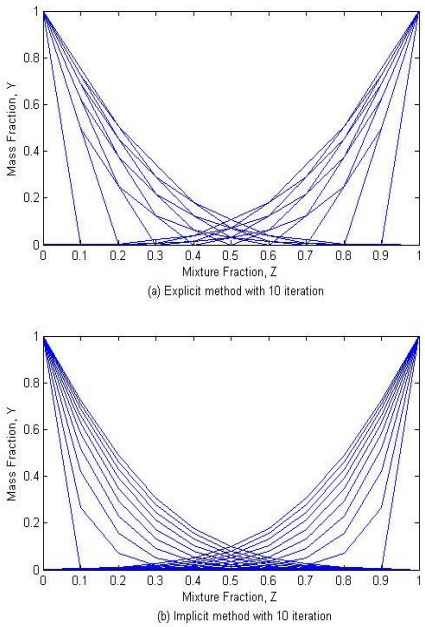

Figure 1: CMC mass fraction vs mixture fraction for first 10 iterations (a) explicit method and (b) implicit method

Note that the first lines (from Y = 1.0 to 0.0 over the gap of Z = 0.1) are the initial conditions that oxidizer (left lines) is zero everywhere except at Z = 0 and fuel is zero everywhere except at Z = 1. The cell size used here is Z = 0.1, which is why the line drops at both edges for both methods. The explicit and implicit methods start to show changes for both air and fuel mass fraction immediately (after the first time step: the second lines in Figure 1). For the explicit method at mixture fraction equal to 0.2, mass fraction is still 0.00 whereas for the implicit method, for mixture fraction equal to 0.2, mass fraction values are between 0.0 and 0.2. This variation is because differences between adjacent cells in the explicit method can only march one cell at a time, while the difference influences all cells immediately in the implicit method. Because of this, the explicit method influences the adjacent cell too much, which results in the method reaching steady-state quicker.

These mixing processes were repeated over many time steps. Figure 2(a) and 2(b) show the progress for 20 time steps. The mixing process will reach steady-state and equilibrium when both air and fuel completely mix with both reaching 0.5 at mixture fraction equal to 0.5 (Figure 3(a) and Figure 3(b)).

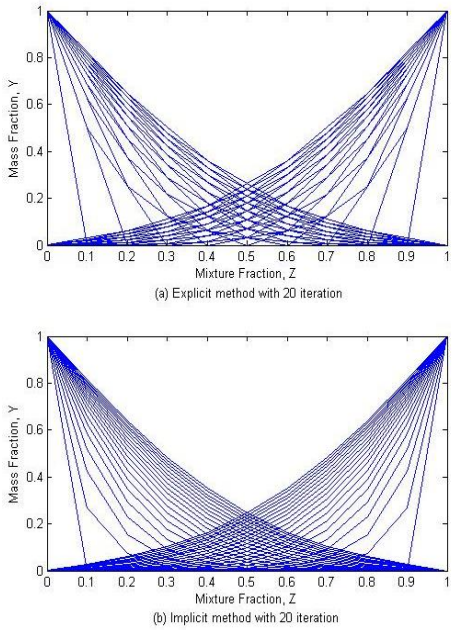

Figure 2: CMC mass fraction vs mixture fraction for first 20 iterations (a) explicit method and (b) implicit method

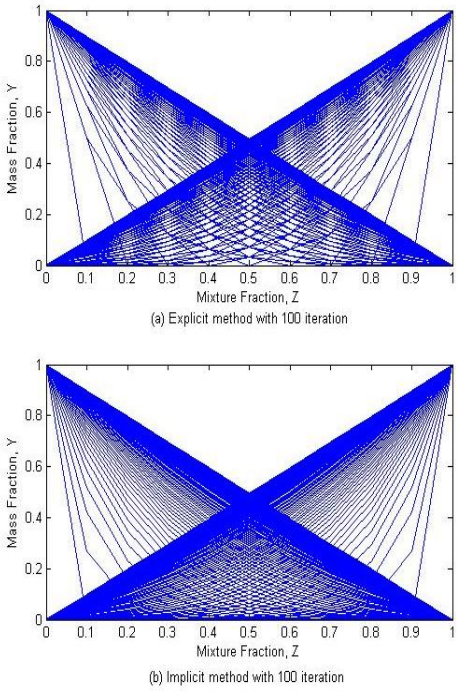

Figure 3: CMC mass fraction vs mixture fraction for first 100 iterations (a) explicit method and (b) implicit method

6.0 Conclusions

The Taylor expansion was utilized to discretize the partial differential equation for the CMC model. The modelling of CMC using explicit and implicit methods was successfully implemented using FORTRAN as the simulation software. From the results, we conclude that the implicit method is more accurate for the same time step, whereas it is much easier to write the FORTRAN code for the explicit method and the computational time to calculate is much shorter for the same time step. When preparing to conduct simulations, the researcher needs to balance the requirements of time step size with the necessary accuracy and time required for the simulations to be run.

Acknowledgments

The authors would like to thank University of Southern Queensland, Universiti Malaysia Pahang and Ministry of Higher Education, Malaysia for providing laboratory facilities and financial support.

References

[1] Pope, SB 1985, 'PDF Methods for Turbulent Reactive Flows', Progress in Energy and Combustion Science, **11**, pp. 119-192.
[2] Pope, SB 2000, *Turbulent Flows*, Cambridge University Press.
[3] Haworth, DC 2010, 'Progress in Probability Density Function Methods for Turbulent Reacting Flows', Progress in Energy and Combustion Science **36(2)**, pp. 168-259.
[4] Peters, N 1980 'Local Quenching Due to Flame Stretch and Non Premixed Turbulent Combustion', Western States Section of the Combustion Inst., spring meeting, Paper WSS-80-4.
[5] Peters, N 1983, 'Local Quenching Due to Flame Stretch and Non Premixed Turbulent Combustion' Combustion Science Technology **30**, pp. 1-17.
[6] Peters, N 1984, 'Laminar Diffusion Flamelet Models in Non-Premixed Turbulent Combustion," Progress in Energy and Combustion Science, **10**, pp. 319-339.
[7] Peters, N 2000, Turbulent Combustion, Cambridge University Press.
[8] Bilger, RW 1980,"in: P.A. Libby, F.A. Williams (Eds.), Turbulent Reacting Flows, Springer-Verlag, Berlin, pp. 65-113.
[9] Chen, H, Chen, S, Kraichnan, RH 1989, 'Probability Distribution of a Stochastically Advected Scalar Field', Physical Review Letters, **63**, pp. 2657-2660.
[10] Pope SB 1991, 'Mapping Closures for Turbulent Mixing and Reaction 'Theoretical and Computational Fluid Dynamics **2(5–6)**, pp.255-270.
[11] Klimenko, AY 1990, 'Multi-component Diffusion of Various Admixtures in Turbulent Flow', Fluid Dynamics, **25(3),** pp. 327-334.
[12] Bilger, RW 1993, 'Conditional Moment Closure for Turbulent Reacting Flow', Physics of Fluids, **5**, pp. 436-444.
[13] Klimenko, AY, Bilger, RW 1999, 'Conditional Moment Closure for Turbulent Combustion', Progress in Energy and Combustion Science, **25**, pp. 595-687.
[14] Klimenko, AY and Pope, SB 2003, 'The Modelling of Turbulent Reactive Flows based on Multiple Mapping Conditioning', Physical of Fluids, **15**, pp. 1907-1925.
[15] Klimenko, AY 2004,'Matching the Conditional Variance as a Criterion for Selecting Parameters in the Simplest Multiple Mapping Conditioning Models', Physical of Fluids, **16**, pp. 4754-4757.
[16] Wandel, AP, 2005 'Development of Multiple Mapping Conditioning (MMC) for Application to Turbulent Combustion', PhD Thesis, University of Queensland, Australia.
[17] Wandel, AP and Klimenko, AY, 2005, 'Multiple Mapping Conditioning Applied to Homogeneous Turbulent Combustion', Physics of Fluids, **17**, p. 128105.
[18] Vogiatzaki, K, Kronenburg, A, Navarro-Martinez, S and Jones WP 2011 'Stochastic Multiple Mapping Conditioning for a Piloted, Turbulent Jet Diffusion Flame', Proceedings of the Combustion Institute **33**, pp. 1523-1531.
[19] Ternat, F., Orellana, O. and Daripa, P 2011, 'Two Stable Methods with Numerical Experiments for Solving the Backward Heat Equation', Applied numerical Method, **61**, pp. 266-284.

[20] Clarke, J, Wandel, AP and Mastorakos, E 2010 'Analysis of Data to Develop Models for Spray Combustion', Southern Region Engineering Conference (SREC), 11-12 Nov, Australia, Paper ID SREC2010-F2-1.
[21] Curl, RL 1963, 'Dispersed Phase Mixing: I. Theory and Effects in Simple Reactors', AIChE Journal, **9**, pp. 175-181.
[22] Levenspiel, O and Spielman, LA 1965, 'A Monte Carlo Treatment for Reacting and Coalescing Dispersed Phase Systems. Chemical Engineering Science, **20**, pp. 247-254.
[23] Janicka, J, Kolbe, W, Kollmann, W 1979, 'Closure of the Transport Equation for the Probability Density Function of Turbulent Scalar Fields', Journal of Non-Equilibrium Thermodynamics, 4, pp. 47-66.
[24] Dopazo, C 1979, 'Relaxation of Initial Probability Density Functions in the Turbulent Convection of Scalar Fields', Physics of Fluids, **4(22)**, pp. 20-30.
[25] Subramaniam, S and Pope, SB 1998, 'A Mixing Model for Turbulent Reactive Flows based on Euclidean Minimum Spanning Trees', Combustion and Flame, **115**, pp.487-514.
[26] Villermaux,J., Devillon, JC.,1972 'Representation de la Coalescence et de la Redispersion des Domaines de Segregation dans un uide par un Modµele d'Interaction Phenomenologique, Second International Symposium on Chemical Reaction Engineering, pp. 1-13.
[27] Wandel, AP, 2011 A Stochastic Micromixing Model based on the Turbulent Diffusion Length Scale, 2011 Australian Combustion Symposium (ACS2011), Paper ID: ACS2011-20.
[28] Noor, MM, Yusaf, TF and Wandel AP, 2011 Study of Random Particle Interactions for Analysis of Diffusion Lengths in Turbulent Combustion Modelling, 2011 Australian Combustion Symposium (ACS2011), Paper ID: ACS2011-36.
[29] Collatz, L 1966, 'The Numerical Treatment of Differential Equations', Springer Verlag, Berlin.
[30] Krause, E. 1971 'Mehrstellen Verfahren zur Integration der Grenzschicht gleichungen', DLR Mitteilungen, **71(13)**, pp. 109-140.
[31] Adam, Y 1975 'A Hermitian Finite Difference Method for the Solution of Parabolic Equations' Computational Mathematics Application, **1**, pp. 393–406.
[32] Rubin, SG and Graves, RA 1975 'Viscous Flow Solutions with a Cubic Spline Approximation' Computational Fluid, **3**, pp. 1-36.
[33] Hirsh, RS 1975, 'Higher Order Accurate Difference Solutions of Fluid Mechanics Problems by a Compact Differencing Scheme' Journal Computational Physics, **19**, pp. 20-109.
[34] Ciment, M and Leventhal, SH 1975, 'Higher Order Compact Implicit Schemes for the Wave Equation' Mathematics Computational, **29**, pp. 885-944.
[35] Adam, Y 1977 'Highly Accurate Compact Implicit Methods and Boundary Conditions', Journal Computational Physics, **24**, pp. 10-22.
[36] Leventhal, SH 1980, 'The Operator Compact Implicit Method for Reservoir Simulation' Journal Society of Petroleum Engineering, **20**, pp. 120-128.
[37] Peyret, R 1978 'A Hermitian Finite Difference Method for the Solution of the Navier–Stokes Equations', Proceedings First of the Conference on Numerical Methods in Laminar and Turbulent Flows, Pentech Press, Plymouth (UK) pp. 43-54.
[38] Peyret, R and Taylor, TD 1982, 'Computational Methods for Fluid Flow', Springer Verlag, New York.
[39] Lele, SK 1992, 'Compact Finite Differences with Spectral-like Resolution', Journal Computational Physics, **103(1)**, pp. 16-42.
[40] Rubin, SG and Khosla, PK 1977 'Polynomial Interpolation Method for Viscous Flow Calculations' Journal Computational Physics, **24**, pp. 217-246.
[41] Chapra CS and Canale RP 2006, 'Numerical Methods for Engineers', McGraw Hill, 5th Edition, Singapore.
[42] Griffiths DV and Smith IM 2006, 'Numerical Methods for Engineers' Chapman and Hall/CRC, 2nd Edition, New York.

[43] D'Acunto, B, 2004, Computational Methods for PDE in Mechanics, World Scientific Publishing Co. Italy.
[44] Kopriva DA and Kolias JH 1996, A Conservative Staggered-grid Chebyshev Multidomain Method for Compressible Flows, Journal of Computational Physics, **125**, pp. 244-261.
[45] Yeung, RW and Ananthakrishnan, P 1992, Oscillation of a Floating Body in a Viscous Fluid, Journal of Engineering Mathematics, **26**, pp. 211-230.
[46] Yeung, RW and Ananthakrishnan, P 1997, Viscosity and Surface-Tension Effects on Wave Generation by a Translating Body, Journal of Engineering Mathematics, **32**, pp. 257-280.
[47] Hirsh, C 2007, Numerical Computation of Internal and External Flows: Fundamentals of Computational Fluid Dynamics, Butterworth-Heinemann, 1, Burlington, USA.
[48] Courant, R, Friedrichs, K and Lewy, H 1928, Über Die Partiellen Differenzengleichungen Der Mathematischen Physik, Mathematische Annalen, **100(1)**, pp 32-74.
[49] Courant, R, Friedrichs, K and Lewy, H 1967, On the Partial Difference Equations of Mathematical Physics, English Translation from German 1928 Paper. IBM Journal, pp. 215-234.

Nomenclature

W	Chemical Source Term
N	Scalar Dissipation Rate
Z	Mixture Fraction (a Conserved Scalar)
P	Favre Joint PDF of Composition
B	Constant (Function of *dt* and *dz*)
E_a	Activation Energy
A, β	Constants
K, j	Grid Point Involved in Space Difference
L, i	Grid Point Involved in Time Difference
S_k	Reaction Rate for Species k
J_{ik}	Molecular Diffusion Flux Vector
RANS	Reynolds Averaged Navier Stokes
CMC	Conditional Moment Closure
MMC	Multiple Mapping Conditioning
PDF	Probability Density Function
EMST	Euclidean Minimum Spanning Tree
MC	Modified Curl's model
YA	Air Mass Fraction
YH	Fuel Mass Fraction
u_i	Favre Mean Fluid Velocity Vector
ψ	Composition Space Vector
u_i''	Fluid Velocity Fluctuation Vector
$\dot{\omega}_i$	Net Formation Rate per Unit Volume
v	Kinematic Viscosity,
α	Thermal Diffusivity, m^2/s
ϕ	Particle Composition
k	Thermal Conductivity, W/mK
k_j	Arrhenius Reaction Rate Coefficient
ρ	Density or Mean Fluid Density, kg/m^3
R	Universal Gas Constant (8.31431 kJ $kmol^{-1}K^{-1}$)
$\overline{w}$	Molecular Weight of a Gas Mixture
$\langle \ \vert \ \rangle$	Conditional Average
$\langle Y\vert Z\rangle$	Mass Fraction of Fuel
$\langle N\vert Z\rangle$	Conditional Scalar Dissipation
$\langle W\vert Z\rangle$	Conditional Chemical Source Term
$\langle S\vert Z\rangle$	Conditional Generation Due to Droplet Evaporation

Appendix A: Part of FORTRAN code for explicit method

```
Comment: Initial Condition
YH(:,1)  = 0.0
YH(L,1) = 1.0
YA(1,:)  = 1.0
YA(L,1) = 0.0
Comment: Main Calculation
Do j=1, K
Do i=2, L
YH(i,j+1) =((1-(2*B))*YH(i,j)) + (B*YH(i+1,j)) + (B*YH(i-1,j))
YA(i,j+1) =((1-(2*B))*YA(i,j)) + (B*YA(i+1,j)) + (B*YA(i-1,j))
End Do
Comment: Boundary Condition
YH(1,j+1)=0
YH(L,j+1)=1
YA(1,j+1)=1
YA(L,j+1)=0
End Do
```

Appendix B: Part of FORTRAN code for implicit method

```
Comment: Initial Condition
A(1,1) = 0.0
A(2,1) = 1.0
A(1,2) = 0.0
Do i=2, dz-1
A(3,i-1) = - B
A(2,i ) = 1.0 + 2.0 * B
A(1,i+1) = - B
End Do
A(3,dz-1) = 0.0
A(2,dz) = 1.0
A(3,dz) = 0.0
Comment: Factor the matrix
Call MatrixC (dz,a,b,FF)
Do j = 2, dt
Call YH (z_min,z_max,t_min,t(j),B(1))
B(2:dz-1) = YH(2:dz-1,j-1)
Call YA (z_min,z_max,t_min,t(j),B(dz))
WW = 0
Call MatrixD (dz,a,b,WW)
YH(1:dz,j) = B(1:dz)
End Do
Comment: Subroutine for matrix
Do i = 1, n-1
If (P(2,i) .eq. 0.0 ) then
info = i
Write (*, '(P)') 'MatrixD - error'
Return
End If
P(3,i) = P(3,i) / P(2,i)
P(2,i+1) = P(2,i+1) - P(3,i) * a(1,i+1)
End Do
```