Towards a CFD Model for Turbulent, Reactive, Two-Phase Flow

Department of Chemical Engineering
University of Cambridge

L. R. McGlashan

Robinson College

Supervised by Dr. Markus Kraft

13th June, 2008

Preface

This work has been carried out between October 2007 and June 2008 at the Department of Chemical Engineering, University of Cambridge, UK.

The work contained in the dissertation, or any part thereof, has not been submitted for any other degree.

This CPGS dissertation contains approximately 9,200 words.

The author would like to acknowledge Dr. M. Kraft for his supervision of the work and J. Akroyd and A. Smith for their collaboration and assistance.

Summary

This dissertation aims to lay down the theoretical foundations for utilising Computational Fluid Dynamics (CFD) to model turbulent, reactive, two-phase flow.

The principles behind the CFD solvers are summarised. The discretisation of the Navier-Stokes equations using the Finite Volume Method (FVM). Methods for solving the linear system of equations that result from the FVM are then outlined.

Methods of modelling turbulence are outlined and compared, namely the k-ε model and Large Eddy Simulations (LES). For a simple column the continuous flowfield was simulated using both methods, which compare well.

A Probability Density Function (PDF) method for solving chemical reactions is implemented, with some success, in a commercial CFD package. There are issues regarding the coupling of the chemical source term closure to the CFD package that need to be resolved. The results obeyed the mass balance but do not model the reaction accurately enough. The influence of the computational mesh needs to be investigated and the possibility of having to decouple the source term calculations from the CFD needs to be considered, although this is undesirable.

Lastly, a survey of the current literature on two-phase flow modelling is presented. It is clear that there are still issues with the modelling of the hydrodynamics of multiphase flow, in particular the accurate simulation of flow regime transitions. New methods for solving the Particle Size Distribution (PSD) of bubbly flow are suggested. The use of DQMoM, a numerical method for solving the PDF transport equation, to model the PSD will be pursued in the future.

Contents

Report produced in LaTeX

1 Introduction

As computers become more powerful the applicability of Computational Fluid Dynamics (CFD) to the solution of complex fluid flows plays an increasingly important role. This research will focus on using CFD to predict the flow characteristics of turbulent, reactive, two-phase flows. These are of interest as many industrial reactors are multiphase. The interaction between the phases is complex and is still a subject for debate. Many of the closures for the interactions are based on empirical correlations and are specific to particular flow configurations.

There are many different CFD packages available, ranging from those used commercially (such as STAR-CD) to research codes used in academia. Although there are of course differences between the codes, most of them use the same underlying principles to solve the Navier-Stokes equations for fluid flow, stemming from the use of the Finite-Volume Method (FVM). In addition to utilising a commercial package, an open source library of C++ FVM codes named OpenFOAM will be used. For the purposes of future work this application will be useful as it is easier to implement new routines within it than it is with STAR-CD. Because of this, both OpenFOAM and STAR-CD have been used in parallel wherever possible.

A great deal of research has gone into the formulation of models that simulate gas-liquid and liquid-liquid flows. The modelling of the hydrodynamics of these flows is still an ongoing topic and there is no agreement on the exact form of the models; what forces should be included and how they are modelled. The use of DNS methods such as volume-of-fluid and front-tracking to model the interactions between individual bubbles, along with experiments, will help to provide better closure for the larger scale models.

This dissertation aims to first introduce the methods employed by CFD solvers. An understanding of how these work is crucial for ensuring that simulation results are realistic. The concept of the FVM will be introduced along with discretisation procedures such as central and upwind differencing that create a linear system of equations from the Navier-Stokes equations, and also the different linear algebra solvers used in a CFD context. The models employed to simulate turbulent flow are then outlined and compared. The choice of turbulence model is important, as they all have limited applicability for different types of flow. A Probability Density Function (PDF) method for simulating reactions is then employed and compared to experiments and previous numerical simulations. This employs the use of moment methods to solve for the chemistry. An overview of the current state of the literature on two-phase flow modelling is then provided, comparing Euler/Euler and Euler/Lagrange methods and introducing the use of population balances to solve for the particle size distribution.

2 Computational Fluid Dynamics (CFD)

2.1 Navier-Stokes Equations

The Navier-Stokes equations for fluid flow are derived by considering the conservation of mass and momentum over a finite volume of fluid.

The equations within this dissertation are presented in explicit form as opposed to the vector form, and use the summation convention (unless otherwise stated). The fluids are assumed to be incompressible and Newtonian throughout.

There are many good references where derivations of the Navier-Stokes equations may be found. Batchelor (1981) gives derivations in vector form and Kay and Nedderman (1985) derivations in cartesian form.

2.1.1 Continuity Equation

When conservation of mass over a finite volume of incompressible fluid is considered, the total mass entering the control volume must equal the total mass exiting the control volume (assuming there are no sources/sinks of mass). This leads to the continuity equation:

$$\frac{\partial u_i}{\partial x_i} = 0. \tag{2.1}$$

2.1.2 Momentum Equation

Applying Newton's second law (the rate of change of momentum on an element equals the total forces acting on that element) to a volume of fluid gives the momentum equation:

$$\rho\frac{\partial u_i}{\partial t} + \rho\frac{\partial u_j u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j \partial x_j} - \rho g_i. \tag{2.2}$$

The left hand side of the momentum equation comprises the rate of change of momentum of a fluid with time and due to convection of momentum by the fluid flow. The right hand side is made up of the pressure, viscous and gravitational forces respectively.

2.1.3 General Scalar Transport Equation

The momentum equation is a specific case of the more generic scalar transport equation given below:

$$\rho\frac{\partial \phi_i}{\partial t} + \rho\frac{\partial u_j \phi_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \Gamma\frac{\partial^2 \phi_i}{\partial x_j \partial x_j} + S_\phi. \tag{2.3}$$

This models the transportation of a property of the fluid (energy or species concentration for example). S_ϕ is a general term encompassing all sources/sinks of the property ϕ. For example in the momentum equation the gravitational force on the fluid is a source of momentum.

2.2 Discretisation of the Navier-Stokes Equations

There are a number of ways to solve the Navier-Stokes equations, including finite difference, finite element and spectral methods. Commercial CFD packages use finite volume methods (a form of finite differencing), which integrate and discretise equations in an intuitive and conservative form, resulting in a linear system of equations which are solved using iterative solution algorithms.

2.2.1 Discretisation Schemes

The moment transport equation used in the modelling of reactions in Section 4 will be used to illustrate how equations are discretised using the FVM:

$$\rho\langle u_i\rangle\frac{\partial\lambda}{\partial x_i}-\frac{\partial}{\partial x_i}\left[\Gamma_T\frac{\partial\lambda}{\partial x_i}\right]=\rho R_\lambda\ . \tag{2.4}$$

Equation (2.4) is formulated to fit the form of the STAR-CD general scalar transport equation.

The computational domain is split up into any number of individual cells (or volumes). The FVM works on the basis that the amount of a property that is convected and diffused into a control volume equals the amount that is convected and diffused out of a control volume, plus/minus any sources/sinks of mass within a control volume (e.g. reaction). This may be represented as follows (Versteeg and Malalasekera, 2007):

Rate of change of a property over time	=	Net rate of increase of a property due to convection into a C.V.	+	Net rate of increase of a property due to diffusion into a C.V.	+	Net rate of creation of a property within a C.V.

Using the concept of the FVM, Equation (2.4) may be written as

$$\int_V\rho\langle u_i\rangle\frac{\partial\lambda}{\partial x_i}dV-\int_V\frac{\partial}{\partial x_i}\left[\Gamma_T\frac{\partial\lambda}{\partial x_i}\right]dV=\int_V\rho R_\lambda dV\ . \tag{2.5}$$

FVM hinges around using the divergence theorem to convert volume integrals to surface integrals, which allows fluxes to be evaluated across surfaces:

$$\int_V\frac{\partial\phi}{\partial x_i}dV=\int_A n_i\phi dA\ . \tag{2.6}$$

Using the divergence theorem on Equation (2.5) gives

$$\int_A n_i\left(\rho\langle u_i\rangle\lambda\right)dA-\int_A n_i\left(\Gamma_T\frac{\partial\lambda}{\partial x_i}\right)dA=\int_V\rho R_\lambda dV\ . \tag{2.7}$$

Equation (2.7) is solved later in a one dimensional test case, so to illustrate how equations are discretised the one dimensional form of Equation (2.7) will be used.

Figure 1 represents an arbitrary node on a uniform computational grid. Equation (2.7) may be written as

$$\left[(\rho\langle u\rangle A\lambda)_{F_{n+1}} - (\rho\langle u\rangle A\lambda)_{F_{n-1}}\right] - \left[\left(\Gamma_T A\frac{d\lambda}{dx}\right)_{F_{n+1}} - \left(\Gamma_T A\frac{d\lambda}{dx}\right)_{F_{n-1}}\right] = \rho\overline{R}_\lambda V \ , \quad (2.8)$$

where the surface integrals are approximated as the sum of the fluxes into and out of the cell. The volume integral of the source term is written as the mean value of the source term in that cell multiplied by the cell volume.

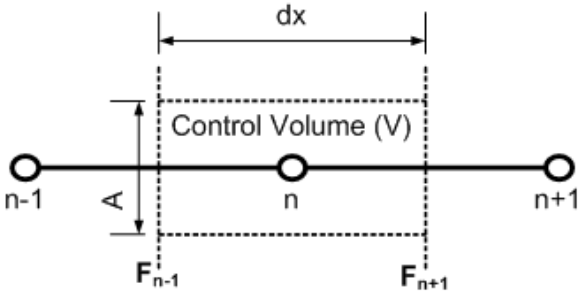

Figure 1: *Control volume.*

The values of ρ, $\langle u\rangle$ and Γ_T are known at the nodes $n-1$, n and $n+1$ (cell centres), but their values at the cell faces F_{n-1} and F_{n+1} are needed in order to solve Equation (2.8). They are approximated simply be linearly interpolating the values between the nodes. For example Γ_T would be:

$$(\Gamma_T)_{F_{n+1}} = \frac{(\Gamma_T)_{n+1} + (\Gamma_T)_n}{2} \ . \quad (2.9)$$

In the diffusion terms, the gradient of λ is usually discretised using central differencing. This is simply the difference between the forward and backward Taylor expansions of the variable, truncated after the first order terms and giving

$$\left(\frac{d\lambda}{dx}\right)_{F_{n+1}} = \frac{\lambda_{n+1} - \lambda_n}{dx} \ . \quad (2.10)$$

The convection terms in this case may be discretised using the upwind scheme as it converges much faster than the central differencing scheme. It takes account of the direction of the flow. Assuming the direction of the flow in Figure 1 is from left to right, the upwind scheme states

$$(\lambda)_{F_n} = (\lambda)_{n-1} \ . \quad (2.11)$$

For reasons that will be described later, the source term $\overline{R}_\lambda$ may be given in linear form

$$\overline{R}_\lambda = R_\alpha + R_\beta\lambda_n \ , \quad (2.12)$$

where R_α and R_β are constants. Substituting the above relations into Equation (2.8) gives

$$
\begin{aligned}
& \left[\rho\frac{\langle u\rangle_n+\langle u\rangle_{n+1}}{2}\lambda_n-\rho\frac{\langle u\rangle_{n-1}+\langle u\rangle_n}{2}\lambda_{n-1}\right] \\
- & \left[\frac{(\Gamma_T)_{n+1}+(\Gamma_T)_n}{2}\frac{\lambda_{n+1}-\lambda_n}{dx}-\frac{(\Gamma_T)_n+(\Gamma_T)_{n-1}}{2}\frac{\lambda_n-\lambda_{n-1}}{dx}\right] \\
= & \left[\rho\ dxR_\alpha+\rho\ dxR_\beta\lambda_n\right]
\end{aligned}
$$

Grouping the terms gives

$$
\begin{aligned}
& \left(\rho\frac{\langle u\rangle_n+\langle u\rangle_{n+1}}{2}+\frac{(\Gamma_T)_{n+1}+(\Gamma_T)_n}{2dx}+\frac{(\Gamma_T)_n+(\Gamma_T)_{n-1}}{2dx}-\rho\ dxR_\beta\right)\lambda_n \\
= & \left(\rho\frac{\langle u\rangle_{n-1}+\langle u\rangle_n}{2}+\frac{(\Gamma_T)_n+(\Gamma_T)_{n-1}}{2dx}\right)\lambda_{n-1} \\
+ & \left(\frac{(\Gamma_T)_{n+1}+(\Gamma_T)_n}{2dx}\right)\lambda_{n+1} \\
+ & \rho\ dxR_\alpha\ .
\end{aligned}
$$

Note that this equation has the form

$$a_n\lambda_n=a_{n-1}\lambda_{n-1}+a_{n+1}\lambda_{n+1}+S\ , \tag{2.13}$$

so for n nodes there will be n equations to be solved. The equations can be written in matrix form

$$\mathbf{A}.\mathbf{x}=\mathbf{b}\ , \tag{2.14}$$

where $\mathbf{A}$ is a matrix of coefficients of the property λ, $\mathbf{x}$ is a vector of the values of the property at the nodal points (cell centres) and $\mathbf{b}$ is a vector of the sources S.

This method is easily extended to 2D/3D Cartesian grids; for example for 2D grids there will simply be 2 more terms in each equation arising from the 2 extra faces. Care must be taken when using the upwind discretisation scheme however, as it does not give good results if the flow is not aligned with the grid.

Unstructured grids are what most CFD solvers use in practice. Unstructured refers to the fact that the cell volumes can have any shape and are not aligned with a coordinate system. One important point that needs to be mentioned is grid orthogonality. A grid is orthogonal if the vector normal to the cell face is parallel to the line connecting the adjacent cell centres. Most unstructured grids are non-orthogonal, which makes the maths complex. More details may be found in Versteeg and Malalasekera (2007).

2.2.2 Discretisation and Solution of Navier-Stokes Equations

In solving Equation (2.4) the flow field is assumed to be known already. In reality flow properties such as velocity, pressure and diffusivity need to be calculated before solving additional transport equations. Equations of fluid flow are often nonlinear, meaning that the matrix of coefficients, $\mathbf{A}$, will be dependent on the flow variable in the vector $\mathbf{x}$. This problem is solved by iterating the solution; initial values for the flow variable are used to generate a solution which is then used as the new set of initial values. This process is repeated until convergence.

The problem with solving the momentum equation (Equation (2.2)) is that in order to solve the velocity field, the pressure field needs to be either known or solved for. For compressible flow an equation of state may be used, however for incompressible flow this is obviously not applicable. For three dimensional flows the most common procedures (these are the ones used by STAR-CD) for solving the pressure and velocity fields are the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) (Patankar, 1980) and PISO (Pressure Implicit with Splitting of Operators) (Issa, 1985) algorithms and their variants.

If the momentum equation were to be discretised as outlined previously, then the pressure term could be linearly interpolated (see Figure 1)

$$\frac{\partial p}{\partial x} = \frac{p_{n+1} - p_{n-1}}{2dx} \ . \tag{2.15}$$

The problem with this arrangement is that if there was a strongly varying pressure field so that $p_{n-1} = 1$, $p_n = 2$ and $p_{n+1} = 1$, then the above interpolation would yield no pressure gradient. One solution to this is to use what is called a staggered grid. This is where the velocities are not calculated at the cell centre (n), but at the cell face (F_{n+1}). Now the pressure term in the momentum equations would be

$$\frac{\partial p}{\partial x} = \frac{p_{n+1} - p_n}{dx} \ , \tag{2.16}$$

and the pressure gradient would have a finite value.

However, for complex geometries with non-cartesian meshes, staggering the grid becomes very complex and inefficient, as the velocities have to be aligned normal to the cell faces in order to calculate fluxes. CFD solvers use what are called co-located grids, which is where all the variables are stored at the cell centres as in the previous examples. To get around the problem of calculating a sensible pressure gradient, better interpolation schemes are used. For example OpenFOAM uses one similar to the Rhie-Chow scheme (Karrholm, 2006).

It is useful to observe how SIMPLE is implemented within OpenFOAM. Details may be found in Jasak (1996). The steps are described below:

1. Formulate the momentum equation without the pressure term in. It is then discretised and solved equal to the pressure term, which is explicitly calculated using the old pressure field. This is known as the predictor step, as a prediction of velocity is found using the old pressure field.

2. Update boundary conditions for the pressure field.

3. Calculate fluxes across the cell faces.

4. Define and solve the pressure correction equation to obtain a new pressure field.

5. Correct the velocity.

6. Iterate until convergence.

2.3 Solution of Linear Systems of Equations

There are many ways to solve a linear system of equations in the form

$$\mathbf{A}.\mathbf{x} = \mathbf{b} \ . \tag{2.17}$$

In CFD applications the matrix $\mathbf{A}$ is usually sparse and diagonally dominant, so in most cases it is a waste of computing resources to invert the matrix directly using elimination and decomposition methods. The methods used in commercial CFD applications are variants of iterative techniques, whose principles are described in the following sections.

2.3.1 Iterative Methods

All iterative methods start by taking an initial starting value for the vector $\mathbf{x}$ and solving the system to get a new value for each element in $\mathbf{x}$.

The simplest iterative methods for solving Equation (2.17) are the Jacobi and Gauss-Seidel methods.

Take the equation for the cell n:

$$\sum_{i=1}^{N} a_{n,i}x_i = b_n \ . \tag{2.18}$$

For cell n this can be rearranged in terms of all the other cell values:

$$x_n = \frac{1}{a_{n,n}} \left(b_n - \sum_{i \neq n} a_{n,i}x_i \right) \ . \tag{2.19}$$

Gauss-Seidel differs from Jacobi simply in that after the equation for a cell has been solved, the old cell value is then substituted for the new cell value in $\mathbf{x}$, as opposed to storing the new cell values in a separate vector. This saves memory and also converges to a solution faster.

In reality these are not used directly but are used in combination with other techniques to 'smooth' a solution. This is because after an iteration of the Jacobi/Gauss-Seidel algorithm, the residual error always decreases for every element, provided that the matrix is diagonally dominant. These methods are also used as the solution algorithms in multigrid methods.

CFD solvers always precondition the matrix of coefficients before solving the system. A simple example is to multiply $\mathbf{A}$ by a diagonal matrix with the same diagonal entries to increase the value of the diagonal of $\mathbf{A}$. This makes the matrix more diagonally dominant and quicker to solve.

2.3.2 Conjugate Gradient and Multigrid Methods

STAR-CD and OpenFOAM use Conjugate Gradient or Multigrid methods for solving systems of equations. These will not be mentioned in detail, but a quick overview of where to find information on the methods will be given.

A good introduction to Conjugate Gradient (CG) Methods is given in Shewchuk (1994). It is based on the method of steepest descent.

The main problem with using iterative solvers is that the finer the computational grid, the longer it takes to reach a converged solution. Algebraic Multigrid (AMG) Methods

work by taking the computational grid and coarsening it several times, then solving the system of equations on each grid using an iterative solver such as Gauss-Seidel. The system of equations is solved on the finest grid and the residuals mapped onto the coarser grids. The iterative solver is then used to work out the error and this error is then interpolated onto the finer grid and added to the initial solution. This process is repeated and a converged solution achieved much faster. There is an online resource for multigrid methods at Douglas (2003). Algebraic refers to the fact that the actual discretisation of the equations is not repeated or changed during the coarsening process, instead the matrix $\mathbf{A}$ simply has its elements combined.

3 Turbulence Modelling

At a high enough Reynolds number, fluid flow becomes turbulent. This means that measuring a property of the flow at the same point in space will give different values at different times, implying randomness.

3.1 Turbulent Length Scales

When averaging the Navier-Stokes equations, they are averaged at time and length scales much larger than the scales that molecules interact at, but at a scale smaller than the spatial/temporal variations of the variable concerned. This is the continuum hypothesis. It allows variables to be defined as constant at a point in space and time.

Turbulent kinetic energy, k, is the energy that the largest eddies (usually of the same order of magnitude as the largest length scale of the flow) extract from the mean flow. This energy is passed down to ever smaller eddies until (at the Kolmogorov length scale) the energy is dissipated, via viscous forces, as heat.

The large scales of turbulence influence the mean flow the most, and Reynolds stress closures usually model the kinetic energy of the turbulence and the dissipation of this energy from the largest eddies.

3.2 Reynolds Averaged Navier-Stokes (RANS) Equations

One way of accounting for the randomness introduced by turbulence is to consider a property of the flow as being composed of a mean value and a fluctuating component. This is known as Reynolds decomposition and is given as:

$$\phi = \langle\phi\rangle + \phi' \tag{3.1}$$
$$u = \langle u\rangle + u' \tag{3.2}$$
$$p = \langle p\rangle + p' \tag{3.3}$$

By inserting these into the Navier-Stokes equations and averaging them, a set of equations in terms of the mean flow properties is formed. The mean of the fluctuations, by definition, is equal to zero ($\overline{\phi'} = 0$). The Reynolds averaged continuity, momentum and scalar transport equations are:

$$\frac{\partial\langle u_i\rangle}{\partial x_i} = 0 \tag{3.4}$$

$$\rho\frac{\partial\langle u_i\rangle}{\partial t} + \rho\frac{\partial\langle u_j\rangle\langle u_i\rangle}{\partial x_j} = -\frac{\partial\langle p\rangle}{\partial x_i} + \mu\frac{\partial^2\langle u_i\rangle}{\partial x_j\partial x_j} - \rho\frac{\partial\langle u_i'u_j'\rangle}{\partial x_j} - \rho g_i \tag{3.5}$$

$$\frac{\partial\langle\phi_i\rangle}{\partial t} + \frac{\partial\langle u_j\rangle\langle\phi_i\rangle}{\partial x_j} = -\frac{\partial\langle p\rangle}{\partial x_i} + \Gamma\frac{\partial^2\langle\phi_i\rangle}{\partial x_j\partial x_j} - \rho\frac{\partial\langle u_i'\phi_j'\rangle}{\partial x_j} + \langle S_\phi\rangle\ . \tag{3.6}$$

The third terms on the right-hand side of the momentum equation and the scalar transport equation are respectively known as the Reynolds stresses and scalar fluxes. They physically represent the effect of the turbulent fluctuations on the mean flowfield.

3.2.1 k-ε Turbulence Model

The k-ε model is a two equation model and is based on Boussinesq's isotropic eddy viscosity assumption. Isotropy refers to the assumption that μ_T is assumed to be uniform in all directions. The Reynolds stresses in a Newtonian fluid are related to the velocity gradients by an eddy (or turbulent) viscosity. So in a Newtonian fluid the Reynolds stresses are proportional to the mean rates of strain:

$$-\rho\langle u_i' u_j' \rangle = \mu_T \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) - \frac{2}{3}\rho k \delta_{ij} \ , \tag{3.7}$$

where the turbulent viscosity (μ_T) is given by

$$\mu_T = \rho C_\mu \frac{k^2}{\varepsilon} \ , \tag{3.8}$$

and the turbulent kinetic energy (k) is defined as

$$k = \frac{1}{2}\langle u_i' u_i' \rangle \ . \tag{3.9}$$

This is related to the turbulent diffusivity (Γ_T) by

$$\Gamma_T = \frac{\mu_T}{\sigma_T} \ , \tag{3.10}$$

where σ_T is a turbulent Prandtl/Schmidt number, usually assumed to be unity due to the Reynolds analogy (turbulent transport of heat/mass/scalar occurs via same process as momentum).

The last term in Equation (3.7) is the isotropic component of the Reynolds stresses, which is subtracted from the trace of the Reynolds stresses because only the anisotropic stresses transport momentum.

The problem has now been reformulated so that the turbulent viscosity is now the unknown in the RANS equations. The k-ε model assumes it is related to two quantities, the turbulent kinetic energy (k) and the dissipation rate (ε), by Equation (3.8). Transport equations for these two quantities may be formulated and solved for. The k equation may be derived by mathematical manipulation of the Navier-Stokes equations, but the ε equation is really an empirical one. The constants have values that have been fitted to large amounts of experimental data.

Although the k-ε model is the most widely used, it does have its limitations. The main problem is that the eddy viscosity assumption is often a poor one, especially when flows are rotational or buoyant, for example. In these cases the velocity gradients are not the sole cause of turbulence.

3.2.2 Other RANS Turbulence Models

There are many variations on the standard k-ε model, but one separate class are the Reynolds Stress Models (RSM). These solve transport equations for the Reynolds stresses directly. However the coupling between these equations and the momentum equation is quite strong and so solving them can take much longer than the k-ε models. On the plus side they do away with the Boussinesq approximation used in the k-ε model, which falls down easily in situations where Reynolds stresses may occur in the absence of velocity gradients (bubbly flow being an example, where the turbulence created by the dispersed

phase moving through the continuous phase contributes to the Reynolds stresses). RSM also provides a higher order closure than the k-ε models.

3.3 Large Eddy Simulations (LES)

Large eddy simulations initially found use in meteorology. They involve calculating the large scale eddies in a turbulent flow, as these are likely to have the largest influence on the flowfield. Two steps are involved in formulating the Navier-Stokes equations for LES:

1. A filter must be applied to the flow variables to split the flowfield into a resolved part and a subgrid part:

$$\phi = \tilde{\phi} + \phi' \tag{3.11}$$

2. A model for the turbulence at length scales smaller than the grid size must be used so that its influence on the larger eddies may be included.

The type of filter used is usually a box filter, which effectively splits the flowfield into a resolved part and subgrid part based on the size of the computational grid cells (Δ is the filter width, so this will correspond to the cell size), in a similar way to the Reynolds decomposition in the RANS method:

$$G(x) = \begin{cases} 1/\Delta & \text{if } |x| \text{ is} \leq \Delta/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

However, an important difference between RANS and LES is that $\tilde{\phi}$ is a random field and is unsteady, whereas $\langle\phi\rangle$ is not. Also, the filtered subgrid part of the field in LES does not (in general) equal zero, whereas the average of the fluctuating part of the field in RANS does equal zero.

The most widely used closure for the smaller turbulence scales is the Smagorinsky model (Smagorinsky, 1963). It uses the Boussinesq isotropic eddy viscosity assumption to close the subgrid stress terms, in a similar fashion to the k-ε models described earlier.

4 Reaction Modelling

4.1 Modelling Reaction Source Terms

The scalar transport equation (Equation (2.3)) may be used for the transport of concentrations or mole fractions of chemical species. If the species react with each other then the creation and destruction of the species is modelled through extra terms in the source term S_ϕ. Averaging these transport equations will introduce additional unclosed terms due to the reaction, as they are coupled and usually nonlinear. In other words, the mean reaction rate does not equal the reaction rate evaluated using the mean values of the species concentrations/mole fractions ($\overline{S(\phi)} \neq S(\overline{\phi})$).

There are a number of methods that may be used to close these source terms and these are widely documented in Pope (2000), Fox (2003) and Cant and Mastorakos (2008).

Here an implementation of the PDF (Probability Density Function) approach will be used. This is a statistical approach whereby the concentration of each species in each cell has a certain probability of having specific values.

Start with the joint composition PDF transport equation [1], the derivation of which may be found in Pope (2000) and Fox (2003):

$$\frac{\partial f_\phi}{\partial t} + \langle U_i \rangle \frac{\partial f_\phi}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\langle u_i |\ \psi \rangle f_\phi\right] = -\frac{\partial}{\partial \psi_\alpha}\left[\left(\langle \Gamma_\alpha \nabla^2 \phi_\alpha |\ \psi \rangle + S_\alpha(\psi)\right) f_\phi\right] \ . \tag{4.1}$$

This equation describes the transport of F_ϕ due to convection by the mean velocity and convection by the velocity fluctuations, and transport in composition space due to molecular mixing and due to reaction.

The scalar conditioned velocity term is closed using the gradient-diffusion model, which assumes that the flux of the PDF is aligned with its gradients:

$$\langle u_i |\ \psi \rangle = -\frac{\Gamma_T}{f_\phi} \frac{\partial f_\phi}{\partial x_i} \ . \tag{4.2}$$

The molecular mixing term is closed using the IEM (Interaction by Exchange with the Mean) mixing model, which states that a variable will relax to its mean value:

$$\langle \Gamma_\alpha \nabla^2 \phi_\alpha |\ \psi \rangle = -\frac{C_{\phi\alpha}}{2\tau_{\phi\alpha}} \left(\psi_\alpha - \langle \phi_\alpha \rangle\right) \ , \tag{4.3}$$

where $C_{\phi\alpha}$ / $\tau_{\phi\alpha}$ is a 'scalar mixing rate' and is approximated as in Tsai and Fox (1994):

$$\frac{C_{\phi\alpha}}{\tau_{\phi\alpha}} \approx \frac{C_\phi}{\tau_\phi} = C_\phi \frac{\sqrt{k}}{d} \ . \tag{4.4}$$

C_ϕ is a fitting parameter. In Tsai and Fox (1994) it was found that $C_\phi = 1.65$ gave good agreement to the experimental data of Li and Toor (1986).

The Direct Quadrature Method of Moments (DQMoM) is a numerical method for solving the PDF transport equation. A derivation of this method is given in Fox (2003). It involves approximating the PDF by weighted delta functions. For this reaction scheme in this dissertation a simpler method based on the principles of DQMoM will be used.

[1] The nomenclature used here may differ from that used in other sections of this dissertation. This is to ensure consistency with the literature on this particular topic.

4.2 Jet Reactor Reaction Scheme

The reaction scheme within the jet reactor is as follows:

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{R} \\ \mathrm{B} + \mathrm{R} &\xrightarrow{k_2} \mathrm{S} \end{aligned}$$

where $k_1 = 5 \times 10^6$ and $k_2 = 1.8 \times 10^3$ m^3 kmol^{-1} s^{-1} (Li and Toor, 1986).

4.3 Transported Equations

The first and second moments of a species allow the mean and variance of the reactants and products to be calculated within the reactor, which are useful engineering quantities. To extract these from the PDF transport equation, moments are taken of the entire equation to give eight equations;

$$\frac{\partial \lambda_{m_A,m_B,m_R,m_S}}{\partial t} + \langle U_i \rangle \frac{\partial \lambda_{m_A,m_B,m_R,m_S}}{\partial x_i} - \frac{\partial}{\partial x_i}\left[\Gamma_T \frac{\partial \lambda_{m_A,m_B,m_R,m_S}}{\partial x_i}\right] = R_{\lambda_{m_A,m_B,m_R,m_S}} \ , \quad (4.5)$$

where m_A is the order of the moment of A etc. So $\lambda_{1,0,0,0}$ is the first moment of A and $\lambda_{2,0,0,0}$ is the second moment of A. Transport equations for the first and second moments of A, B, R and S will be solved, which adds up to eight transport equations in total. The source terms for each moment transport equation are

$$\begin{aligned}
R_{\lambda_{1,0,0,0}} &= -k_1 \frac{C_T}{M_B}\left[\lambda_{1,1,0,0}\right] && (4.6)\\
R_{\lambda_{0,1,0,0}} &= -k_1 \frac{C_T}{M_A}\left[\lambda_{1,1,0,0}\right] - k_2 \frac{C_T}{M_R}\left[\lambda_{0,1,1,0}\right] && (4.7)\\
R_{\lambda_{0,0,1,0}} &= k_1 \frac{C_T M_R}{M_A M_B}\left[\lambda_{1,1,0,0}\right] - k_2 \frac{C_T}{M_B}\left[\lambda_{0,1,1,0}\right] && (4.8)\\
R_{\lambda_{0,0,0,1}} &= k_2 \frac{C_T M_S}{M_B M_R}\left[\lambda_{0,1,1,0}\right] && (4.9)\\
R_{\lambda_{2,0,0,0}} &= \frac{C_\phi}{\tau_\phi}\left[\lambda^2_{1,0,0,0} - \lambda_{2,0,0,0}\right] - 2k_1 \frac{C_T}{M_B}\left[\lambda_{2,1,0,0}\right] && (4.10)\\
R_{\lambda_{0,2,0,0}} &= \frac{C_\phi}{\tau_\phi}\left[\lambda^2_{0,1,0,0} - \lambda_{0,2,0,0}\right] - 2k_1 \frac{C_T}{M_A}\left[\lambda_{1,2,0,0}\right] - 2k_2 \frac{C_T}{M_R}\left[\lambda_{0,2,1,0}\right] && (4.11)\\
R_{\lambda_{0,0,2,0}} &= \frac{C_\phi}{\tau_\phi}\left[\lambda^2_{0,0,1,0} - \lambda_{0,0,2,0}\right] + 2k_1 \frac{C_T M_R}{M_A M_B}\left[\lambda_{1,1,1,0}\right] - 2k_2 \frac{C_T}{M_B}\left[\lambda_{0,1,2,0}\right] && (4.12)\\
R_{\lambda_{0,0,0,2}} &= \frac{C_\phi}{\tau_\phi}\left[\lambda^2_{0,0,0,1} - \lambda_{0,0,0,2}\right] + 2k_2 \frac{C_T M_S}{M_B M_R}\left[\lambda_{0,1,1,1}\right] \ . && (4.13)
\end{aligned}$$

The flowfield has already been solved for, so the mean velocities and turbulent diffusivities are known. To close the source terms the mixed moments need to be calculated. These are found using MoMEC.

4.4 MoMEC Algorithm

Only a brief overview of the MoMEC algorithm will be given here as the details are given in another dissertation [1]. The focus here will be on its implementation within a CFD solver for this reaction set.

When the reaction source terms are averaged there are terms that need to be closed. This will be done by approximating the PDF of each scalar by two particles, which will give the first and second moments for each scalar. These moments will be transported directly. The PDF is approximated by weighted delta functions:

$$f_\phi \approx \sum_{n=1}^{2} w^{(n)} \left[\prod_{\alpha=1}^{4} \delta \left[\varphi_\alpha - \varphi_\alpha^{(n)} \right] \right] . \tag{4.14}$$

MoMEC differs from DQMoM in that the weights are fixed, so in this case $w^{(1)} = w^{(2)} = 0.5$ as the sum of the weights must equal unity.

Taking moments of the above approximation gives the following expression for the mixed moments:

$$\lambda_{m_A,m_B,m_R,m_S} = \sum_{n=1}^{2} w^{(n)} \prod_{\alpha=1}^{4} \left[\varphi_\alpha^{(n)} \right]^{m_\alpha} . \tag{4.15}$$

From this the first and second moments for each species arise. As the first and second moments are being transported, but the positions are required to calculate the mixed moments, the positions must be calculated from the first and second moments and their distribution amongst the two particles reconstructed. The positions are given by:

$$\psi_\alpha^{(n)} = \langle \lambda_\alpha^1 \rangle \pm \sqrt{\langle \lambda_\alpha^2 \rangle - \langle \lambda_\alpha^1 \rangle} . \tag{4.16}$$

Now the distribution of the positions amongst the two particles is reconstructed so that the mixed moments may be found. The mixed moments in the reaction source terms can now be found using Equation (4.15), which enables the reaction source terms to be calculated.

4.5 Coupling MoMEC into a CFD Solver

4.5.1 User-Defined Subroutines in STAR-CD

User-defined subroutines are used to specify the source terms for the eight moment transport equations. It is worth noting how STAR-CD calls the source term subroutine. The transport equations are solved sequentially after the pressure/velocity corrections are performed within the SIMPLE/PISO algorithms. The MoMEC algorithm only needs to be called once per iteration, but the subroutine is called by STAR-CD once for each transport equation, per iteration, per computational cell. Therefore MoMEC is only run when the first transport equation is called for each cell and the values are stored for use in the other transport equations.

[1] J. Akroyd (2008)

4.5.2 Linearisation of Source Terms

Linearisation of the source terms is required to ensure numerical stability and to improve convergence. Patankar (1980) gives an account of source term linearisation. If the source term is a function of the dependent variable the dependency can be extracted through the linearisation

$$\bar{R} = S_C + S_P \phi_n \ , \tag{4.17}$$

where S_C and S_P may also depend on ϕ. A linear relationship is used as the system of equations will be solved using linear solvers.

If the source term is just a constant (S_C, which may depend on ϕ), then it is added to the vector $\mathbf{b}$ in the linear system of equations $\mathbf{A}.\mathbf{x} = \mathbf{b}$. By giving the source term a linear dependence on ϕ, S_P will be subtracted from the diagonal elements of the matrix $\mathbf{A}$. This will worsen the conditioning of the matrix which needs to be diagonally dominant for iterative solvers to work well. As a result a negative value for S_P is preferred and can often be desirable as it will improve the diagonal dominance of the matrix $\mathbf{A}$.

Another important point to note is that ideally all the coefficients in the linear system of equations should have the same sign. This ensures that the solution does not become unrealistic. Obviously, if there are source terms arising from reactions, this rule may not be obeyed.

For the reaction within the jet reactor, consider the source term from equation (4.11). It will be supplied to the CFD solver in the following form:

$$R_{\lambda_{2,0,0,0}} = S_C + S_P \lambda_{2,0,0,0} \ . \tag{4.18}$$

The mixing part of the source term fits naturally into this form:

$$\begin{aligned} R_{\lambda_{2,0,0,0}} &= \frac{C_\phi}{\tau_\phi}\lambda^2_{1,0,0,0} - \frac{C_\phi}{\tau_\phi}\lambda_{2,0,0,0} \\ S_C &= \frac{C_\phi}{\tau_\phi}\lambda^2_{1,0,0,0} \\ S_P &= -\frac{C_\phi}{\tau_\phi} \ . \end{aligned}$$

The conditions of S_P being negative and S_C being positive are satisfied. The problem is where to put the reaction part of the source term. It has a dependency on $\lambda_{2,0,0,0}$ but not explicitly. One option is to place it within S_C:

$$\begin{aligned} S_C &= \frac{C_\phi}{\tau_\phi}\lambda^2_{1,0,0,0} - 2k_1\frac{C_T}{M_B}\lambda_{2,1,0,0} \\ S_P &= -\frac{C_\phi}{\tau_\phi} \ . \end{aligned}$$

Note, however, that a potential problem is that S_C could become negative. This linear form will also not model the actual nonlinear behaviour of the source term very accurately. This problem expresses itself in the source terms for the other moments.

Another option is to take a first order Taylor expansion of (for this source term) $\lambda_{2,1,0,0}$. In general, for a function of a variable x, a linear approximation of it around a point a is

$$f(x) = [f(a) - af'(a)] + x\,[f'(a)] \ . \tag{4.19}$$

Taking the source term of the second moment of A as an example:

$$\begin{aligned} f(\lambda_{2,0,0,0}) &= \lambda_{2,1,0,0} \\ &= \left[\lambda_{2,1,0,0} - \lambda_{2,0,0,0}\frac{d\ f(\lambda_{2,0,0,0})}{d\lambda_{2,0,0,0}}\right] + \lambda_{2,0,0,0}\left[\frac{d\ f(\lambda_{2,0,0,0})}{d\lambda_{2,0,0,0}}\right] \ . \end{aligned}$$

Now the derivative has to be found. Central differencing was initially used:

$$\frac{d\ f(\lambda_{2,0,0,0})}{d\lambda_{2,0,0,0}} = \frac{1}{2\Delta_p}\left[f(\lambda^+_{2,0,0,0}) - f(\lambda^-_{2,0,0,0})\right] \ . \tag{4.20}$$

New values for $\lambda_{2,1,0,0}$ need to be found when $\lambda_{2,0,0,0}$ is changed slightly, i.e. perturbed, by an amount Δ_p. These values are found by running MoMEC again but with a small perturbation in the values of the moments.

However, this caused problems. It was realised that if a perturbation is applied to one moment, then the mass balance is no longer obeyed. So the derivative must also account for changes to the other moments [1]. Take for example how a mixed moment depends on the first moment of A:

$$\begin{aligned} \frac{d\ f(\lambda_{m_A,m_B,m_R,m_S})}{d\lambda_{1,0,0,0}} &= \frac{\partial f(\lambda_{m_A,m_B,m_R,m_S})}{\partial\lambda_{1,0,0,0}}\frac{d\ \lambda_{1,0,0,0}}{d\lambda_{1,0,0,0}} + \frac{\partial f(\lambda_{m_A,m_B,m_R,m_S})}{\partial\lambda_{0,1,0,0}}\frac{d\ \lambda_{0,1,0,0}}{d\lambda_{1,0,0,0}} \\ &\quad + \frac{\partial f(\lambda_{m_A,m_B,m_R,m_S})}{\partial\lambda_{0,0,1,0}}\frac{d\ \lambda_{0,0,1,0}}{d\lambda_{1,0,0,0}} + \frac{\partial f(\lambda_{m_A,m_B,m_R,m_S})}{\partial\lambda_{0,0,0,1}}\frac{d\ \lambda_{0,0,0,1}}{d\lambda_{1,0,0,0}} \ . \end{aligned}$$

As a result of this change the mass balance was obeyed. Preliminary results of the test cases and the jet reactor are given below.

[1] This was suggested by A. Smith and implemented and studied by J. Akroyd.

5 Two Phase Flow

There are three length scales at which two phase flow may be modelled. At the smallest level, where it is sought to obtain information on the deformation of the interface between two phases, DNS methods can be used. At an intermediate level, Euler/Lagrangian methods give information on the motion of individual bubbles by tracking them within a Lagrangian coordinate system. Finally, at the largest length scale, Euler/Euler (or multi-fluid) methods treat both the continuous and dispersed phases as interpenetrating continua, solving transport equations for both phases within an Eulerian coordinate system.

An overview of the Euler/Euler and Euler/Lagrange formulations are given in both Sokolichin and Eigenberger (1997) and Jakobsen *et al.* (1997). The two approaches will be compared here with regards to their applicability to modelling the characteristics of bubbly flow.

5.1 Euler/Euler Model

The Navier-Stokes equations may be applied to two-phase flows if the dispersed phase does not significantly affect the overall density of the flow (i.e. low density of the dispersed phase and/or low volume fraction). The corresponding equations are of the form:

$$\frac{\partial \left(\alpha^k u_i^k\right)}{\partial x_i} = 0 \ , \tag{5.1}$$

$$\frac{\partial \left(\rho^k \alpha^k u_i^k\right)}{\partial t} + \frac{\partial \left(\rho^k \alpha^k u_j^k u_i^k\right)}{\partial x_j} = -\alpha^k \frac{\partial p}{\partial x_i} + \alpha^k \mu \frac{\partial^2 u_i}{\partial x_j^2} + \mathbf{M}^k \ . \tag{5.2}$$

Here $\mathbf{M}$ accounts for the transfer of momentum between the two phases ($k = 1, 2$). This may include the drag, virtual mass, lift, rotation and strain forces acting on the dispersed and continuous phases. Note that $\mathbf{M}$ will have different signs for each phase. There are many different correlations that exist for the coefficients within the expressions for these forces, and one of the criticisms that may be made of multi-fluid models in general is that these coefficients are 'over-fitted' so that the model gives better results. The models also do not in general cope well when small perturbations to the flow are introduced (Monahan and Fox, 2007).

For turbulent bubbly flow, the turbulence generated by the bubbles needs to be taken into account. At high Reynolds numbers the k-epsilon model used in single phase flow needs to be altered to account for the turbulence introduced by the presence of the bubbles. Lain *et al.* (2002) added a source term into the transport equation for k in the standard k-ε model to account for this turbulence. This is done as the k-ε model is built on the eddy viscosity model which states that the Reynolds stresses are proportional to the velocity gradients in the flow, and bubble induced turbulence is not included in this hypothesis.

The main reason for using a multi-fluid model is also its biggest drawback. Because the dispersed phase is not tracked explicitly, the computational requirements are lower as only a couple of extra transport equations are required. However this means that details of particle/particle interactions and particle internal coordinates (sizes, physical properties) are not available (Crowe *et al.*, 1998).

To get information on the particle sizes, a multi-fluid model may be used in conjunction with a population balance model. This may be solved inside, for example, the SIMPLE algorithm after the velocity/pressure correction stages.

There are many different ways of solving population balance equations. Mass Flow Algorithm (MFA, Eibeck and Wagner (2001)), Single Particle Method (SPM, Vikhansky and Kraft (2005)), fixed-pivot technique (Kumar and Ramkrishna, 1996), Multiple Size Group (MUSIG) or moment methods (QMoM and DQMoM).

MUSIG assumes that all bubbles, whatever their size, travel with the same velocity. Bhole *et al.* (2008) have used the fixed-pivot technique to solve a population balance equation for the particle number density in an Eulerian coordinate system. They removed the aforementioned assumption of the MUSIG model by ignoring the inertial and stress terms in the gas phase momentum balance, making the computational requirements smaller.

Deen *et al.* (2000b) have performed 2D and 3D transient Euler-Euler (two-fluid) simulations of bubbly flow using the k-ε turbulence model that incorporates turbulence generated by the flow of bubbles. These simulations were compared to experimental observations from Deen *et al.* (2000a). They found that the 2D simulations corresponded well to the experiments but the 3D simulations did not.

The same authors then performed LES simulations and found that they were in better agreement with experiments than the RANS simulations (Deen *et al.*, 2001).

Fan *et al.* (2004) has applied DQMoM to solve the population balance equations coupled to a multi-fluid model. It is an improvement on previous moment methods as it also, like the fixed pivot method, allows different sized bubbles to have different velocities. This is due to the internal coordinates (such as particle size) having different weights (such as velocities) in the quadrature approximation. However, problems may arise if the distribution had a non-uniform shape. It should also be noted here that MoMEC would not be able to assign different velocities to different particle sizes as it uses fixed weights.

5.2 Euler/Lagrange Model

In the Euler-Lagrange model the standard Navier-Stokes equations are used for the continuous phase, but a Lagrangian coordinate system is used to track the dispersed phase. An equation for must be solved for each bubble:

$$\frac{d(m_b \mathbf{u}_d)}{dt} = \sum_i \mathbf{F}_i \ . \tag{5.3}$$

Lagrangian methods allow the break-up and coalescence of bubbles to modelled. These may be modelled using a population balance equation containing terms for coalescence along the lines of the Smoluchowski coagulation equation:

$$\begin{aligned} \frac{\partial n(t,x)}{\partial t} &= \frac{1}{2}\int_0^x K(x-x',x')n(t,x-x')n(t,x')dx' \\ &\quad - \int_0^\infty K(x,x')n(t,x)n(t,x')dx' \ . \end{aligned}$$

Where $n(t,x)$ is the number density of particles that have mass x at time t.

In words the above equation states that the number density of particles can increase by coagulation of two particles smaller than x and decrease by coagulation of a particle of size x with any particle.

This may be solved using DQMoM as done in Fan *et al.* (2004) or with SPM (Vikhansky and Kraft, 2005). The rates of coagulation and breakage are given by the kernels, K and the determination of these kernels is important in ensuring a physical solution.

Darmana *et al.* (2007) has used an Euler-Lagrange model to account for mass transfer and reaction in an isothermal bubble column, but without modelling any coalescence or breakup. The mass transfer closures need improving however, as the transfer rates are overpredicted.

In Deen *et al.* (2004) the front-tracking method was used to obtain information on the behaviour of individual bubbles. This information was used to then close terms in an Euler-Lagrange formulation. Although this approach is very useful, the front-tracking method would ideally be performed again for different flow configurations.

5.3 Modelling Turbulent, Two-Phase Reactions

Much work has been done on modelling the hydrodynamics of bubbly flow, but not so much on coupling mass transfer to these models, mainly due to the computational requirements imposed by this (Darmana *et al.*, 2007). The results of Deen *et al.* (2001) and Darmana *et al.* (2007) also show that in order to accurately model reactive bubbly flow, full 3D simulations are required. It is also suggested that LES resolves the transient hydrodynamics more accurately than RANS methods (Deen *et al.*, 2001), as the large scale turbulence structures vary with time. Simonnet *et al.* (2007) used STAR-CD to investigate different formulations of the drag force and their prediction of regime transitions, which is important for a general CFD model to achieve and yet has not been widely considered. DNS methods such as front-tracking (Tryggvason, 2001) have proven capable of providing closures for the force coefficients used in the Euler-Lagrange model (Deen *et al.*, 2004).

In light of these observations, it is clear that the hydrodynamics of two-phase flows is still an open problem, especially when modelling flow transitions. This must be kept in mind whilst coupling population balances to the Euler/Euler and Euler/Lagrange models.

6 Results

6.1 Turbulence Modelling - Jet Reactor

6.1.1 Geometry

The geometry of the tubular reactor is given in Figure 2, as described in Li and Toor (1986). The inlet conditions were assumed to be fully developed pipe annular flows. The inlet flow for the outer pipe was taken to be laminar ($Re < 2100$ based on d_h and U), with the turbulent intensity set to $I = 4\%$ (Tsai and Fox, 1994). The inner pipe was assumed to have a fully developed turbulent profile. This was generated by running a CFD simulation for a pipe of the same size. The solution was compared to an analytical one (Figure 3). Flowfields for $Re = 3530$ and $Re = 7552$ were obtained.

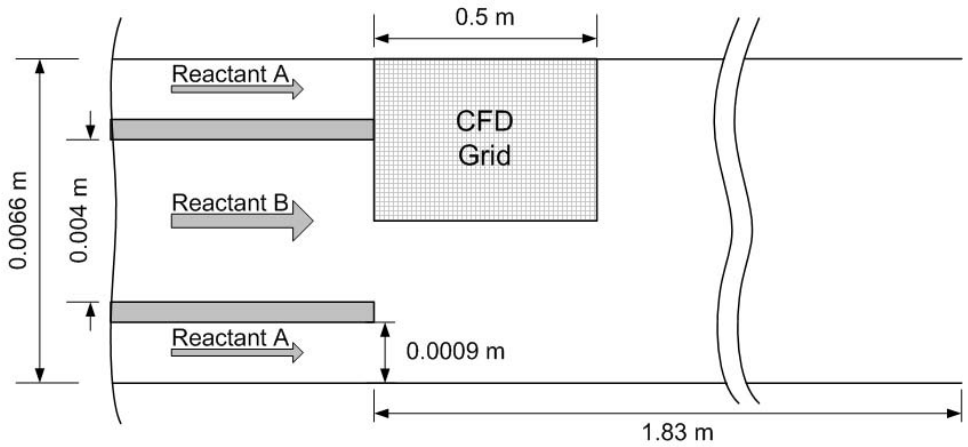

Figure 2: *Geometry for the single jet tubular flow reactor.*

Tsai and Fox (1994) contains figures of the variations in k and ε through the jet reactor. These compare very well with the profiles obtained from the simulations in this dissertation.

The mesh used was pseudo-2D. A 5 degree wedge was used with symmetrical boundaries on the faces in the radial direction. The boundary conditions are shown in Figure 4. Also, only the first 0.5 m of the reactor were simulated, as by this point the reactions had completed and the radial concentrations of all the species were uniform. A convergence study was performed to ensure that the solution was grid independent. The final mesh had around 10,000 cells. It should be noted at this point that although the flowfield is independent of grid size at this cell density, there is no guarantee that this grid is dense enough for the solution of the moment transport equations.

A monitoring cell was also set up near the inlet of the reactor, where the values of k and ε were highest, to check that their values had levelled off during the course of the simulations.

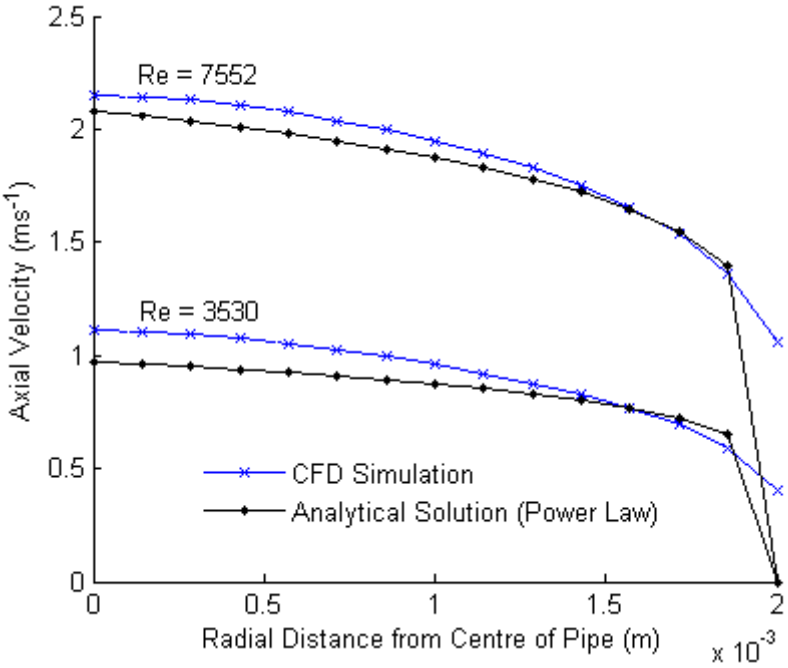

Figure 3: *Comparison of the turbulent inlet velocity profiles used in simulations for the jet reactor.*

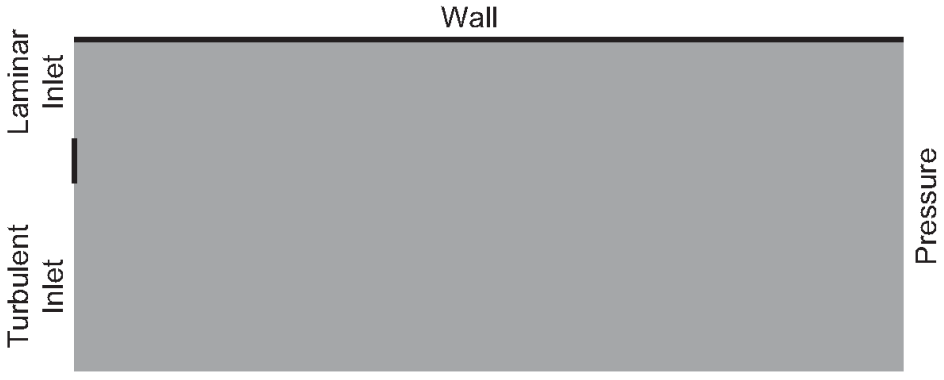

Figure 4: *Boundary conditions used on the mesh.*

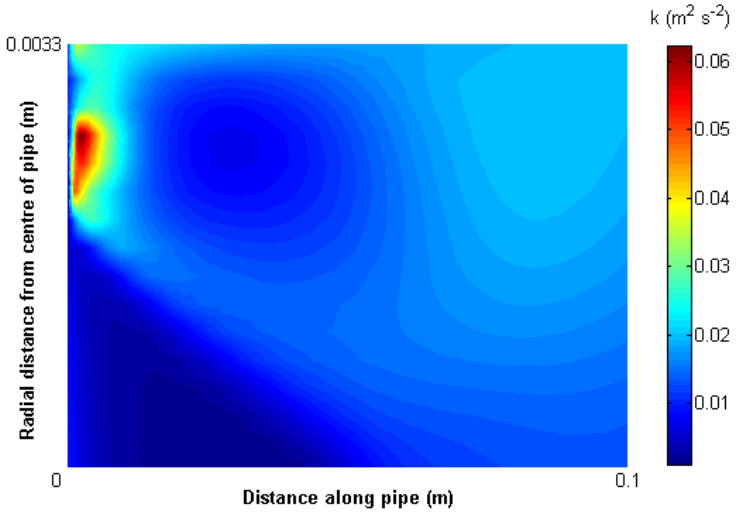

Figure 5: *Variation of k through the first 0.1 m of the jet reactor ($Re = 7552$).*

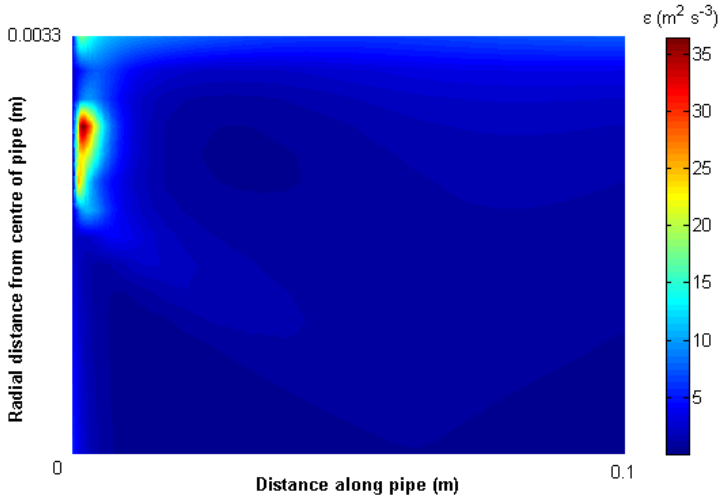

Figure 6: *Variation of ε through the first 0.1 m of the jet reactor ($Re = 7552$).*

6.2 Turbulence Modelling - Bubble Column

The flowfield of a bubble column was solved for using both RANS and LES models. The bubble column geometry is shown in Figure 7 and the fully developed radial velocity profile for both simulations is given in Figure 8. The RANS simulations took minutes to solve for, whereas LES took around a day using STAR-CD. The Reynolds number of the flow used was 5000. The velocity profiles are very similar. A useful step would be to compare these simulations with the experimentally determined velocity profiles from NMR techniques.

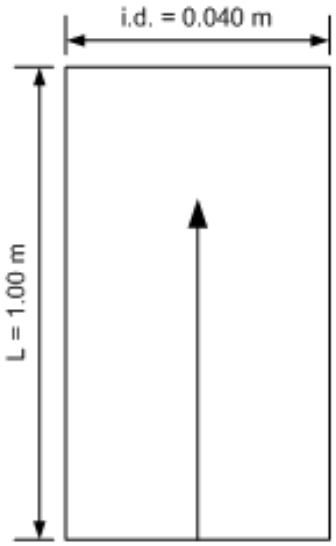

Figure 7: *Geometry of the bubble column.*

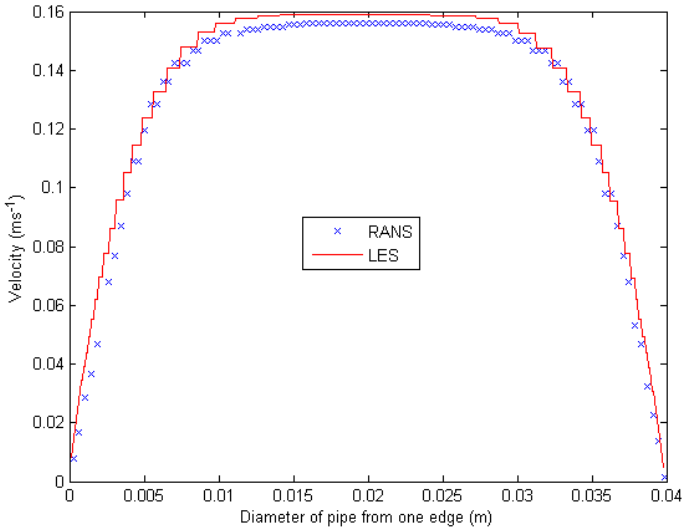

Figure 8: *Comparison of turbulent velocity profiles using RANS and LES.*

6.3 Reaction Modelling - MoMEC in 1D Test Cases

As an initial check to test MoMEC when coupled into a CFD solver, two simple cases were run in STAR-CD and compared to stand-alone solvers [2]. A mixing-only case was also run within OpenFOAM, and the results identical to those of STAR-CD. However this was not pursued further as OpenFOAM is written in C++, so the full Fortran77 MoMEC code would need to be rewritten in order to compare the two. This will only be considered once MoMEC works within STAR-CD.

The initial conditions for both cases were a mean mass fraction of 0.5 for the reactants A and B and a variance of 0.25 (or second moment of 0.5) for both reactants A and B.

Case 1) A single cell with no convection or diffusion. The reaction is allowed to run for one second. The CFD package effectively solves the following set of equations:

$$\frac{\partial \lambda_j}{\partial t} = R_{\lambda_j} \tag{6.1}$$

Case 2) A chain of cells one metre long, with a fluid flowing at one metre per second, at steady state with no diffusion. The CFD package solves the following set of equations:

$$\langle U_i \rangle \frac{\partial \lambda_j}{\partial x_i} = R_{\lambda_j} \tag{6.2}$$

It was found that the results for both cases 1 and 2 were very sensitive to the temporal/spatial step sizes. Figure 9 shows that a ten-fold increase in the number of time steps causes the solution to match better to the converged simulation results from a monte-carlo solver, but still not that well.

Figures 10 and 11 show a converged solution for case 2 that matches the monte-carlo results. This was achieved by having a variable spatial step size. It was recognized that smaller step sizes were required in the first 0.2 m of the reactor than the last 0.6 m due to the stronger gradients of the scalars in this region. For this case there were 50000 cells in the first 0.2 m, 20000 cells from 0.2 m to 0.4 m and 100 cells from 0.4 m to 1 m. Figure 11 shows that there were no negative variances for this case, which can be a problem when larger step sizes are used and the variance is very small.

This highlights a difficulty with implementing MoMEC within a 2D/3D environment. A suitable grid density will be important in order to arrive at an accurate solution, but this may come at a large computational cost.

6.4 Reaction Modelling - MoMEC in Jet Reactor

The solution of the mixing and reaction of species A, B, R and S was computed separately from the flowfield (see Section 6.1).

The results for the jet reactor shown here are the results obtained before the new linearisation routine was implemented (Equation (4.21)). As a result mass is not conserved.

6.4.1 Initial Conditions

The initial conditions at each of the inlets is given in Table 1 and the results are plotted in Figure 13.

[2] The stand-alone solvers were altered and maintained by J. Akroyd. They used high order integrators such as RADAU5 or Monte-Carlo solvers.

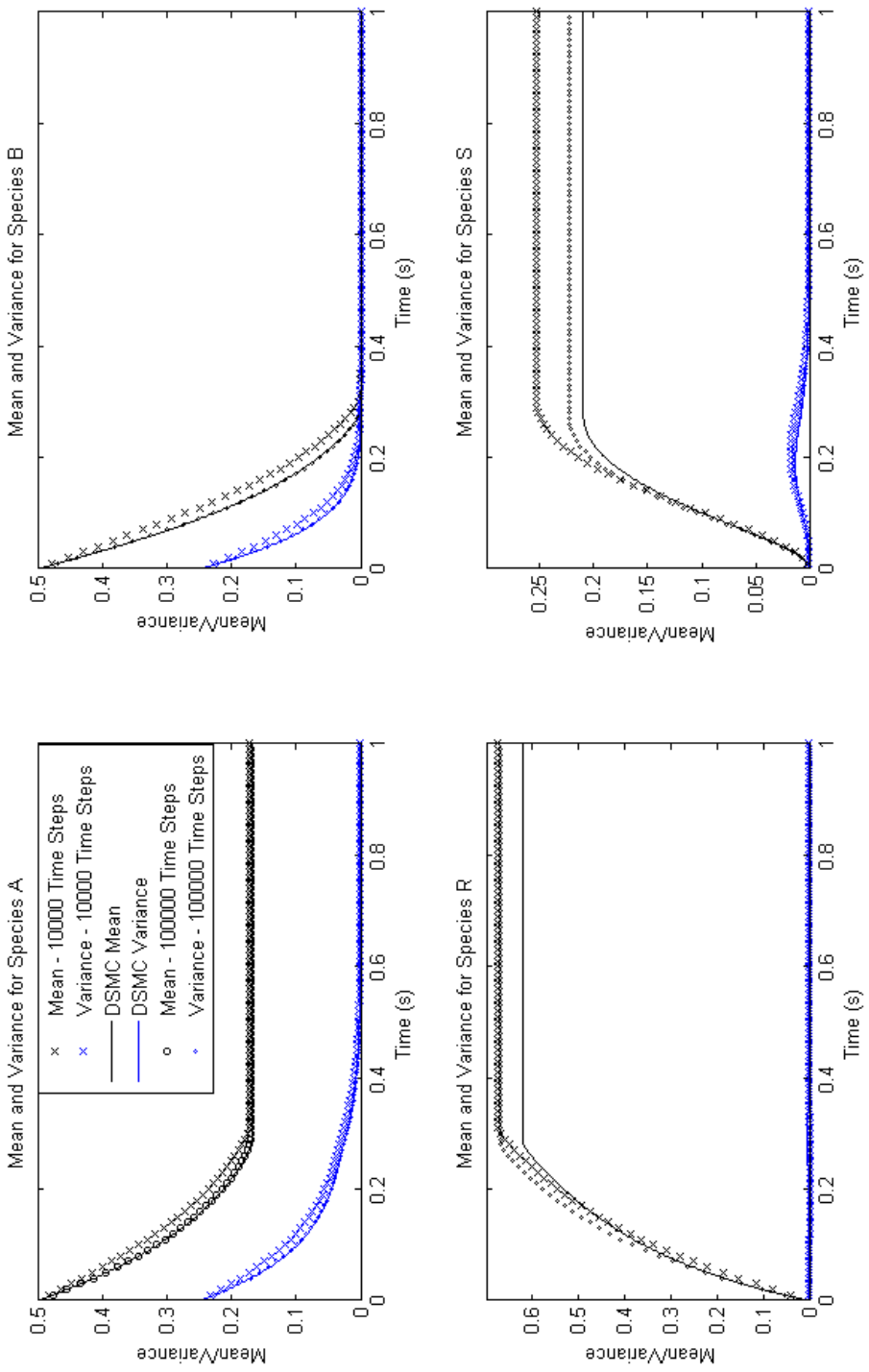

Figure 9: *Evolution of mean and variance with time (Case 1).*

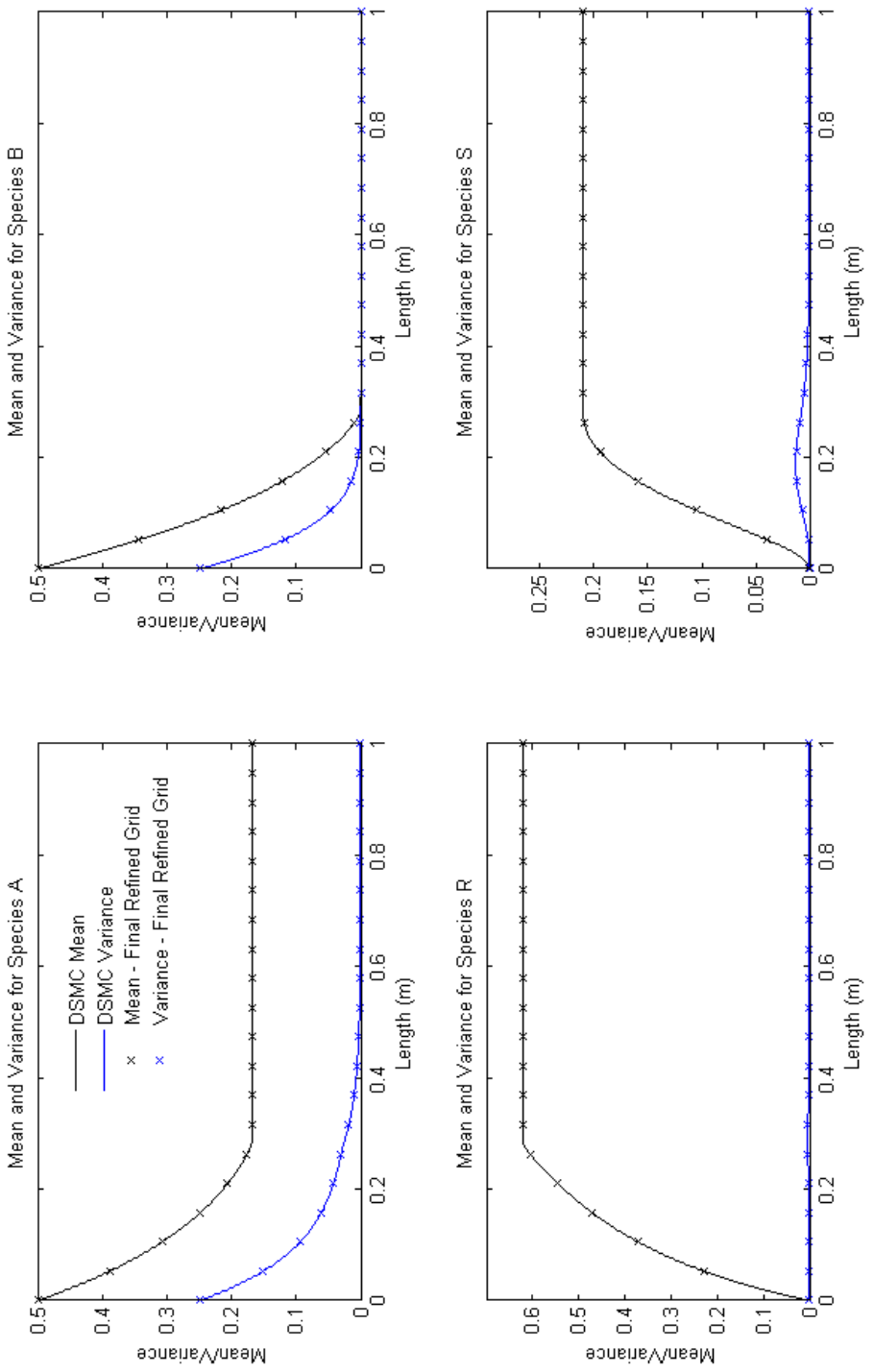

Figure 10: *Evolution of mean and variance with distance (Case 2).*

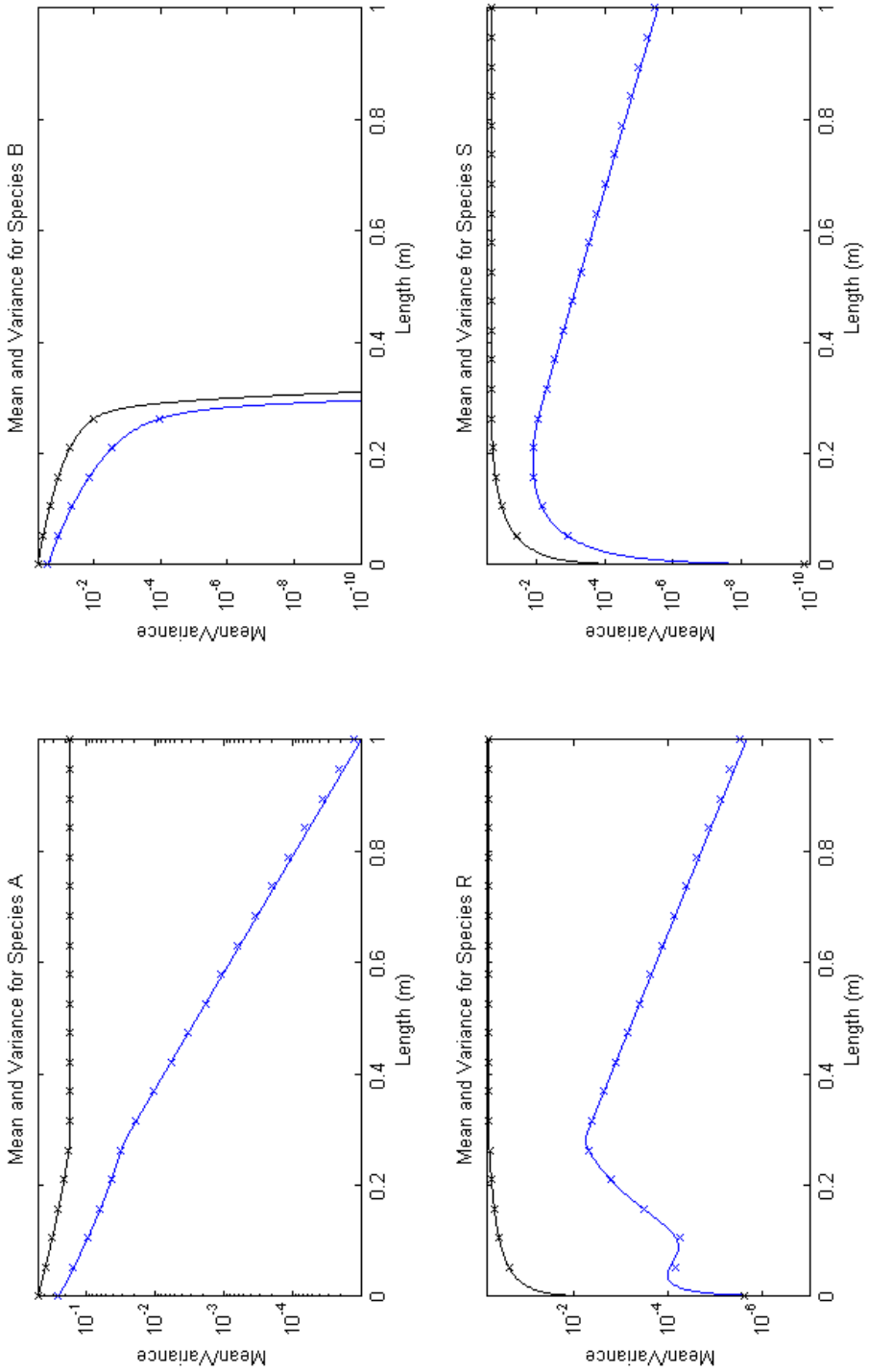

Figure 11: *Evolution of mean and variance with distance (Case 2). Legend same as for Figure 10.*

Table 1: *Initial conditions for the jet reactor.*

	First Moment		Second Moment	
Species	Jet	Outer Annulus	Jet	Outer Annulus
A	0	0	1	1
B	1	1	0	0
R	0	0	0	0
S	0	0	0	0

A graph of the yield of R at different initial concentrations of reactants and different Reynolds numbers is shown in Figure 12. It compares experimental data with the numerical data. The yield of R is given in Tsai and Fox (1994) as:

$$Y_R = \frac{\langle C_R \rangle}{\langle C_R \rangle + 2\langle C_S \rangle}. \tag{6.3}$$

The mean concentrations were found in Tsai and Fox (1994) by taking the average of the values over 20 time steps.

Although the right trend in the change of values is shown in Figure 12, the amount of R produced by the reaction is overpredicted. Grid refinement was attempted in the regions where the reactants A and B are found but no significant change in the results was observed. This still needs to be investigated, as it was found in the 1D test cases that the grid density was important. A method of grid refinement by refining the cells where the scalar gradients of R are large will be pursued.

It is also possible that the chemistry is not being solved for with sufficient accuracy. Some CFD codes such as KIVA solve the chemistry using high order integrators external to the CFD solver.

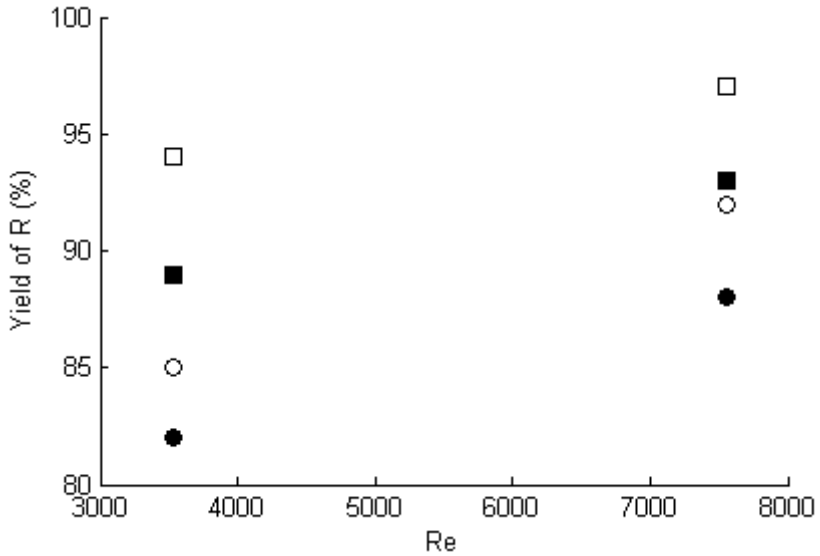

Figure 12: *Comparison between experimental results and numerical simulations for the yield of R. Square symbols are for higher initial concentration of C_B and round symbols are for lower initial concentration of C_B in the inlet stream. Shaded symbols represent experimental data from Li and Toor (1986) and hollow symbols represent the solutions from the CFD.*

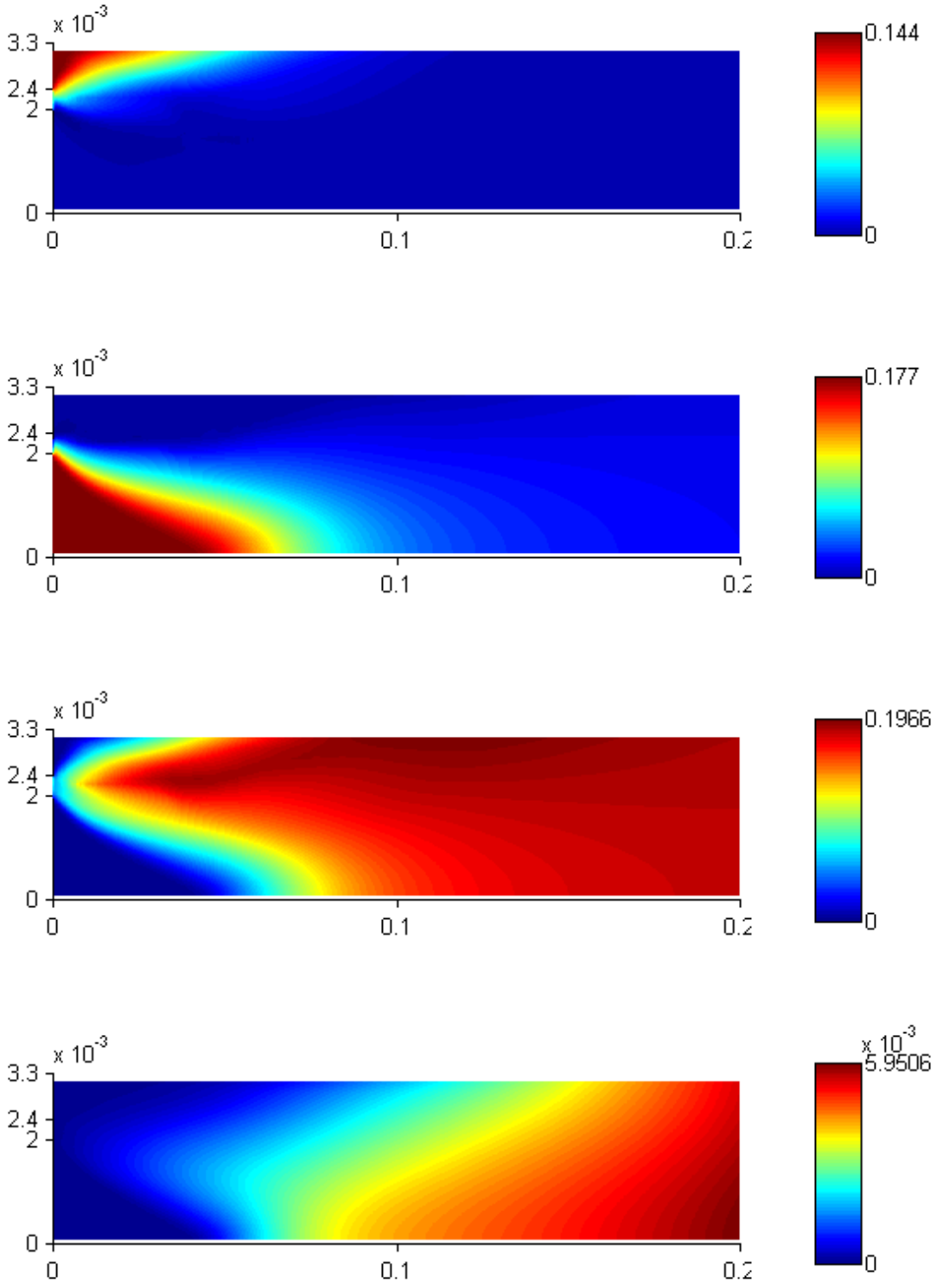

Figure 13: *Mean concentrations of A, B, R and S in the jet reactor.*

7 Future Work

7.1 Short Term

In the short term, the current deficiencies of the MoMEC implementation within STAR-CD need to be addressed. Adaptive grid refinement in the regions where the rate of production of species R are high will be investigated. It will also be instructive to investigate how other CFD codes such as KIVA solve fast chemistry; it may be necessary to decouple the solution of the chemistry from the transport of the moments. Possibilities include chemical look-up tables or an external integration routine. However, this approach would rather be avoided as it does not make use of the available CFD solvers.

Validation of the RANS and LES velocity fields against NMR experiments for the bubble column should be made in order to establish a working relationship and to see how the simulations and experiments can be compared in a quantitative manner. A population balance for the particle size distribution of bubbles will then be implemented to test coagulation and breakage models.

7.2 Long Term

The ideal scenario is to be able to solve large systems of complex chemical reactions within commercial CFD codes. To get to this stage flow modelling of turbulent, reactive, multiphase flows needs to be improved.

The use of DQMoM to solve population balances in Euler/Euler simulations appears promising. Work needs to be done on coupling mass transfer and reaction into two-phase flow models, so better mass transfer closures need to be formulated than are used currently. Another interesting area is how CFD models simulate flow regime transitions, which is an important requirement for any more general two-phase model. Most models are only tested in the homogeneous regime, so that the particle size distribution keeps a relatively constant shape. DQMoM will need to be examined under these conditions.

When MoMEC has been successfully tested in STAR-CD, it can be written in C++ (utilising the object-oriented capabilities of the language) and used within OpenFOAM.

A DQMoM algorithm is also close to being implemented in STAR-CD. In future this can also be ported to OpenFOAM and used to solve population balances. This will be useful as OpenFOAM has no population balance routines publicly available.

Nomenclature

$\mathbf{A}$	Matrix of Coefficients for all Nodes	
$\mathbf{b}$	Vector of Source Terms	
A	Cell Face Area	m^2
a_n	Coefficient for Node 'n'	
C_T	Total Concentration of Reactants and Products	kmol m^{-3}
C_α	Concentration of Species α	kmol m^{-3}
C_μ	Turbulent Viscosity Constant	
C_ϕ	Constant in IEM Mixing Model	
d_h	Hydraulic Diameter	m
F_n	Cell Face 'n'	
f_ϕ	Joint Composition Probability Density Function	
g	Gravitational Force	ms^{-2}
I	Turbulent Intensity	%
k	Turbulent Kinetic Energy	m^2s^{-2}
k_1	Reaction Rate of Reaction A + B $\rightarrow$ R	kmol^{-1} s^{-1}
k_2	Reaction Rate of Reaction B + R $\rightarrow$ S	kmol^{-1} s^{-1}
M_n	Relative Molecular Mass of Species 'n'	kg kmol^{-1}
m_n	Order of Moment of Species 'n'	
n_i	Component Normal	
p	Pressure	Pa
S	Source Terms	
S_C	Constant Part of Reaction Source Term	
S_P	Coefficient of Dependent Part of Reaction Source Term	
t	Time	s
U	Mean Velocity	ms^{-1}
u	Velocity	ms^{-1}
V	Cell Volume	m^3
w	Weight of Particle 'n'	
Y_R	Yield of R	

Symbols

$\langle\ \rangle$	Mean or Expectation of a Quantity
$\overline{\phi}$	Mean of a Quantity
ϕ'	Fluctuating Variable - RANS or Filtered Variable - LES
$\tilde{\phi}$	Resolved Variable - LES
$f()$	Function

Dimensionless Numbers

σ_T	Prandtl/Schmidt Number	
Re	Reynolds Number	

Greek Symbols

α	Volume Fraction	
Δ	Filter Width	
Δ_p	Perturbation Size	m
ε	Energy Dissipation Rate	$\mathrm{m^2 s^{-3}}$
Γ	Molecular Diffusivity	$\mathrm{m^2 s^{-1}}$
Γ_T	Eddy/Turbulent Diffusivity	$\mathrm{m^2 s^{-1}}$
λ	Moment	
ϕ	Sample Space Variable	
μ	Viscosity	Pa s
μ_T	Eddy/Turbulent Viscosity	Pa s
ϕ	Scalar	
ρ	Density	$\mathrm{kgm^{-3}}$
τ_ϕ	Scalar Timescale	s

Subscripts

α	Species
i, j, k	Directional Coordinate (1[x],2[y],3[z])

Superscripts

$+$	Positive Perturbation
$-$	Negative Perturbation
n	Particle Number

References

Batchelor, G. K. (1981). *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge, UK.

Bhole, M. R., Joshi, J. B. and Ramkrishna, D. (2008). Cfd simulation of bubble columns incorporating population balance modeling, *Chemical Engineering Science* **63**(8): 2267–2282.

Cant, R. S. and Mastorakos, E. (2008). *An Introduction to Turbulent Reacting Flows*, Imperial College Press, London, UK.

Crowe, C., Sommerfeld, M. and Tsuji, Y. (1998). *Multiphase Flows with Droplets and Particles*, CRC Press, Florida, USA.

Darmana, D., Henket, R. L. B., Deen, N. G. and Kuipers, J. A. M. (2007). Detailed modelling of hydrodynamics, mass transfer and chemical reactions in a bubble column using a discrete bubble model: Chemisorption of co_2 into naoh solution, numerical and experimental study, *Chemical Engineering Science* **62**: 2556–2575.

Deen, N. G., Hjertager, B. H. and Solberg, T. (2000a). Comparison of piv and lda measurement methods applied to the gas-liquid flow in a bubble column, *10th Intl. Symp. on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal*.

Deen, N. G., Solberg, T. and Hjertager, B. H. (2000b). Numerical simulation of the gas-liquid flow in a square cross-sectioned bubble column, *CHISA 14th Intl. Congress of Chemical and Process Engineering, Praha, Czech Republic*.

Deen, N. G., Solberg, T. and Hjertager, B. H. (2001). Large eddy simulation of the gas-liquid flow in a square cross-sectioned bubble column, *Chemical Engineering Science* **56**: 6341–6349.

Deen, N. G., van Sint Annaland, M. and Kuipers, J. A. M. (2004). Multi-scale modeling of dispersed gas-liquid two-phase flow, *Chemical Engineering Science* **59**: 1853–1861.

Douglas, C. C. (2003). Multigrid net. http://www.mgnet.org.

Eibeck, A. and Wagner, W. (2001). Stochastic particle approximations for smoluchowski's coagulation equation, *The Annals of Applied Probability* **11**(4): 1137–1165.

Fan, R., Marchisio, D. L. and Fox, R. O. (2004). Application of the direct quadrature method of moments to polydisperse gas-solid fluidized beds, *Powder Technology* **139**: 7–20.

Fox, R. O. (2003). *Computational Models for Turbulent Reacting Flows*, Cambridge University Press, Cambridge, UK.

Issa, R. I. (1985). Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of Computational Physics* **62**: 40–65.

Jakobsen, H. A., Sannaes, B. H., Grevskott, S. and Svendsen, H. F. (1997). Modeling of vertical bubble-driven flows, *Ind. Eng. Chem. Res.* **36**(10): 4052–4074.

Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method with Applicaitons to Fluid Flows*, PhD thesis, Imperial College.

Karrholm, F. P. (2006). Rhie-chow interpolation in openfoam. http://www.tfd.chalmers.se/hani/kurser/OF_PhD_2007/rhiechow.pdf.

Kay, J. M. and Nedderman, R. M. (1985). *Fluid Mechanics and Transfer Processes*, Cambridge University Press, Cambridge, UK.

Kumar, S. and Ramkrishna, D. (1996). On the solution of population balance equations by discretization - 1. a fixed pivot technique, *Chemical Engineering Science* **51**(8): 1311–1332.

Lain, S., Broder, D., Sommerfeld, M. and Goz, M. F. (2002). Modelling hydrodynamics and turbulence in a bubble column using the euler-lagrange procedure, *International Journal of Multiphase Flow* **28**: 1381–1407.

Li, K. T. and Toor, H. L. (1986). Turbulent reactive mixing with a series-parallel reaction: Effect of mixing on yield, *AICHE Journal* **32**(8): 1312–1320.

Monahan, S. M. and Fox, R. O. (2007). Effect of model formulation on flow-regime predictions for bubble columns, *AIChE Journal* **53**(1): 9–18.

Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, New York.

Pope, S. B. (2000). *Turbulent Flows*, Cambridge University Press, Cambridge, UK.

Shewchuk, J. R. (1994). An introduction to the conjugate-gradient method without the agonizing pain, *Tech. Rep. CMU-CS-94-125 (Carnegie Mellon University)*.

Simonnet, M., Gentric, C., Olmos, E. and Midoux, N. (2007). Cfd simulation of the flow field in a bubble column reactor: Importance of the drag force formulation to describe regime transitions, *Chemical Engineering and Processing*.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations i. the basic experiment, *Monthly Weather Review* **91**(3): 99–164.

Sokolichin, A. and Eigenberger, G. (1997). Dynamic numerical simulation of gas-liquid two-phase flows: Euler/euler versus euler/lagrange, *Chemical Engineering Science* **52**(4): 611–626.

Tryggvason, G. (2001). A front-tracking method for the computations of multiphase flow, *Journal of Computational Physics* **169**: 708–759.

Tsai, K. and Fox, R. O. (1994). Pdf simulation of a turbulent series-parallel reaction, *Chemical Engineering Science*.

Versteeg, H. K. and Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Pearson Education Limited, Harlow, UK.

Vikhansky, A. and Kraft, M. (2005). Single-particle method for stochastic simulation of coagulation processes, *Chemical Engineering Science* **60**(4): 963–967.