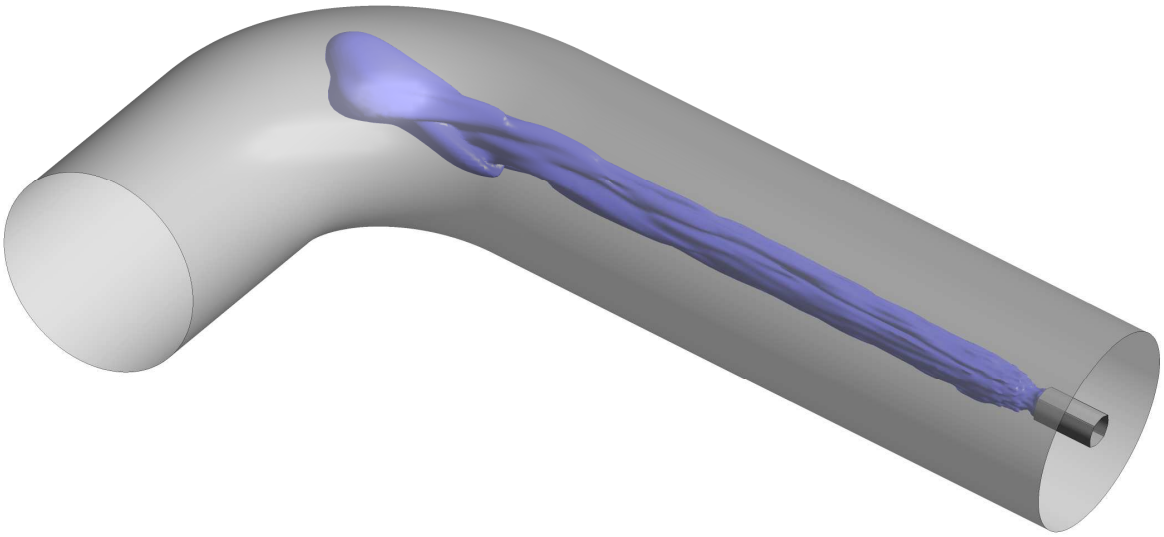
CHALMERS



Solving stiff systems of ODEs in CFD applications

Developing and evaluating the *Stiff ODE Suite*

Master of Science Thesis

HENRIK ALFREDSSON

Department of Chemical and Biological Engineering
Division of Chemical Reaction Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2010

Abstract

In common engineering applications such as fast equilibrium reactions, catalysis, adsorption and evaporation the solution of systems of stiff ordinary differential equations is of key importance. In an attempt to solve such systems of equations encountered in CFD applications a standalone software module, Stiff ODE Suite, was developed and coupled to the commercial CFD code FLUENT v12.1, Ansys inc. The functionality of the hybrid software package was validated for usage in common engineering applications.

The Stiff ODE Suite software package is based on an adaptive time step maximum fifth order Backwards Differential Formulation. Said algorithm introduces a validated and versatile stiff and non-stiff ODE solver potentially outperforming MATLABs ODE15s. Functioning as a standalone module the Stiff ODE Suite is however best applied in a hybrid scheme with FLUENT v12.1 or other compatible CFD codes. Such hybrid couplings vastly increase the applicability of otherwise limited CFD codes.

The Stiff ODE Suite was used to with satisfactory accuracy predict solution pH in neutralization of hydrochloric acid using a sodium carbonate solution. The water auto proteolysis is an extremely fast equilibrium reaction posing a significant challenge to any chemical reaction engineering solver. The Stiff ODE Suite also enabled the simulation of rapid humidification of dry heated air using a water droplet jet in a pipe segment. In this simulation the Stiff ODE Suite proved capable of coupling fast mass and heat transfer between phases in multiphase applications.

By introduction of the Stiff ODE Suite the CFD engineer is able to use much larger time steps in transient simulations than the time scale of the stiff system of ODEs governing the solution. In addition the software module largely extends the applicability of FLUENT v12.1 since the user is no longer limited to what can be described using the graphical user inter phase.

Contents

1. Introduction

1.1 Background

Most phenomena in nature are mathematically governed by differential equations of some form. In reaction engineering and catalysis the governing equations frequently encountered are autonomous ordinary differential equations on the form presented in equation 1.1.

$$\dot{y} = f(y), \qquad y(t_0) = y_0, \quad y \in \vec{R}^N \tag{1.1}$$

Numerous systems of ordinary differential equations, ODEs, can be found which are stiff in nature meaning that the system is governed by time scales of vastly different sizes. To exemplify the problem formulation assume the existence of a system of equations on the form presented in equation 1.2.

$$\dot{y} = ky, \qquad y(t_0) = y_0, \quad y \in \vec{R}^N \tag{1.2}$$

With $y(t_0) = 1$ and $t_0 = 0$ the solution to equation 1.2 is $y = e^{kt}$. In figure 1.1 and 1.2 the solution to equation 1.2 with k=1 and k=5000 respectively is shown.

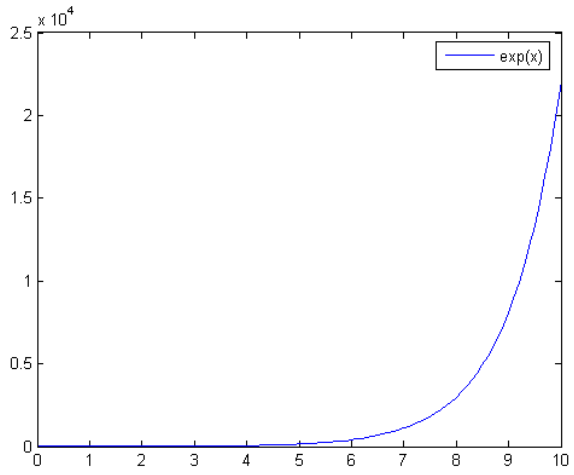


Figure 1.1 $y = e^{t}$

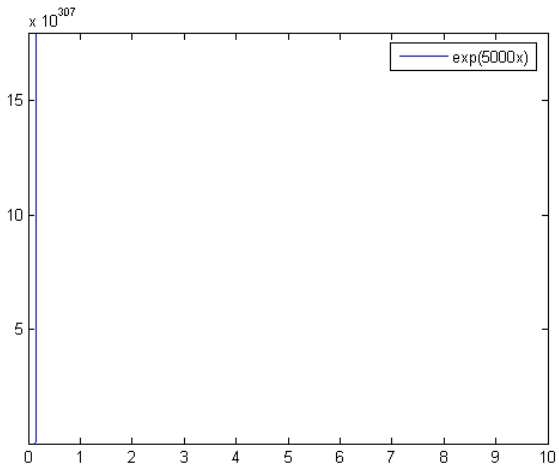


Figure 1.2 $y = e^{5000t}$

Already at moderate values of k the solution to the simple system 1.2 depicts rapid growth. At k=5000 machine precision is lost and the solution grows rapidly towards infinity. In naturally occurring systems such as liquid phase chemical equilibria or fast evaporation sequences it is not uncommon to encounter $k=10^{20}$ which hence introduces numerical issues in modelling engineering applications and especially in CFD simulations.

The commercial CFD software FLUENT v12.1 uses either explicit or simple implicit algorithms when solving stiff systems of ODEs. Using explicit algorithms is a time consuming ordeal whereas the simple implicit algorithm has been shown to converge slowly and inefficiently and with systems of considerable stiffness the simulation has a tendency to crash altogether.

1.2 Objective and method

The objective of the thesis is to produce a reliable methodology to solving stiff systems of ODEs in the FLUENT v12.1 framework and then to evaluate the applicability of the methodology in said framework. In order to be practically useful the methodology needs to take advantage of the benefits of multi processing as introduced by modern day cluster computers. The methodology also needs to be computationally efficient and problem adaptable.

The proposed methodology to solving stiff and non-stiff systems of ODEs in the FLUENT v12.1 framework is the Stiff ODE Suite. This stand alone module is based on a commonly available solution algorithm and written in the programming language C. By using the benefits of the FLUENT source code the Stiff ODE Suite can be used in any CFD problem formulation only limited to what can be described using ordinary differential equations. The applicability of the inter software coupling is tested in modelling extremely fast equilibrium reactions when neutralizing strong hydrochloric acid using a sodium carbonate solution. Furthermore the ability to model the previously troublesome area of fast transient evaporation sequences is tested in simulating humidification of dry heated air using a water droplet jet.

1.3 Thesis outline

The thesis is essentially divided into two parts. In part one fundamental theory needed to grasp used concepts and notations is outlined along with the problem formulations used to evaluate the applicability of the Stiff ODE Suite. In chapter 2 basic theory behind Computational Fluid Dynamics is presented along with turbulence modelling and the Eulerian-Lagrangian approach to multi phase flows. Furthermore the mathematical algorithm used in the Stiff ODE Suite is outlined. In chapter 3.1 the Stiff ODE Suite is evaluated for simulation of fast equilibrium reactions more specifically neutralization of hydrochloric acid using a sodium carbonate solution. Chapter 3.2 covers the simulation and problem formulation behind fast evaporation sequences exemplified in humidification of dry heated air using a water droplet jet.

In part two results from performed simulations are presented along with a detailed discussion on interesting phenomena and the performance of the Stiff ODE Suite. Specifically chapter 4 presents a short description of characteristic graphical results obtained from the simulations whereas chapter 5 is dedicated to discussing the thesis results.

PART I

2. Theory

2.1 Eulerian-Lagrangian Particle Tracking

The Eulerian-Lagrangian framework otherwise known as the Discrete Particle Model or Discrete Element Model treats individual particles or clouds of particles as isolated entities being tracked using the fundamental theory behind Newtons´ second law of motion. In essence this means that the entity, cloud or particle is tracked by solving equation 2.1.

$$m_p \frac{d^2 x}{dt^2} = \sum F_i = F_{drag} + F_{bouyancy} + F_{saffman} + F_{magnus} + F_{addedmass} + F_{historyforce} + F_{pressure} + \ldots \quad (2.1)$$

$$F_{drag} = \frac{1}{2}\rho_f C_D A \left|\vec{u}_f - \vec{u}_p\right| \left(\vec{u}_f - \vec{u}_p\right) \quad (2.2)$$

$$F_{bouyancy} = \vec{g} V_p \left(\rho_p - \rho_f\right) \quad (2.3)$$

$$F_{saffman} = 1.61 d_p^2 \sqrt{\rho_f \mu_f} \frac{\left(\vec{u}_f - \vec{u}_p\right) \times \vec{\omega}_f}{\sqrt{\left|\vec{\omega}_f\right|}} \quad (2.4)$$

$$F_{magnus} = \frac{\pi}{8} d_p^3 \rho_f \left(\frac{1}{2}\nabla \times \vec{u}_f - \vec{\omega}_p\right) \times \left(\vec{u}_f - \vec{u}_p\right) \quad (2.5)$$

$$F_{addedmass} = \frac{1}{2} m_p \frac{\rho_f}{\rho_p} \frac{d}{dt}\left(\vec{u}_f - \vec{u}_p\right) \quad (2.6)$$

$$F_{historyforce} = \sqrt{\pi \rho_f \mu_f} \frac{m_p}{\rho_p d} \int_0^t \frac{1}{\sqrt{1-\tau}} \frac{d}{dt}\left(\vec{u}_f - \vec{u}_p\right) d\tau \quad (2.7)$$

$$F_{pressure} = m_p \frac{\rho_f}{\rho_p}\left(\frac{D\vec{u}_f}{Dt} - \vec{g}\right) \quad (2.8)$$

In equation 2.1 only the most commonly applicable forces are outlined leaving the interested reader to refer to Crowe et al for additional information. The Eulerian-Lagrangian framework is the least simplified theoretical approach to multiphase flows limited only by the applied turbulence model and the description of applicable forces. In theory it is possible to track irregularly shaped objects of any material with vastly different sizes flowing in any geometry modelling, amongst other things, collisions, coalescence and droplet breakup. There is however a trade off between computational accuracy and computational time and in the specific problem formulation in section 3.2 several simplifications have been made.

In section 3.2 the Eulerian-Lagrangian framework is used tracking computational parcels (particle clouds) instead of individual particles. This approach can be used assuming

- All particles in a parcel are identical and only reside in one computational cell
- The volume fraction of particles is small enough to be neglected, <10%
- Particles colliding is a very rare event hence being neglected

Tracking computational parcels is hence a very powerful method able of economically predicting exchange of mass, heat and momentum between the carrier phase and the discrete phase. [1]

2.1.1 Discrete Random Walk

In the Discrete Random Walk approach to modelling subgrid scale turbulent dispersion of particles the following assumptions are made. Subgrid turbulence, as in the case of using DRW in a LES approach, can be regarded as Gaussian distributed random fluctuations with an eddy life time τ_e. It should be noted that the LES approach resolves the large scale turbulent fluctuations leaving only the subgrid fluctuations to be taken into account in the DRW modelling.

The random velocity fluctuations are calculated according to equation 2.9. In the LES approach to subgrid turbulence the subgrid turbulent kinetic energy is calculated according to $k_{subgrid} = 0.5\left(\overline{u_j^2} - \overline{u}_j^{\,2}\right)$ and hence random subgrid components are generated using equation 2.9.

$$u_i' = \varsigma\sqrt{\overline{u_i}^2} = \varsigma\sqrt{\frac{2k_{subgrid}}{3}} \tag{2.9}$$

The particle is affected by the additional random component calculated in equation 2.9 for the duration of the smaller of a) the turbulent eddy life time and b) the time it takes for the particle to transverse the eddy. This is what is essentially shown in equation 2.10.

$$\tau_e = \min\left(2C_L\frac{l_{cell}}{\overline{u}}, \tau_p \log\left(1 - \frac{L_e}{\tau_p\left(u' - u_p\right)}\right)\right) \tag{2.10}$$

In this context L_e denotes the subgrid eddy length scale resolved in the LES approach. τ_p is the particle relaxation time scale. C_L is the integral time scale constant set by the user. An interesting feature of equation 2.10 is that FLUENT uses the residence time of a fluid element in the computational cell as an approximation of the LES subgrid eddy lifetime. [2] A better approximation would clearly be to model said timescale but since 80% of the turbulent kinetic energy is resolved directly in the LES approach the subgrid approximation in equation 2.10 only accounts for a small part of the particle random movement. Equation 2.10 hence seems viable.

2.1.2 Transient particle tracking in FLUENT

Internally in FLUENT there are essentially three methods of tracking particles in the Eulerian-Lagrangian framework. The most applicable and commonly used approach is to transiently track particles moving through a transiently resolved carrier phase. There is however the option to transiently track particles moving through a steady state flow field. Furthermore there is also the option to track particles in a steady state mode moving through a steady state flow field. This chapter will be devoted to describing the least simplified method being the transient approach where time resolved particles move through an unsteady flow field.

Fundamentally the FLUENT iteration sequence in the Eulerian-Lagrangian framework is subdivided into two regimes being a) carrier phase iterations and b) particle updating sequence. During the particle updating sequence the particles are tracked individually by time of birth. This essentially means that when a particle is born at an injection nozzle per say the particle is given a unique identification number resulting in particles with a long residence time having a lower identification number than the particles with a shorter residence time. When the particle updating sequence is commenced the particle with the lowest identification number will be tracked through the carrier phase solving equation 2.1 using preferably an implicit Runge-Kutta or trapezodial formulation with a user adaptable particle time step. Regardless of the user specified time step the particle will never move through more than one cell per particle time step. If the user defined particle time step is too large FLUENT will automatically decrease it in order for each particle to be updated in every cell along its trajectory. Once the particle has been tracked a sufficient number of steps the carrier phase time step is reached and the next particle in terms of identification number is tracked.

In each cell along the trajectory of the particle the tracking sequence is temporarily halted and a set of predefined functions are executed. In effect this means that amongst other things inter phase mass and heat equations can be solved in each cell along the particle trajectory. The exchange of heat and mass between particles and the carrier phase are essentially solved in the same manner as chemical reactions ie FLUENT resolves heat and mass transfer rates and assumes these transfer rates to be constant during the time step at hand. This particular solution method is fast and easily implemented but useless when dealing with stiff physical systems. Experience shows that when simulating unsteady fast evaporation of water droplets FLUENTs internal models will predict cell temperatures below the condensation temperature of air.

Another drawback with the internal models in FLUENT particle tracking framework is the manner in which the cell properties are updated. In the vicinity of the injection nozzle the very same computational cell will witness multiple particles passing through during one carrier phase time step. The first in the sequence of particles to pass through the cell will correctly experience prevailing cell conditions in the evaporation sequence. The cell properties will however not be updated until the end of the particle tracking sequence leaving particle two to pass through the cell to experience the exact same conditions as particle one. This is a fundamentally erroneous description of the events occurring on a cell level disabling the transient solution of important properties such as particle-cell equilibrium.

Discussing this fundamental problem with the FLUENT source code developers did however shed some light on the issue. The internal particle tracking algorithm was originally developed for steady state simulation purposes which in a sense explains the lack of applicability in transient simulations. It should however be noted that inter phase mass transfer of species with a lower heat of vaporization than water are inherently easier to model.[2]

2.1.3 Rosin-Rammler particle distribution

A commonly used particle distribution in modelling droplet sprays is the Rosin-Rammler size distribution. Equation 2.11 outlines the mass fraction of droplets with diameter greater than diameter d.

$$Y_d = e^{\left(\frac{d}{\bar{d}}\right)^n} \tag{2.11}$$

In equation 2.11 $\bar{d}$ is the particle mean diameter whereas n is a model specific size distribution parameter best determined from experimental data. [2]

2.2 Basic theory of Computational Fluid Dynamics

The following section is not intended to fully cover the fundamental theory of fluid mechanics. Instead the intention is to present short review of the fundamental concepts behind Computational Fluid Dynamics. Instead the reader is assumed to have an appropriate academic background in the field or is referred to Andersson et al and Welty et al.

The general transport equation of a scalar, vector or tensor is presented in continuous form in equation 2.12 and in discretized form in equation 2.13.

$$\frac{\delta\phi}{\delta t} + U_i \frac{\delta\phi}{\delta x_i} = \frac{\delta}{\delta x_i}\left(\Gamma \frac{\delta\phi}{\delta x_i}\right) + S(\phi) \tag{2.12}$$

$$\int_{CV} \frac{\delta\phi}{\delta t} dV + \int_{CS} U_j \vec{n} \phi dA = \int_{CS} \Gamma \frac{\delta\phi}{\delta x_i} \vec{n} dA + \int_{CV} S_\phi dV \tag{2.13}$$

CV denotes the discretized control volume and CS the area surrounding said control volume. Equation 2.12 can be extended into the Navier Stokes equations by setting $\phi_i = \rho U_i$ and specifying the source term $S(\phi)$ yielding equation 2.14.

$$\frac{\delta U_i}{\delta t} + U_j \frac{\delta U_i}{\delta x_j} = -\frac{1}{\rho}\frac{\delta P}{\delta x_i} - \frac{1}{\rho}\frac{\tau_{ji}}{\delta x_j} + g_i \tag{2.14}$$

The discretized form of the Navier Stokes equations along with the equation of continuity outlined in equation 2.15 form the basis of any Computational Fluid Dynamics simulation.

$$\frac{\delta\rho}{\delta t} + \frac{\delta(\rho U_j)}{\delta x_j} = 0 \tag{2.15}$$

Setting $\phi = C$ in equation 2.12 yields the species transport equation 2.16 whilst setting $\phi = \rho C_p T$ yields the temperature equation 2.17. [3]

$$\frac{\delta C}{\delta t} + U_i \frac{\delta C}{\delta x_i} = \frac{\delta}{\delta x_i}\left(D \frac{\delta C}{\delta x_i}\right) + S(C) \tag{2.16}$$

$$\frac{\delta \rho C_p T}{\delta t} + U_i \frac{\delta \rho C_p T}{\delta x_i} = \frac{\delta}{\delta x_i}\left(\lambda \frac{\delta \rho C_p T}{\delta x_i}\right) - P\frac{\delta U_j}{\delta x_j} + \tau_{kj}\frac{\delta U_k}{\delta x_j} + S(T) \tag{2.17}$$

In the Eulerian-Lagrangian approach to multiphase flow outlined in section 2.1 the equation of motion, the species transport equation and the equation of continuity are augmented with an additional source term resulting from the inter phase transport of momentum and mass outlined in equation 2.18-2.20.

$$\frac{\delta U_i}{\delta t} + U_j \frac{\delta U_i}{\delta x_j} = -\frac{1}{\rho}\frac{\delta P}{\delta x_i} - \frac{1}{\rho}\frac{\tau_{ji}}{\delta x_j} + g_i + S_i(U_j, U_{particle}) \tag{2.18}$$

$$\frac{\delta C}{\delta t} + U_i \frac{\delta C}{\delta x_i} = \frac{\delta}{\delta x_i}\left(\Gamma \frac{\delta C}{\delta x_i}\right) + S(C) \tag{2.19}$$

$$\frac{\delta \rho}{\delta t} + \frac{\delta(\rho U_j)}{\delta x_j} = S(C) \tag{2.20}$$

Equation 2.18 through 2.20 are valid under the assumption of negligible particle volume fraction as imposed by tracking computational parcels. [1]

2.3 Turbulence modelling

However theoretically possible the direct numerical solution (DNS) to the Navier Stokes equations 2.14 resolving mixing down to Kolmogorov scale is still practically impossible in engineering applications. Instead turbulence- and closure models have been mathematically developed, implemented and tested in order to increase the modelling possibilities in CFD.

2.3.1 Reynolds Averaged Navier Stokes equations (RANS)

Modelling turbulence using Reynolds Averaged Navier Stokes equations implies the fundamental assumption of averaging pressure and velocity. The Reynolds decomposition reduces the instantaneous variable into a time averaged mean and a fluctuating component according to equation 2.21 and 2.22. The timescale on which the averaging is performed is deduced from local flow variables.

$$U_i = \langle U_i \rangle + u_i \tag{2.21}$$

$$P = \langle P \rangle + p \tag{2.22}$$

Inserting the Reynolds averaged values into Navier Stokes equation 2.14 and assuming incompressible Newtonian flow yields equation 2.23.

$$\frac{\delta\left(\langle U_i \rangle + u_i\right)}{\delta t} + \left(\langle U_j \rangle + u_j\right)\frac{\delta\left(\langle U_i \rangle + u_i\right)}{\delta x_j} = -\frac{1}{\rho}\frac{\delta\left(\langle P \rangle + p\right)}{\delta x_i} + \nu\frac{\delta^2\left(\langle U_i \rangle + u_i\right)}{\delta x_i \delta x_j} \tag{2.23}$$

Rearranging and defining the Reynolds stress tensor $\tau_{ij} = -\rho\langle u_i u_j \rangle$ equation 2.23 evolves into the Reynolds Averaged Navier Stokes equations (RANS) outlines in equation 2.24.

$$\frac{\delta\langle U_i \rangle}{\delta t} + \langle U_i \rangle\frac{\delta\langle U_i \rangle}{\delta x_j} = -\frac{1}{\rho}\frac{\delta}{\delta x_j}\left(\langle P \rangle\delta_{ij} + \mu\left(\frac{\delta\langle U_i \rangle}{\delta x_j} + \frac{\delta\langle U_j \rangle}{\delta x_i}\right) - \rho\langle u_i u_j \rangle\right) \tag{2.24}$$

The Reynolds stresses represent the average momentum flux due to velocity fluctuations and introduce a viable coupling between the mean velocity field and the fluctuating component. In all RANS turbulence models the main closure problem arises in introducing the Reynolds stresses and hence has to be modelled.

A large portion of the RANS turbulence models proposed are turbulent viscosity models relying on the existence of turbulent viscosity. The principal assumption is that of isotropic turbulence implying that the flow is statistically unchanged if all positions are shifted by the same displacement and invariant under rotation. Furthermore a major assumption lies within the Boussinesq approximation outlined in equation 2.25. The Boussinesq approximation essentially states that the Reynolds Stresses are proportional to the mean velocity gradients implying and fundamentally assuming that turbulent momentum flux can be modelled as diffusion transport due to local mean velocity gradients.

$$\frac{\tau_{ij}}{\rho} = -\langle u_i u_j \rangle = \nu_T \left(\frac{\delta \langle U_i \rangle}{\delta x_j} + \frac{\delta \langle U_j \rangle}{\delta x_i} - \frac{2}{3} k \delta_{ij} \right) \tag{2.25}$$

With the introduction of the turbulent viscosity yet another closure problem arises and closed differently in various turbulent viscosity models. All models however use local appropriate length and velocity scales to calculate the turbulent viscosity. With the assumptions introduced in the Boussinesq approximation and with the introduction of the turbulent viscosity the RANS equation can be extended into equation 2.26.

$$\frac{\delta \langle U_i \rangle}{\delta t} + \langle U_j \rangle \frac{\delta \langle U_i \rangle}{\delta x_j} = -\frac{1}{\rho} \frac{\delta \langle P \rangle}{\delta x_i} - \frac{2}{3} \frac{\delta k}{\delta x_i} + \frac{\delta}{\delta x_j} \left[(\nu + \nu_T) \left(\frac{\delta \langle U_i \rangle}{\delta x_j} + \frac{\delta \langle U_j \rangle}{\delta x_i} \right) \right] \tag{2.26}$$

In equation 2.26 the turbulent kinetic energy k has been introduced according to equation 2.27.

$$k = \frac{1}{2} \langle u_i u_i \rangle \tag{2.27}$$

The Realizable k-epsilon model is a two equation model based on the standard k-epsilon model with the additional advantage of being most suitable for flows including streamline curvature and rotation. Specifically the Realizable k-epsilon model augments the Boussinesq approximation in modelling the Reynolds stresses as shown in equation 2.28.

$$\langle u_i u_j \rangle = \frac{2}{3} k - 2\nu_T \frac{\delta \langle U_i \rangle}{\delta x_j} \tag{2.28}$$

With the final formulation of the RANS equation in 2.26 three parameters need be modelled to fully close the Realizable k-epsilon model. In equation 2.29 through 2.31 turbulent viscosity, turbulent kinetic energy and turbulent energy dissipation are modelled. [3]

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \tag{2.29}$$

$$\frac{\delta k}{\delta t} + \langle U_j \rangle \frac{\delta k}{\delta x_j} = \nu_T \left[\frac{\delta \langle U_i \rangle}{\delta x_j} \left(\frac{\delta \langle U_i \rangle}{\delta x_j} + \frac{\delta \langle U_j \rangle}{\delta x_i} \right) \right] - \varepsilon + \frac{\delta}{\delta x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\delta k}{\delta x_j} \right] \tag{2.30}$$

$$\frac{\delta \varepsilon}{\delta t} + \langle U_j \rangle \frac{\delta \varepsilon}{\delta x_j} = C_{\varepsilon 1} \frac{\varepsilon}{k} \nu_T \left[\frac{\delta \langle U_i \rangle}{\delta x_j} \left(\frac{\delta \langle U_i \rangle}{\delta x_j} + \frac{\delta \langle U_j \rangle}{\delta x_i} \right) \right] - C_{\varepsilon 2} \frac{\varepsilon^2}{k} + \frac{\delta}{\delta x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\delta \varepsilon}{\delta x_j} \right] \tag{2.31}$$

2.3.2 Large Eddy Simulation

Turbulent flows are characterized by turbulent eddies on a wide range of length and time scales. The smallest eddies are responsible for viscous dissipation of turbulent energy on a length scale below which no turbulence can exist (Kolmogorov scale). The largest eddies

however contain most of the total turbulent energy and are typically on a length scale characterized by the mean flow. In order to fully and accurately describe the turbulent flow behaviour the entire spectrum of turbulent scales has to be modelled ie performing a Direct Numerical Simulation. Time is however finite and in engineering applications it is with the computational power of today seldom or never feasible to perform exact unsimplified simulations. Instead a method where the large eddies are directly resolved whereas the eddies on the smallest of length scales are modelled has been proposed in the Large Eddy Simulation (LES) framework.

The LES formulation hence has an increased ab initio predictive power compared to RANS but is still inferior to DNS. Practical experience from experiments and a large amount of simulations has shown that momentum, mass and energy are mostly transported by large eddies. Large eddies also tend to be more problem dependent largely affecting the overall simulation result. Small eddies on the other hand tend to be less dependent on geometry, more isotropic and it has been shown to be easier to find a universal turbulence model for small eddies compared to large ones. All these characteristics favour the formulation of the LES turbulence model but it should be firmly noted that the predictive power of LES does not come for free. A very fine grid has to be used at the same time as a considerable amount of computational power has to be used in order to simulate a sufficient time span as to resolve reliable flow statistics. This makes LES several orders of magnitude more expensive than RANS.

The governing equations in LES are obtained by filtering the Navier Stokes equations effectively filtering out eddies whose scale are smaller than either the computational grid or the filter width. In equation 2.32 the implicit filtration as imposed by the discretization of a continuous domain is shown. This should be compared with the filtering operator shown in equation 2.33-2.34.

$$\bar{\phi}(\vec{x}) = \frac{1}{V}\int_V \phi(\vec{x}')d\vec{x}' \tag{2.32}$$

$$\bar{\phi}(\vec{x}) = \int_D \phi(\vec{x}')G(\vec{x},\vec{x}')d\vec{x}' \tag{2.33}$$

$$G(\vec{x},\vec{x}') = \begin{cases} 1/V, \vec{x}' \in V \\ 0, \vec{x}' \notin V \end{cases} \tag{2.34}$$

The filtered Navier Stokes equations thereby obtained are along with the continuity equation outlined in equation 2.35 and 2.36.

$$\frac{\delta \rho}{\delta t} + \frac{\delta}{\delta x_i}\left(\rho \bar{u}_i\right) = 0 \tag{2.35}$$

$$\frac{\delta}{\delta t}\left(\rho \bar{u}_i\right) + \frac{\delta}{\delta x_j}\left(\rho \bar{u}_i \bar{u}_j\right) = \frac{\delta}{\delta x_j}\left(\sigma_{ij}\right) - \frac{\delta \bar{p}}{\delta x_i} - \frac{\tau_{ij}}{\delta x_j} \tag{2.36}$$

The stress tensor due to molecular viscosity, σ_{ij}, and the subgrid scale stress tensor, τ_{ij}, are hence introduced and defined in equation 2.37 and 2.38 respectively.

$$\sigma_{ij} = \left[\mu \left(\frac{\delta \bar{u}_i}{\delta x_j} + \frac{\delta \bar{u}_j}{\delta x_i} \right) \right] - \frac{2}{3} \mu \frac{\delta \bar{u}_i}{\delta x_i} \delta_{ij} \tag{2.37}$$

$$\tau_{ij} = \rho \overline{u_i u_j} - \rho \bar{u}_i \bar{u}_j \tag{2.38}$$

The subgrid stress tensor as introduced by the filtering operation needs closing. In the LES framework this is done using the Boussinesq approximation resulting in a subgrid stress tensor formulation as shown in equation 2.39.

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2 \mu_T \bar{S}_{ij} \tag{2.39}$$

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\delta \bar{u}_i}{\delta x_j} + \frac{\delta \bar{u}_j}{\delta x_i} \right) \tag{2.40}$$

Assuming the applicability of the Boussninesq approximation inherently introduces the concept of turbulent viscosity used at turbulent length scales below the filtering width. The LES framework allows for several methods of approximating the turbulent viscosity of which the perhaps most commonly used is the Smagorinsky-Lilly model. In the Smagorinsky-Lilly model the turbulent viscosity is modelled as shown in equation 2.41 through 2.44.

$$\mu_T = \rho L_S^2 \left| \bar{S} \right| \tag{2.41}$$

$$\left| \bar{S} \right| = \sqrt{2 \bar{S}_{ij} \bar{S}_{ij}} \tag{2.42}$$

$$L_S = \min(\kappa d, C_S \Delta) \tag{2.43}$$

$$\Delta = V_{cell}^{1/3} \tag{2.44}$$

L_S denoted the mixing length of subgrid scales, κ the von Kármán constant, C_S the Smagorinsky constant and Δ the characteristic cell length. The obvious shortcoming of this model is the existence of the universal Smagorinsky constant which has to be adapted to the problem at hand which reduces the ab initio predictable power of the Smagorinsky-Lilly subgrid model. For common engineering applications however a value of 0.17 has been suggested and shown to predict isotropic subgrid turbulence with acceptable accuracy.

There other approaches to modeling the LES subgrid turbulence. Two worth mentioning are the Dynamic Smagorinsky-Lilly Model and the Dynamic Kinetic Energy Subgrid Scale Model. In the Dynamic Smagorinsky-Lilly Model the Smagorinsky constant C_s is calculated from the resolved scales of motion. In the Dynamic Kinetic Energy Subgrid Scale Model the subgrid turbulent kinetic energy is modeled similarly to what is done in the standard k-epsilon model. The Dynamic Kinetic Energy Subgrid Scale Model is hence a fundamentally more sound description of the subgrid fluid motion. [2]

2.4 Boundary conditions

In solving the discretized and closed versions of the Navier Stokes equations as shown in section 2.3.1 and 2.3.2 boundary conditions are required. The full description of all boundary conditions used in the simulations would substantially lengthen the report whereby only the most problem specific boundary condition formulations are shown. The interested reader is referred either to FLUENT v12.1 manual or any available CFD literature.

2.4.1 A common approach to inlet turbulence

In the best of worlds the boundary is specified far from the region of interest hence having a limited effect on the flow field turbulent properties. In the real world however the geometry is limited as to reduce the computational cost and hence the boundary will affect the global solution within the domain. A manner in which to artificially model turbulence at the inlet has been proposed using data from experiments and the methodology is outlined in equations 2.45 through 2.48.

$$\mathrm{I} = \frac{u'}{\overline{U}} = 0.16\mathrm{Re}^{-1/8} \tag{2.45}$$

$$k = \frac{2}{3}\left(\overline{U}\mathrm{I}\right)^2 \tag{2.46}$$

$$l = 0.07L \tag{2.47}$$

$$\varepsilon = C_\mu^{3/4}\frac{k^{3/2}}{l} \tag{2.48}$$

In this context I denotes the inlet turbulent intensity and l the turbulent length scale. It should however be noted that this methodology is valid at high Reynolds numbers and should furthermore not be regarded as an exact approach but merely as a rule of thumb. [3]

2.4.2 Particle wall film model in FLUENT

In an attempt to improve the internal particle tracking models in FLUENT additional boundary conditions than the sometimes unphysical models escape, trap and reflect have lately been proposed. The latest in the field is the introduction of the FLUENT wall film model. This model mimics the interaction possibilities as a particle or droplet hits a wall. With that being said the model was according to FLUENT core team developers originally proposed for use in fuel sprays impinging the inside of a cylinder and should hence be used with great caution when applied elsewhere. The general recommendation is always to construct a user specified particle boundary condition that mimics the physics in the problem at hand.

The FLUENT internal wall film model can be sub divided into four parts. In the first step the interaction between the particle and the wall boundary is mimicked. In the second part the particle stuck on the wall is tracked governed by film variables being calculated in the third step. The forth step describes the coupling with the gaseous phase surrounding the wall film. In the mathematical description of these steps a few fundamental assumptions have been made. The wall film is assumed to be thinner than 500 microns resulting in a linear velocity

profile in the film. Furthermore and most importantly the temperature in the film is assumed to vary slowly resulting in a non stiff evaporation sequence. This is by far the most limiting assumption in the wall film model. In addition the temperature of the film is assumed never to exceed the boiling temperature. The in depth mathematical formulation of the particle wall film model is to be found in the FLUENT v12.1 manual and the specifics is beyond the scope of this report. The reader should however note the fundamental assumption of a non stiff evaporation sequence essentially meaning that the externally used time step in FLUENT is sufficiently small to correctly resolve the mass transfer. [2]

2.5 Stiff ODE Suite mathematical formulation

The system of equations solvable in the Stiff ODE Suite framework is presented on a general form in equation 2.49.

$$\dot{y} = f(y,t), \quad y(t_0) = y_0, \quad y \in \vec{R}^N \tag{2.49}$$

However not restricted to autonomous problem formulations the Stiff ODE Suite gains additional numerical precision by assuming time independency resulting in the code itself being optimized according to equation 2.50. Note the differences in the formulation of the derivatives. As most conceivable real physical systems are autonomous this limitation results in numerical gains with no obvious drawbacks.

$$\dot{y} = f(y), \quad y(t_0) = y_0, \quad y \in \vec{R}^N \tag{2.50}$$

In order to solve the system of equations in equation 2.50 a general algorithm is needed. The Stiff ODE Solver is based on a variable order Backwards Differential Formulation (BDF) similar to the algorithm used in ODE15s in MATLAB. Equation 2.51 outlines the BDF formulation.

$$\sum_{i=0}^{K_1} \alpha_{n,i} y_{n-i} + h_n \sum_{i=0}^{K_2} \beta_{n,i} \dot{y}_{n-i} = 0 \tag{2.51}$$

$$h_n = t_n - t_{n-1} \tag{2.52}$$

For stiff systems of equations, which essentially is why Stiff ODE Suite was constructed to begin with, $K_1 = q$ and $K_2 = 0$. q denotes the order of the BDF formulation and varies between 1 and 5 depending on system characteristics. In each time step the non linear system of equations outlined in equation 2.53 and 2.54 are constructed and solved.

$$G(y_n) = y_n - h_n \beta_{n,0} f(t_n, y_n) - a_n \tag{2.53}$$

$$a_n = \sum_{i>0} \left(\alpha_{n,i} y_{n-i} + h_n \beta_{n,i} \dot{y}_{n-i} \right) \tag{2.54}$$

In solving this system of equations the Stiff ODE Suite uses a Newton iteration scheme outlined in equation 2.55.

$$M\left(y_{n(m+1)} - y_{n(m)}\right) + G\left(y_{n(m)}\right) = 0 \tag{2.55}$$

$$M = \mathrm{I} - \gamma J \tag{2.56}$$

$$J = \frac{\delta f}{\delta y} \tag{2.57}$$

$$\gamma = h_n \beta_{n,0} \tag{2.58}$$

The algorithm itself, originally proposed by Bill Gear, is advanced and regarded as one of the most versatile solution strategies to stiff systems of ordinary differential equations available. The solution methodology is to initially decompose the integration time [0,T] into non

equidistant time steps shown in equation 2.52. On each time step equation 2.51 inserted into 2.53 essentially forms an independently solvable predictor equation shown in 2.59.

$$y_n^p = \sum_{i=1}^{k} \gamma_i y_{n-i} \tag{2.59}$$

Using the approximate solution y_n^p at time t_n as an initial guess in iterating equation 2.55 results in the real intermediate solution y_n at time t_n. With both solutions available a local truncated error can be constructed and compared to a predefined user specified value. In the case of too large an error the algorithm will as a first measure if possible increase the order of the BDF formulation. Secondly it will decrease the internal time step and the solution procedure is repeated. The interested reader is referred to the thorough work of Stabrowski for a full description of the BDF based algorithm.

Summing up the algorithm the Stiff ODE Suite supplies a versatile solution strategy via automatically adapted time steps, an economical solution methodology via problem specific BDF order and a most importantly reliable performance comparable with or better than ODE15s. It should be noted that considerable effort has been put into validating the functionality of the standalone solver as introduced by the Stiff ODE Suite. No discrepancies in the algorithm or code have yet been found. [4]

2.6 Coupling the Stiff ODE Suite to FLUENT

In the coupling of the Stiff ODE Suite to FLUENT some fundamental assumptions need to be made. Solving the system of ODEs outlined in figure 2.1, as any system of ODEs, requires a dimension over which to integrate. In the case of reaction rates defined in units of [kmol/m^{3}s] the obvious choice of characteristic dimension is time. Since differential equations commonly encountered in real life are autonomous the start time can always be specified as $\tau_{start} = 0$ whereas the relative final time τ_{final} needs to be accurately specified. In essence this means that the parameter of importance is $\Delta\tau$ after which a solution to the system of ODEs is required. The existence of $\Delta\tau$ however raises the question of what said parameter actually implies.

In a transient simulation the answer is straightforward as the solution progresses with a user specified or machine calculated time step. However in a segregated steady state simulation the concept of time step is undefined. This poses a dilemma as problem formulations where transients are unimportant are most efficiently solved using a steady state formulation. This is illustrated in the simulation of fast equilibrium reactions where a quasi steady state approach is developed and used. Theoretically the pressure based coupled solver could be used in steady state simulations. The coupled solver actually uses an internal time step calculated using, amongst other parameters, the cell Courant to progress towards a final steady state solution. At steady state the time dependent solution satisfies $\phi^{n+1} = \phi^n$ where n is the time step at hand. Further details are given in section 5.1.

With the time step or the lack thereof determined the actual theory behind the FLUENT – Stiff ODE Suite coupling can be outlined. Theoretically a control volume in the computational domain can be regarded as a small semi-continuous batch reactor governed by equation 2.60.

$$\int_V \frac{\partial \rho\phi}{\partial t} dV + \oint \rho\phi \mathbf{v} \cdot d\vec{A} = \oint \Gamma_\Phi \nabla\phi \cdot d\vec{A} + \int_V S_\phi dV \qquad (2.60)$$

Additional information on the terms in the equation outlined above is to be found in section 2.2. In this case the term of interest is the source term S_Φ as it describes generation and decay of a species Φ due to chemical reactions or any other physical event.

The proposed method behind the Stiff ODE Suite is both versatile and stable and uses the large benefits of linearizing the source term on the time step at hand. In figure 2.1 an arbitrary system reaching equilibrium on a very small time scale is illustrated.

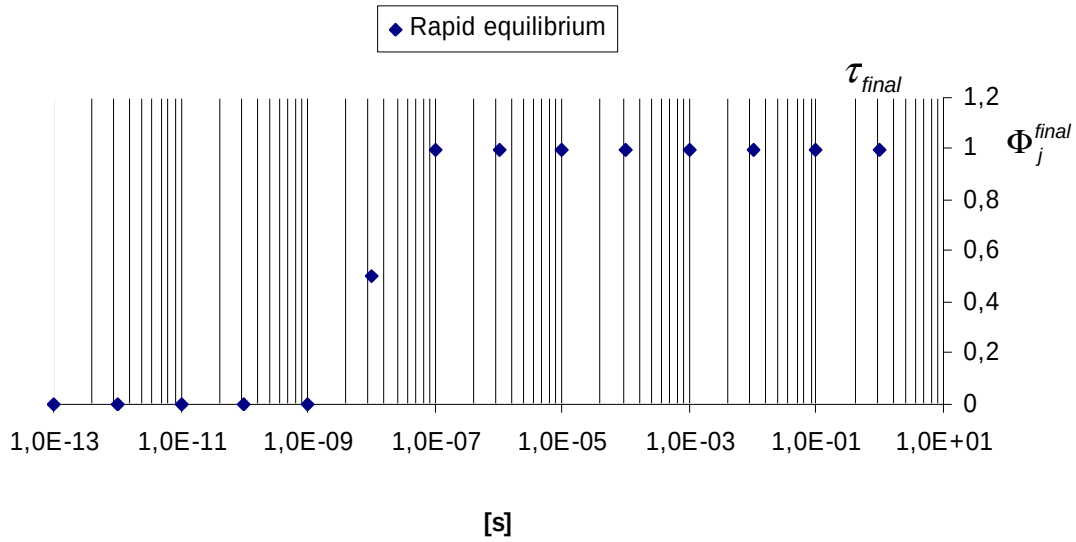


Figure 2.1 Characteristic behavior of a rapidly equilibrated system

As seen in figure 2.1 the equilibrium is reached on a scale of nano seconds. Without using the Stiff ODE Suite this means that FLUENT needs to progress with a time step of equivalent size in order to resolve the extremely rapid change occurring on this time scale. In addition experience shows that even at a time step of this magnitude the solution will tend to be unstable. Using the Stiff ODE Suite makes it possible to resolve the source term S_Φ and the simulation can hence progress with a time step determined by the flow instead of the fast reactions in the system. Characteristically the flow time step is in the vicinity of 1ms rather than 1ns giving a hint of the computational benefits of using the Stiff ODE Suite.

3. Evaluation of the Stiff ODE Suite

The Stiff ODE Suite on its own is a powerful tool in solving very stiff systems of ordinary differential equations. This does, however, not validate the use of the Stiff ODE Suite in engineering applications. In order to test the functionality and versatility of the Stiff ODE Suite – FLUENT coupling two vastly different cases were studied. The first case involved resolving chemical reaction and mixing between a strong acid and a buffering base in a pipe segment. The extreme reaction rates observed in liquid systems involving fast equilibrium reactions form a very stiff system of ODEs previously unsolvable using FLUENT. In the second case fast evaporation of a water droplet jet was studied. With water droplet sizes in the range of micro meters entering heated dry air extreme evaporation rates are expected vouching for a stiff system previously unsolved in a transient simulation using FLUENT.

3.1 Fast equilibrium reactions

Fast equilibrium reactions are a common occurrence in everyday chemical engineering being found in everything from combustion engineering and catalysis to simple liquid phase reactions such as water auto proteolysis. The last example is an interesting problem as the equilibrium between oxonium ions and hydroxide ions is a seemingly instantaneous reaction with theoretical reaction rate constants reaching beyond conceivable numbers. With this in mind it is academically appealing to be able to resolve something as simple as pH when neutralizing a strong acid using a buffering base.

3.1.1 Simulation setup

In the quasi-steady state simulation of neutralizing a strong acid using a buffering base hydrochloric acid is injected through a nozzle into a buffering sodium carbonate solution flowing in a pipe segment outlined in figure 3.1. The hydrochloric acid is injected as a 2M water solution at 0.15 kg/s whereas the sodium carbonate, also a 2M water solution, is injected at the far higher rate of 15.7 kg/s through the respective inlets. The mass flow rate of the sodium carbonate has been adjusted as to ascertain a Reynolds number of roughly 100 000 in the main bulk. The reactions occurring within the pipe segment are assumed to be isothermal setting the simulation temperature to 25^{o}C. The assumption of isothermal reactions is valid since the reaction rates, as indicated below in section 3.1.4, are already set to extreme values removing the necessity of further increasing the reaction rates with temperature.

Figure 3.1 Pipe segment used in neutralization of hydrochloric acid using a sodium carbonate solution. In the figure A denotes the acid inlet. B denotes the sodium carbonate solution inlet. C is the geometry outlet.

The three dimensional computational domain measures 1.15m in length and 0.7m in width with a pipe diameter of 0.2m. The geometry and mesh were constructed using the powerful tools available in ANSA computer software. The mesh in all contains 1.8M cells, which is a sizeable value taking the geometry into account, with a considerable refinement close to the pipe centre axis. This refinement enables the simulation to resolve large species gradients resulting from large changes in the solution pH. Effects on the simulation resulting from large intra cell gradients are discussed in detail in section 5.1.

In the simulation a quasi steady simulation was preferred over a steady state simulation. This means that a converged flow field was obtained by first running a steady state simulation after which a few 0.001s time steps were taken in transient mode. For illustrative purposes a full transient simulation was also performed using a 0.01s time step which is considerably longer than the time scale of any reactions occurring within the system.

3.1.2 Model description

In the case of fast equilibrium reactions in a quasi-steady simulation a truthful description of the mixing within the system is of importance to the final results. As outlined in section 2.3.1 the Realizable k-epsilon model uses a formulation involving the turbulent kinetic energy to describe convective mixing even though the turbulence model by definition does not resolve the transient turbulent eddies. The simplicity and robustness of this general turbulence model combined with the fact that the overall simulation aim is to prove the applicability of the Stiff ODE Suite – FLUENT coupling makes the Realizable k-epsilon model with standard wall functions the turbulence model of choice. The effect of mixing on the final result is discussed in detail in section 5.1.

3.1.3 Boundary conditions

Using the k-epsilon model, unlike the LES model, enables a robust method of specifying inlet turbulent properties using the analogies outlined in section 2.4.1. At the sodium carbonate inlet a turbulent intensity of 3.7% and a turbulent length scale of 0.014m are specified whereas the corresponding characteristics at the acid inlet are 4% and 0.0035m respectively. The turbulent properties of the inlet are of great importance since the very region of interest,

ie where the strong acid meets the buffering base, is situated some 10 cm downstream from the inlet. In order to somewhat reduce the influence of the boundary conditions at the acid inlet the injector, referring to figure 3.1, has been hollowed out leaving a short distance from the actual inlet to the region where the hydrochloric acid meets the sodium carbonate.

At the outlet a pressure outflow is specified setting a constant pressure over the entire surface of the outlet. The boundary condition specified at the pipe wall is no slip as imposed by using standard wall functions.

3.1.4 Reaction kinetics

The reaction kinetics in the hydrochloric acid – sodium carbonate system is of key importance since the very aim is to resolve the fast equilibrium in said reactions. The series of reactions are described as outlined in table 3.1 with the corresponding reaction rates shown in table 3.2.

$$HCl \xrightarrow{r_1} H_3O^+ + Cl^-$$
$$2H_2O \xleftrightarrow{r_2, r_3} H_3O^+ + OH^-$$
$$CO_3^{2-} + H_3O^+ \xleftrightarrow{r_4, r_5} HCO_3^- + H_2O$$
$$HCO_3^- + H_3O^+ \xleftrightarrow{r_6, r_7} H_2CO_3 + H_2O$$

Table 3.1 Reaction stoichiometry

$$r_1 = k_1[HCl] = 10^{20}[HCl]$$
$$r_2 = k_2[H_2O]^2 = 10^6$$
$$r_3 = k_3[H_3O^+][OH^-] = 10^{20}[H_3O^+][OH^-]$$
$$r_4 = k_4[CO_3^{2-}][H_3O^+] = 10^{20}[CO_3^{2-}][H_3O^+]$$
$$r_5 = k_5[HCO_3^-][H_2O] = 4.69 \cdot 10^9[HCO_3^-]$$
$$r_6 = k_6[HCO_3^-][H_3O^+] = 10^{20}[HCO_3^-][H_3O^+]$$
$$r_7 = k_7[H_2CO_3][H_2O] = 2 \cdot 10^{16}[H_2CO_3]$$

Table 3.2 Reaction rates

The reaction constants have deliberately been set extremely large as to construct a very stiff system of ODEs presenting a considerable challenge to any reaction kinetics solver. It should however be noted that the actual equilibrium between the species is preserved at real values. Appendix A contains a full derivation of the reaction rates proposed in table 3.2. To model diffusion transport in the reacting solution a characteristic two component diffusion coefficient of 10^{-9} m^2/s was set to approximate diffusion of dilute species in a water solution.

3.2 Humidifying dry air using a water droplet jet

Spray evaporation is an intriguing phenomenon frequently used in modern day industrial applications. Common applications include fuel injection into pistons and burners as well as the simple need of humid air in air conditioning apparatus. In all of these cases the evaporation sequence of the injected droplets is a key issue since the gas phase fuel is the reacting phase in the pistons and burners and in the case of humid air the water vapour is the final goal.

3.2.1 Simulation setup

In the transient simulation of humidification of dry heated air using a water droplet jet a pipe segment with a water droplet injector was used. The geometry is shown in figure 3.2. In this setup dry air heated to 300^{o}C is injected in the highlighted section at a velocity of 10 m/s. Water droplets at 50^{o}C are injected as a cone shaped spray at a velocity of 100 m/s from the injection nozzle. The relatively high droplet temperature was chosen as to avoid the issue of droplets being cooled below the freezing point of water – an effect arising from rapid fluid acceleration discussed in detail in section 5.2. Mass flow rates in the system were chosen to obtain moist air at 150^{o}C once all the water droplets had been evaporated and the system fully mixed.

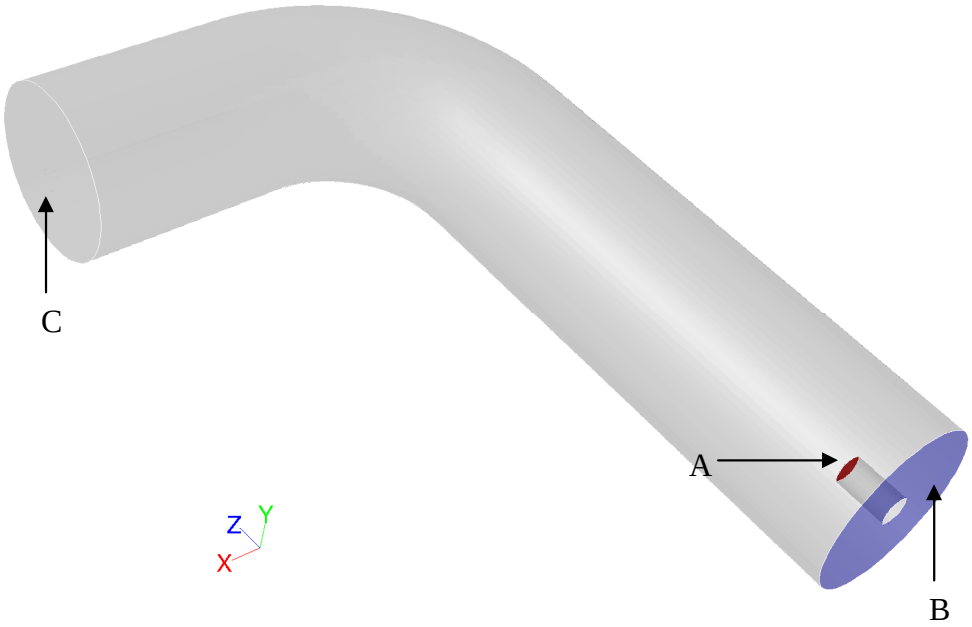


Figure 1.2 Pipe bend used in evaporation simulations. A denotes the droplet injector nozzle. B is the heated air inlet whereas C shows the pipe segment outlet.

The three dimensional computational domain measures roughly 1.8m in length and 1m in width with the pipe diameter spanning 0.4m. In order to minimize the chance of spray particles residing in the same cell a very fine mesh was constructed consisting mostly of extruded tetrahedral elements with a considerable refinement along the pipe centre axis. A characteristic cross section is shown in figure 3.3. The mesh grid is stepwise coarsened towards the pipe wall resulting in a mesh containing roughly 2 million cells.

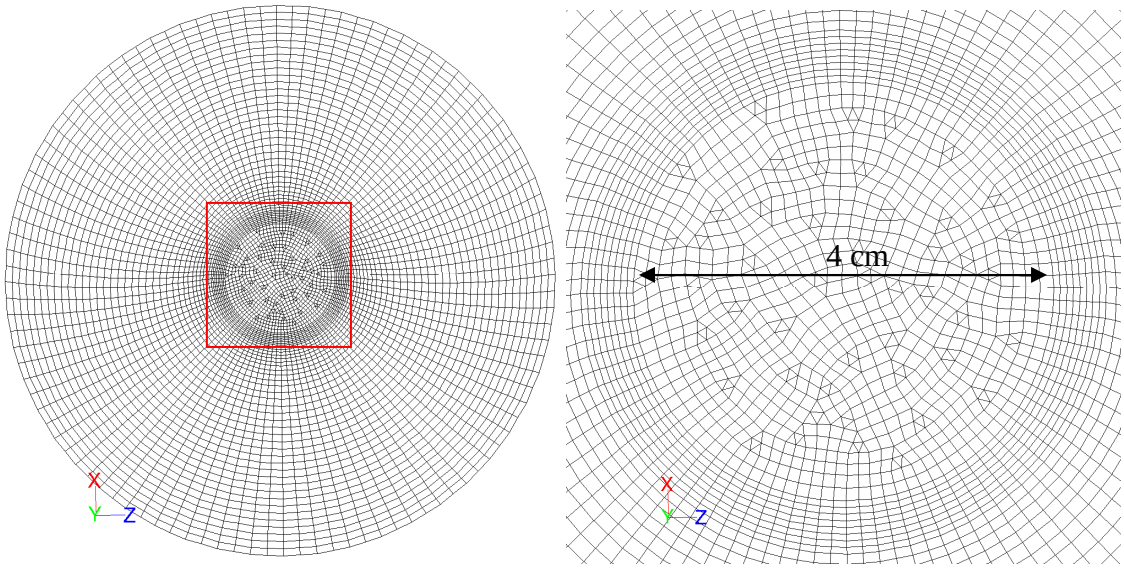


Figure 3.3 Characteristic pipe cross section mesh structure

The particles are injected at a cone angle of 20°. To study the influence of different particle distributions on the overall performance of the air humidification two different Rosin-Rammler particle distributions were simulated whilst keeping all other parameters constant. Figure 3.4 depicts the particle distribution in the different cases.

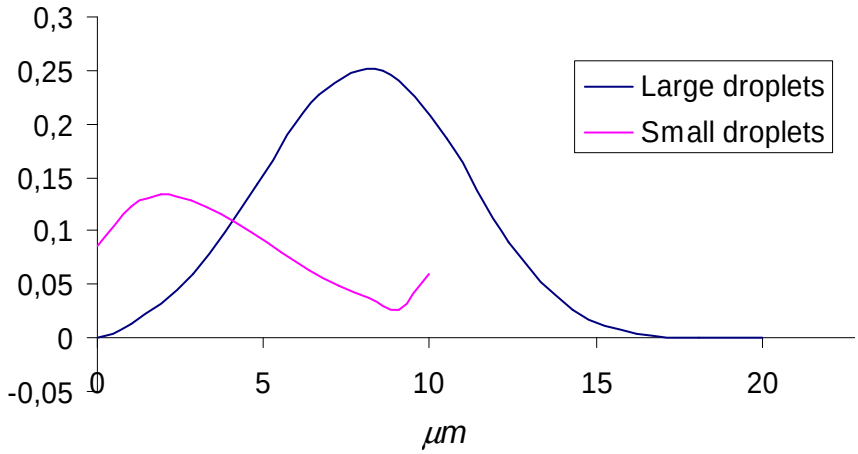


Figure 3.4 Injected droplet size distributions

The transient simulation was performed using an FLUENT external time step of 1ms. This is a very large time step considering what would have to be used had the Stiff ODE Suite not been available. Section 2.6 contains the theory behind the immense computational benefits using the Stiff ODE Suite in transient simulations.

3.2.2 Model description

In modelling the spray a two way coupled Eulerian-Lagrangian approach, described in section 2.1, was adopted. To enable tracking of a large amount of particles individual droplets of identical size are lumped together to form a cloud which from here on will be referred to as “a particle”. As described in section 2.1 parcel tracking imposes certain simulation specific restrictions which have to be taken into account when setting up the simulation. In addition the following reasonable assumptions are made. The droplets are assumed to be small and

dispersed enough to neglect droplet break up and collision but large enough for Brownian motion to be negligible. Furthermore the particles are assumed to have zero rotational velocity setting the Magnus Force to zero. With water droplets being the dispersed phase and air being the continuous phase the ratio $\frac{\rho_f}{\rho_p}$ is very small hence the added mass-, history- and pressure force can be neglected. The significant forces acting on the particles will hence be drag and buoyancy. Section 2.1 outlines the relevant forces.

The phenomenon of fast evaporation is by definition less dependent on subgrid mixing compared to the governing physics of fast equilibrium reactions. In the case of a fast reaction sequence species will react in a wave front which, in order to realistically describe the system, has to be resolved. This in turn gives rise to the need of a very dense computational mesh resolving even the smallest scales of mixing (Kolmogorov scale). With a particle cloud by definition covering the entire computational cell the assumption of a cloud average seeing a local cell average is likely masked by fundamental assumption of the existence of the cloud itself. However in order to resolve large scale transient mixing realistically a compressible LES turbulence model was used to solve the transport equations in the fluid. The theory behind the LES formulation is outlined in section 2.3.2. In LES the large scale turbulent eddies are resolved leaving the small scale turbulent eddies to be modelled. In essence this means that large eddies resolved in a LES model will influence the movement of particles due mainly to drag. The small scale fluctuations important for small scale particle movement however remain unmodelled. Without any further description of this small scale movement particles would seemingly behave predictable in a large scale structured manner. To avoid this somewhat unphysical behaviour a DRW model is introduced as outlined in section 2.1.1. To put the theory into context the DRW model introduces a random component to the local fluid velocity artificially modelling the influence of small scale turbulence.

3.2.3 Boundary conditions

In the pipe segment shown in figure 3.2 there is an apparent risk of unevaporated particles impinging the wall. The particle-wall interaction is an interesting topic calling for a master thesis on its own to be truthfully predicted. As the aim of the simulation was to resolve fast evaporation in the fluid bulk the near wall behaviour was not of primary concern and was hence modelled using FLUENTs internal wall-film model. This model is briefly discussed in section 2.4.2 and the interested reader is referred to section 5.2 for a discussion on the behaviour of said model.

Using the LES turbulence model affects the manner in which boundary conditions are applicable. Ideally the inflow is situated at a position far upstream to the position of interest as to allow the velocity fluctuations to develop into fully turbulent flow. In reality however and especially when the turbulent characteristics of the flow is not the main scope of the simulation the inflow is situated closer to the inlet. There is still the possibility of using a velocity inlet boundary condition and algebraically specify random movement at the inlet but this setting has been shown to be numerically unstable compared to the mass flow inlet boundary condition. Instead the inlet was set as a mass flow inlet with a specified mass flow leaving no room to specify inlet turbulent properties. In order to stabilize the solution and allow for turbulent flow to develop close to the injection nozzle a time span of 0.7s was simulated whilst not injecting the water droplet jet. This is an equivalent of five times the mean residence time of a fluid element in the pipe segment leaving more than enough time for

the solution to stabilize and hence create a fully turbulent flow downstream of the injection nozzle.

At the outflow a constant pressure outlet is specified and any backflow occurring through outflow turbulence is assumed to consist of dry heated air. At the walls in the pipe bend a no slip boundary condition is specified. In order to fully resolve the near wall turbulent generation in LES a much finer mesh grid would have to have been used. However, as the turbulent properties are not the main scope of the simulation computational speed is gained by this slight simplification. As per solving the energy equation the walls are assumed to be adiabatic.

3.2.4 Evaporation Kinetics

The physics behind an evaporating particle cloud residing in a computational cell is somewhat complicated. As evaporation of water is an energy demanding endothermic process the system is largely governed by the availability of heat in the computational cell. As section 5.2 reveals the evaporation process is mass transfer limited at large droplet sizes. Overall the evaporation is a parallel process consisting of transport of heat from the cell bulk fluid to the particle followed by evaporation of water by diffusion and convection at the particle surface. The evaporated mass hence results in a reduction of heat in the particle as well a reduction of the particle diameter. In the odd event of recondensation the heat is added to the particle and the particle diameter is inherently increased. In the evaporation sequence the formed water vapour forms a film surrounding the particles which is then transported by forced convection into the cell bulk. Heat is hence withdrawn from the cell bulk to heat the water vapour to the resulting cell temperature. This is mathematically described as a series of six coupled autonomous ordinary differential equations shown below.

$$\dot{T}_{cell} = \frac{-hn_{part}A_{part}(T_{cell} - T_{part}) - n_{part}\dot{m}_{part}C_{p,H_2O(g)}(T_{cell} - T_{part})}{(m_{air,cell} + m_{H_2O,cell})C_{p,cell}} \tag{3.4}$$

$$\dot{T}_{part} = \frac{hA_{part}(T_{cell} - T_{part}) + \dot{m}_{part}\Delta H_{vap}}{m_{part}C_{p,part}} \tag{3.5}$$

$$\dot{d}_{part} = -\frac{2\dot{m}_{part}}{\pi\rho_{part}d_{part}^2} \tag{3.6}$$

$$\dot{m}_{H_2O,cell} = n_{part}\dot{m}_{part} \tag{3.7}$$

$$\dot{C}_i = \frac{600}{d_{part}}(1.5\cdot 10^6 e^{\frac{-43000}{RT_{part}}} - 0.27695RT_{cell}^{0.5}C_i) \tag{3.8}$$

$$\dot{m}_{part} = kA_{part}M(C_i - C_{cell}) \tag{3.9}$$

Equation 3.8 is a reformulation of evaporation and condensation of water in terms of gas phase water vapour concentration for which a full derivation is available in Appendix B. Furthermore equation 3.6 is a rewritten expression for the change in particle diameter coupled to the actual loss in particle mass. The interested reader is yet again referred to Appendix B.

To close the stated system of equations knowledge of the external heat and mass transfer coefficients h and k is needed. These are obtained using semi empirical correlations evaluated at local cell properties using equation 3.10 – 3.13.

$$Nu = 2 + 0.6\,\mathrm{Re}^{1/2}\,\mathrm{Pr}^{1/3} \tag{3.10}$$

$$h = \frac{Nu \cdot \lambda}{d_{part}} \tag{3.11}$$

$$Sh = 2 + 0.6\,\mathrm{Re}^{1/2}\,Sc^{1/3} \tag{3.12}$$

$$k = \frac{Sh \cdot D_{H_2O,air}}{d_{part}} \tag{3.13}$$

Additional material data are appended in Appendix B. [5]

PART II

4. Results

4.1 Fast equilibrium reactions

In the neutralization of a strong hydrochloric acid solution by use of a sodium carbonate solution a key parameter in validating the strength of the Stiff ODE Suite – FLUENT coupling is the reliability of pH prediction. pH is determined by the water auto proteolysis which by definition is an extremely fast reaction rendering this equilibrium reaction fundamentally difficult to describe using any chemical reaction solver. In figure 4.1 the pH of the quasi steady state simulation at the pipe entrance as calculated using the Stiff ODE Suite is shown.

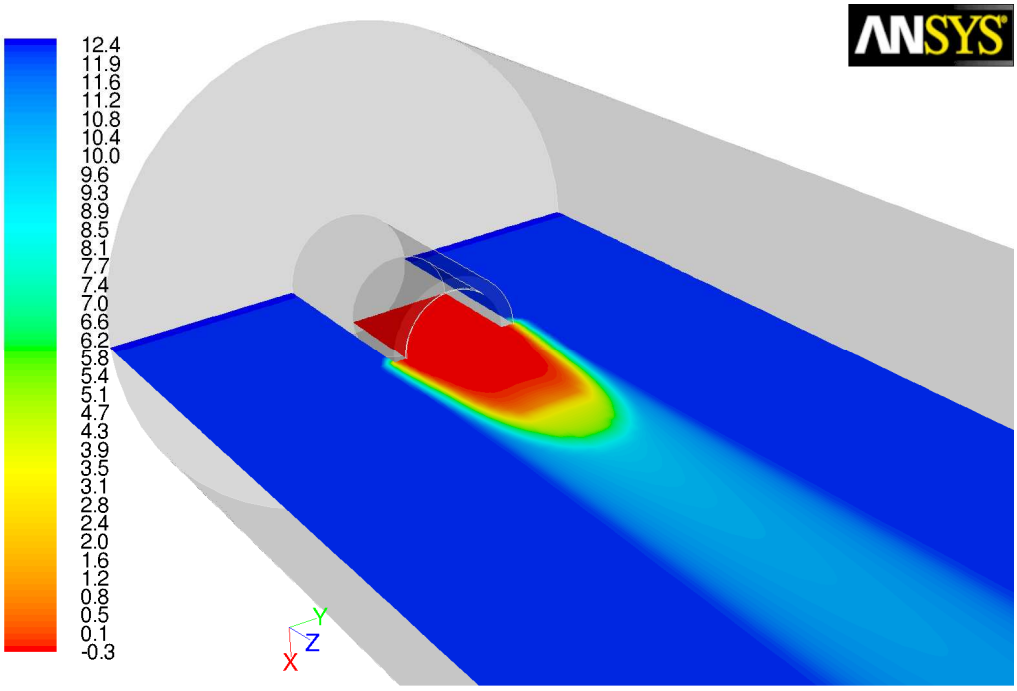


Figure 4.1 Contours of pH at pipe inlet

Figure 4.1 depicts a distinct neutralization wave front between the strong acid inlet and the main high pH sodium carbonate solution. In the wake of the acid inlet the pH is lowered considerably giving rise to the equilibrium formation of HCO_3^-. Due to the buffering capabilities of the sodium carbonate solution the HCO_3^- ion is favoured in regions of low pH. The concentration of HCO_3^- is shown in figure 4.2. Note that all eight modelled species are present at prevailing system equilibrium. However not important to the main scope of the simulation the contour plots of said species are excluded from the report.

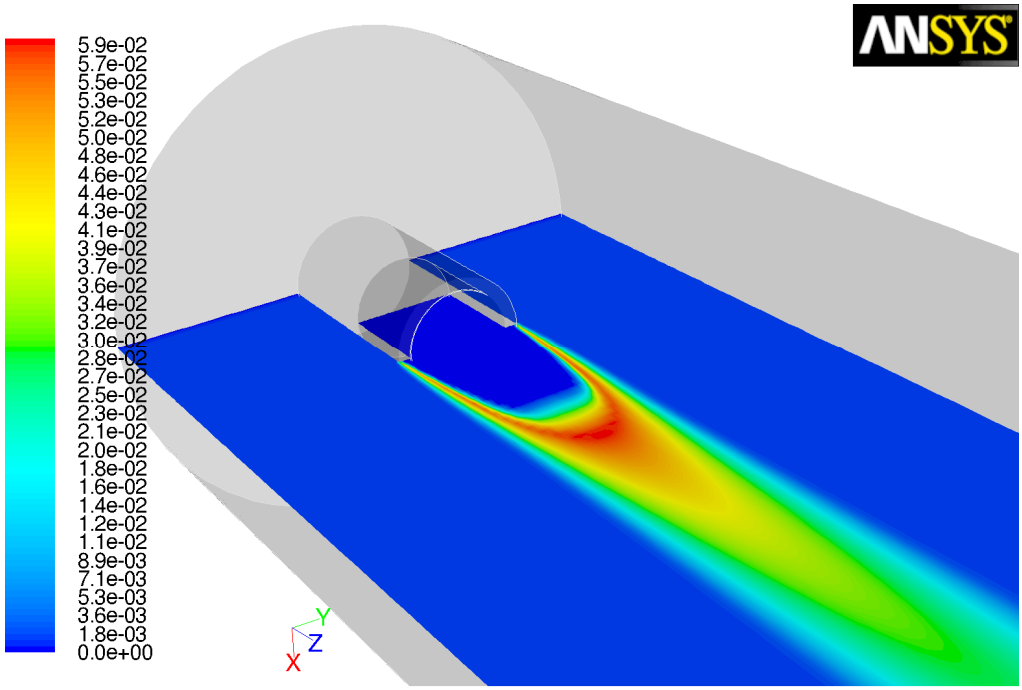


Figure 4.2 Mass fraction of HCO3- at pipe inlet

An interesting measure on the quality of the solution is resulting simulation value of the water auto proteolysis equilibrium constant which at equilibrium is $K_w=10^{-14}$. In figure 4.3 a 2D cross section of the only region of erroneously predicted values of K_w is shown. It however goes to show that almost the entire region outlined in figure 4.3 is falsely predicted by less than a factor 100 with only a few computational cells close to the solid nozzle head being falsely predicted with simulation values of K_w far from the true value.

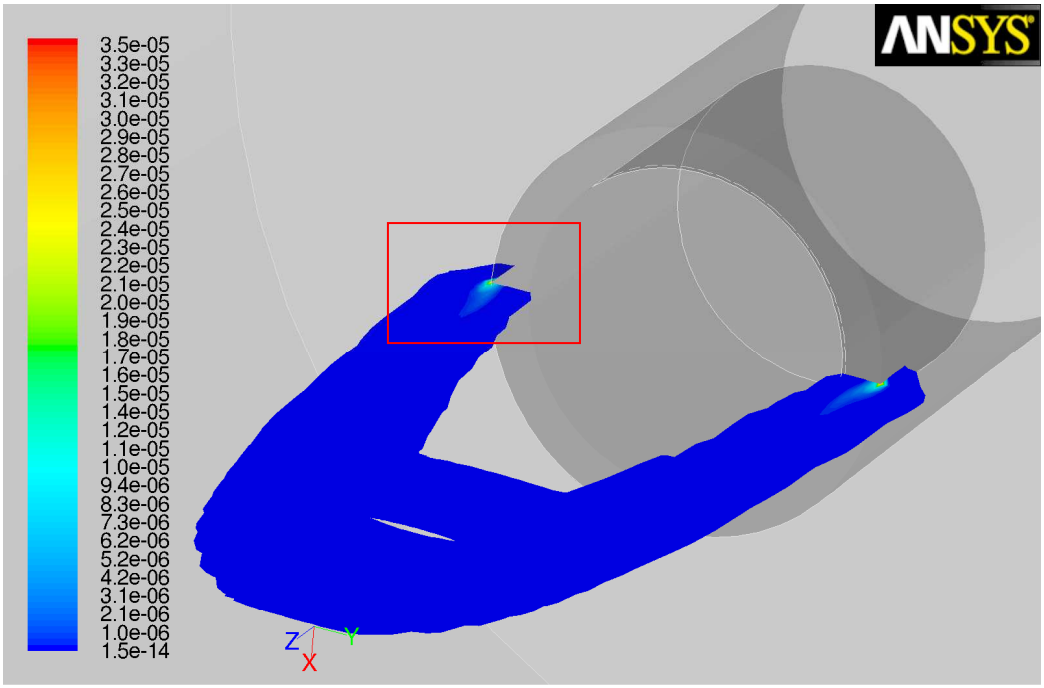


Figure 4.3 Contours of K_w in the vicinity of the acid inlet nozzle. 1ms time step

Figure 4.4 and figure 4.5 depict the pH wave front and the reaction rate of H_3O^+ ions respectively in the region shown in figure 4.3. These figures are most relevant for a further

discussion on the reason behind the erroneously predicted values of K_w. A detailed discussion is to be found in section 5.1. In order to give a complete picture of the domain in the vicinity of the acid inlet nozzle the computational grid is included in the figures.

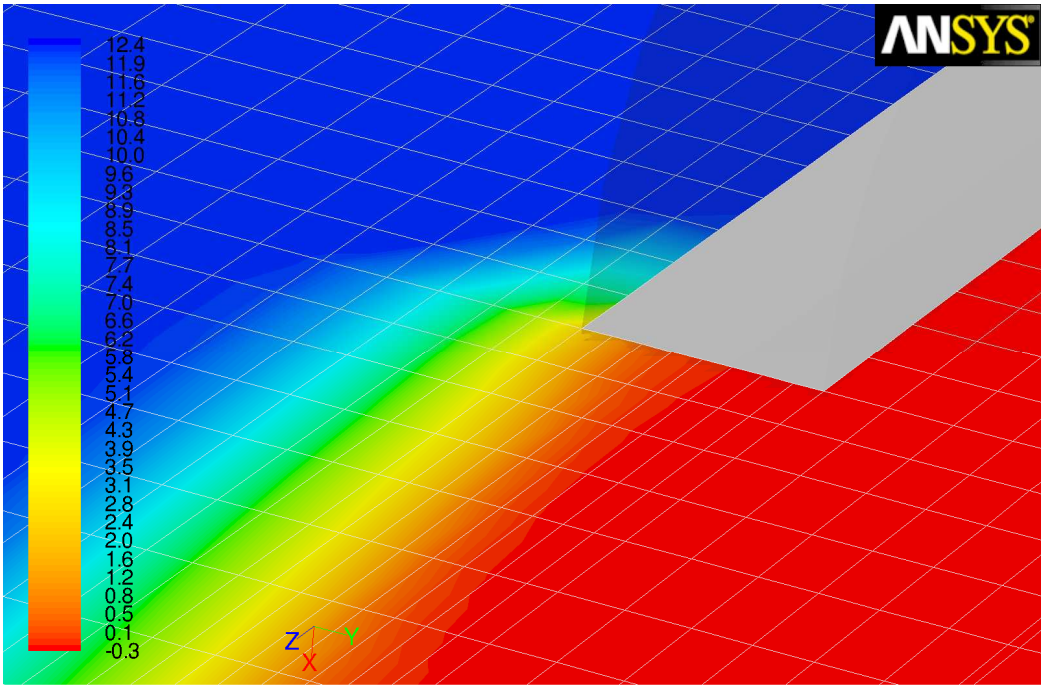


Figure 4.4 Contours of pH. Close up at reaction wave front

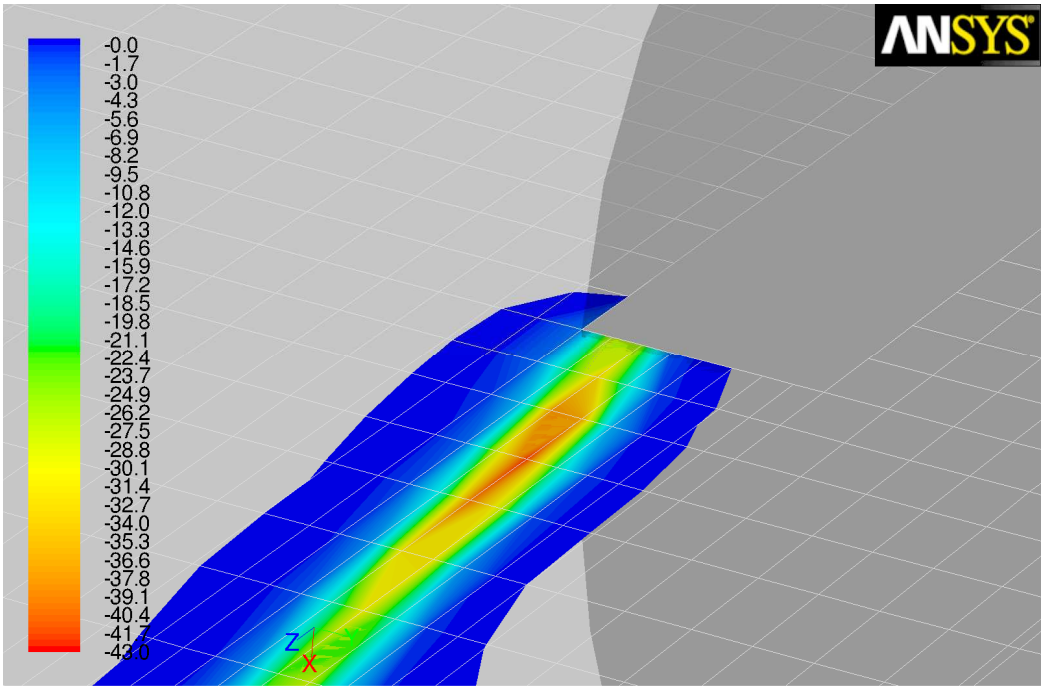


Figure 4.5 H_3O^+ reaction rate [kmol/m^3]

4.2 Humidifying dry air using a water droplet jet

In the transient simulations of air humidification using a water droplet jet two size distributions were simulated under otherwise identical physical conditions. In figure 4.6 and 4.7 the vastly different sequences of events are shown. Figure 4.6 depicts the larger droplets in the spray surviving the harsh conditions in the pipe bend finally striking the pipe wall creating a wall film and a mist of reflected droplets. The particles, in figure 4.6 coloured by temperature, form a cold core in the jet surrounded by sparsely spaced particles with temperatures reaching and surpassing the temperature of the surrounding fluid. Figure 4.6 b and c reveal the turbulent properties of the flow affecting the particle movement in the pipe bend forming a chaotic tip of the jet.

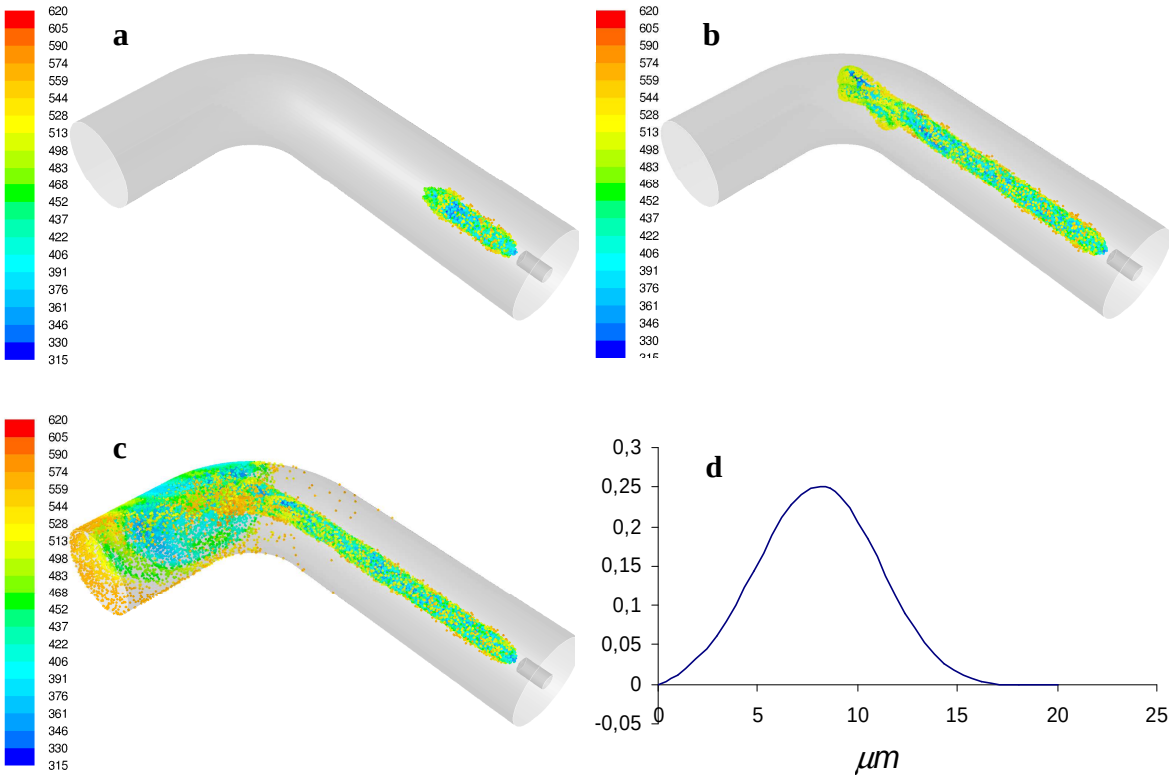


Figure 4.6 Particles coloured by temperature. Solution at times a) 0.01s b) 0.04s c) 0.07s d) Size distribution of injected droplets

In the simulation of the smaller droplets however the sequence of events are quite different. It is again empathized that both simulations are performed using identical mass flow rates of water. Figure 4.7 depicts the water jet propagating through the heated air keeping the temporarily surviving droplets at a low temperature. Furthermore no droplets impinge with the wall but are instead instantaneously evaporated as they move into a region of higher temperature. It should be noted that the characteristics of figure 4.7 b) and c) represent a simulation steady state prevailing even as the simulation progresses past 0.15s and onwards. Again the tip of the jet is governed by seemingly chaotic movement as imposed by the LES resolved turbulent eddies close to the pipe bend.

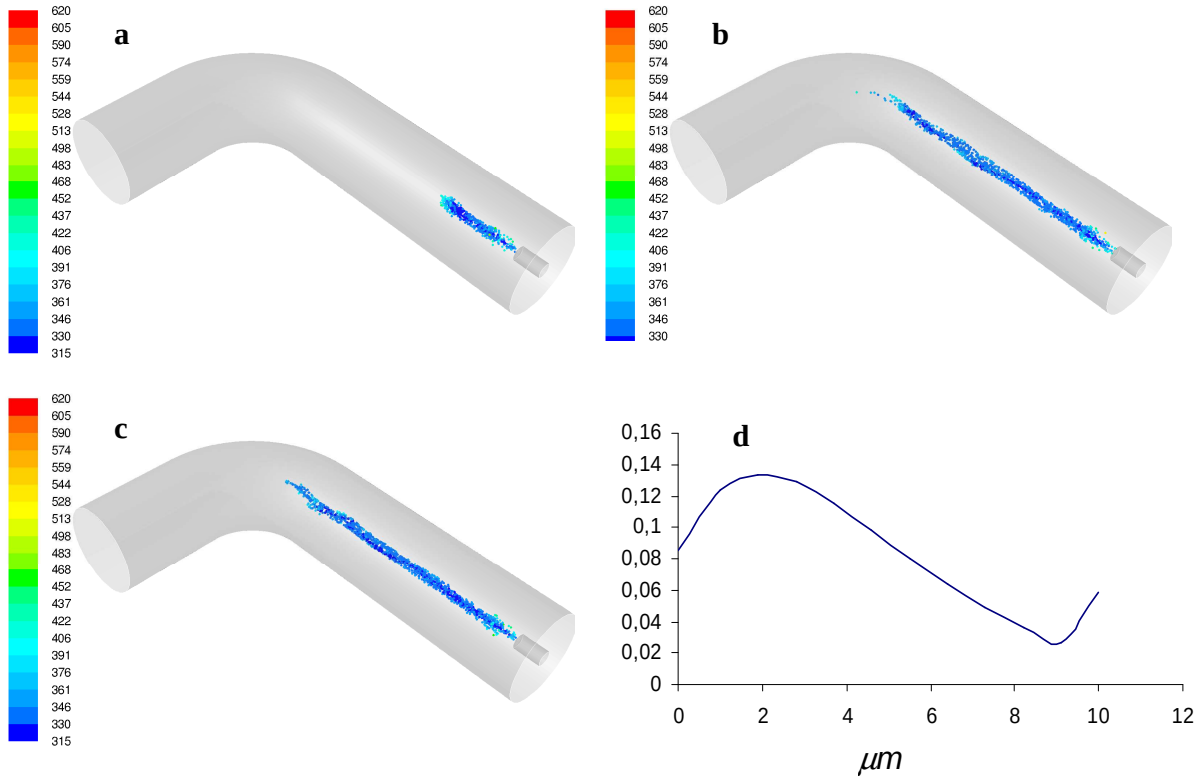


Figure 4.7 Particles coloured by temperature. Solution at times a) 0.01s b) 0.04s c) 0.07s d) Size distribution of injected droplets

In figure 4.8 the initial flow field used in both water jet simulations is shown. As proven in section 2.3.2 the LES framework fundamentally enables the resolution of transient turbulent eddies. Figure 4.8 depicts the formation of such eddies at several positions in the pipe segment with a notable concentration in the wake caused by the separation at the 90^{o} bend. The droplet injection nozzle itself causes a wake to be formed with induced chaotic behaviour in the mean flow towards the pipe bend.

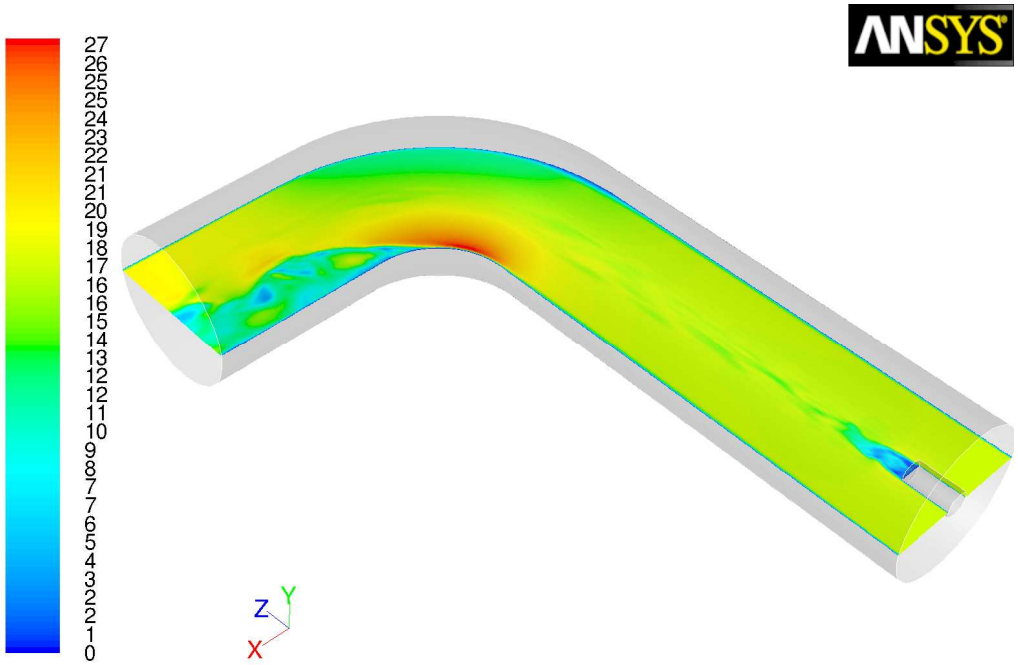


Figure 4.8 Converged LES velocity field [m/s]. Solution at time 0.7s without the injection of droplets

Figure 4.9 depicts the molar concentration of water vapour in the fluid bulk as produced by the evaporation in the smaller droplets shown in figure 4.7 d). The water vapour concentration is notably high down stream close to the injection nozzle proving that a substantial part of the total system evaporation takes place immediately following the injection of the droplets. Injection of the larger droplets, depicted in figure 4.6 d), results in a similar characteristic water vapour concentration plot.

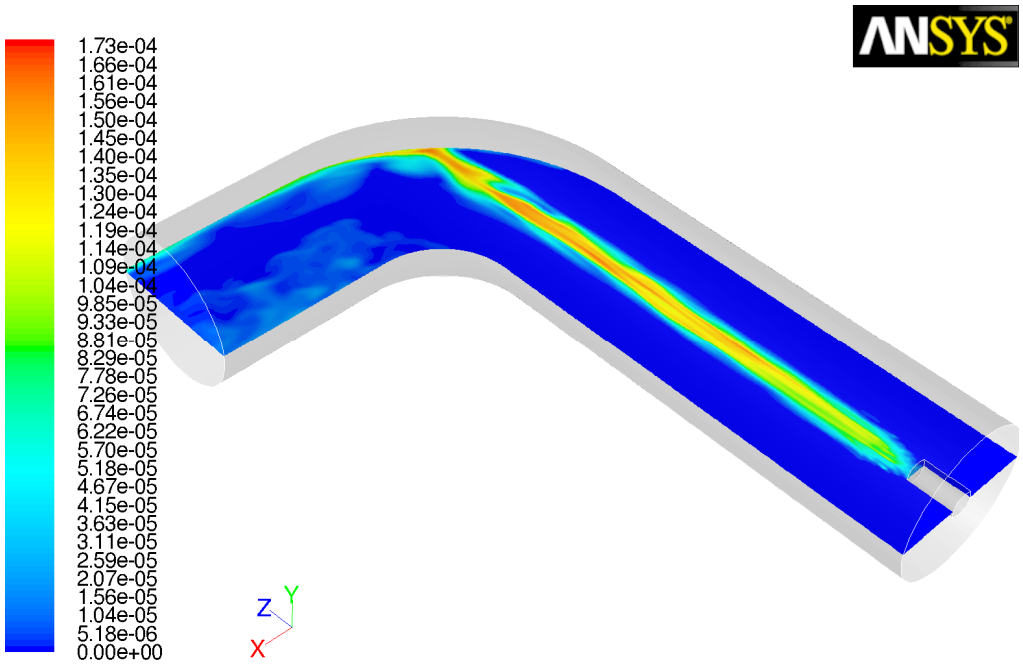


Figure 4.9 Molar concentration of water vapour [kmol/m^3]. Solution at time 0.15s with droplet size 1-10 micro meter

Figure 4.10 through 4.12 illustrate the fluid bulk temperature, the wall film height being formed after the pipe bend and the water vapour source term respectively. These are supplied

as a basis of further discussion to be found in section 5.2. Notable in figure 4.10 however is the jet core temperature at certain positions being ~25K lower than the temperature of the initially injected droplets. Most of the jet core however pertains the initial droplet temperature of 325K.

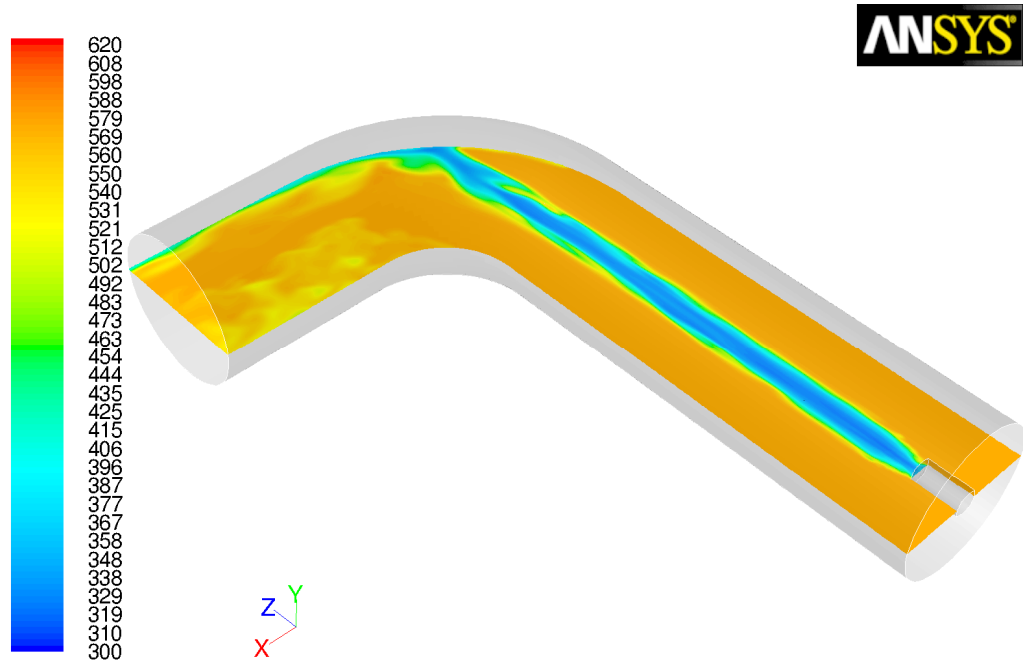


Figure 4.10 Temperature profile in fluid bulk [K]. Solution at time 0.15s with droplet size 1-10 micro meter

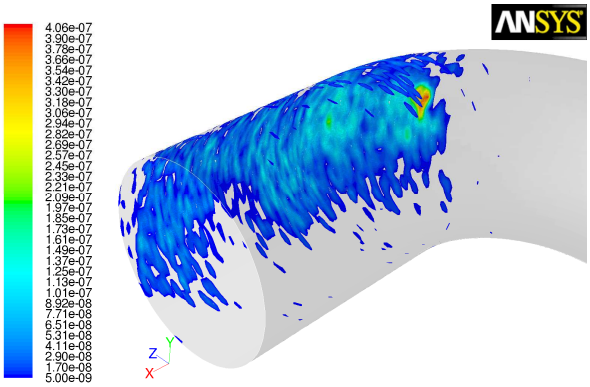


Figure 4.11 Artificial wall film height [m]

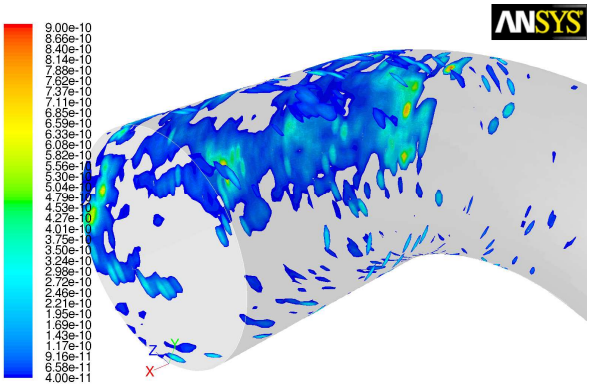


Figure 4.12 Water vapour source term [kg/m^{3}s]

5. Discussion

5.1 Fast equilibrium reactions

In the simulation of neutralization of hydrochloric acid using a sodium carbonate solution a region of inferior predictability was discovered and shown in figure 4.3. The phenomenon itself was initially assumed to be an artefact stemming from unconverged simulations but regardless of any measure to increase convergence the issue remained. Instead the intriguing phenomenon is due to the ratio between system mixing and reaction rate. Examining figure 4.4 in detail reveals that the solution pH varies between -0.3 and 12.1 over a mere 8 computational cells. A characteristic length scale of 8 cells in the region of interest is 0.01m over which the concentration of H_3O^+ ions essentially changes from 2M to 0M. With a local effective diffusivity of size ~10^{-5}-10^{-4} m^2/s taking both diffusion transport and convective transport, through mean flow and modelled turbulent eddies, into account the inter cell transport is characterized by equation 5.1.

$$n_i = A_i D_{eff} \frac{\delta[H_3O^+]}{\delta x_i} \approx \frac{V_{cell}^{2/3}}{V_{cell}} \cdot D_{eff} \frac{[H_3O^+]_{cell9} - [H_3O^+]_{cell1}}{x_{cell9} - x_{cell1}} = [1.68,16.8] \frac{kmol}{m^3 s} \quad (5.1)$$

In forming equation 5.1 it has been assumed that the computational cells in the region can be regarded as perfect cubes with a characteristic region specific volume of $1.7 \cdot 10^{-9} m^3$. The calculated size range is not in any way an exact measure but a hint on the inter cell transport of H_3O^+ ions in the region. Comparing the inter cell transport to the reaction rate in the region depicted in figure 4.5 spanning [0.05,43] kmol/m^3s the Dahmköhler number $\mathrm{Da} = \frac{\text{typical time of mixing}}{\text{typical time for chemical reactions}} \propto \frac{\text{rate of reaction}}{\text{rate of mixing}}$ turns out to be in the vicinity of unity. It can be shown that this is the case in the entire region outlined in figure 4.4. With a Dahmköhler number close to unity, more precisely put not in the region of Da<<1 or Da>>1, the system mixing becomes a key parameter in determining the chemical reaction rates and the assumption of a cell being treated as a semi continuous batch reactor according to section 3.1.5 no longer holds.

Numerically the issue of falsely predicted values of K_w stems from the manner in which FLUENT iterates. In regions of high values of pH at the edge of the neutralization wave front the concentration of H_3O^+ will be very small with values essentially in the region of ~10^{-12} kmol/m^3 hence giving extremely small reaction rates of said ion. The transport of the H_3O^+ ions to the cell will however be much larger as predicted by equation 5.1. In the iteration sequence FLUENT will hence compare two quantities, reaction rate and inter cell transport, of vastly different sizes. The result is that a “converged” solution is obtained with values of $[H_3O^+]$ at ~10^{-9} kmol/m^3 because the reaction source term exemplified below is approximately zero compared to the inter cell transport of ~1.7 kmol/m^3s.

$$S_{H_3O^+} = -\frac{(\text{iteration concentration - equilibrium concentration})}{\Delta t} = -\frac{10^{-9} - 10^{-12}}{0.001} = -10^{-6} \frac{kmol}{m^3 s} \quad (5.2)$$

Hence K_w will be erroneously computed by a factor of 1000.

$$[OH^-][H_3O^+] = 10^{-2} \cdot 10^{-9} = 10^{-11} << K_w = 10^{-14}$$

The issue can be resolved in a few different ways. First and foremost if the actual resolution of the computationally intensive wave front is not the scope of the simulation but instead the solution pH at a position further downstream the erroneous zone can be completely disregarded. The global mass in the system is maintained and as soon as a fluid element moves away from the wave front the equilibrium will instantaneously be corrected. Another remedy in which the wave front itself can be predicted is to use the measure of [OH^-] in regions of high pH and [H_3O^+] in regions of low pH. Using these measured values to calculate pH will give a less numerically distorted result. This method does however not resolve the region where pH is close to 7 where both concentrations will be equally distorted. Instead in order to fully describe the system the grid cell size needs refinement down towards DNS level. This is fundamentally the only truthful way to describe the system mixing including all relevant stages ie inertial-convective-, viscous-convective- and viscous-diffusive mixing. With the time span available such a simulation is however hopelessly unpractical.

In a quasi steady state simulation the time step could be decreased towards the characteristic time of the actual reactions occurring in the system. This would, in accordance with equation 5.2, increase the source term hence reducing the numerical issues at hand. When using a time step of 1ns the region of falsely predicted values of K_w is largely reduced. Figure 5.1 depicts the erroneously predicted region. The characteristically edged form of the contour in figure 5.1 proves that a further grid refinement at the acid inlet would increase resolution of K_w. The fundamental error of falsely predicted system mixing however remains. Using these extremely short time steps made possible in a quasi steady state simulation is furthermore not feasible in the transient framework since the real world simulation time would be uneconomical at best.

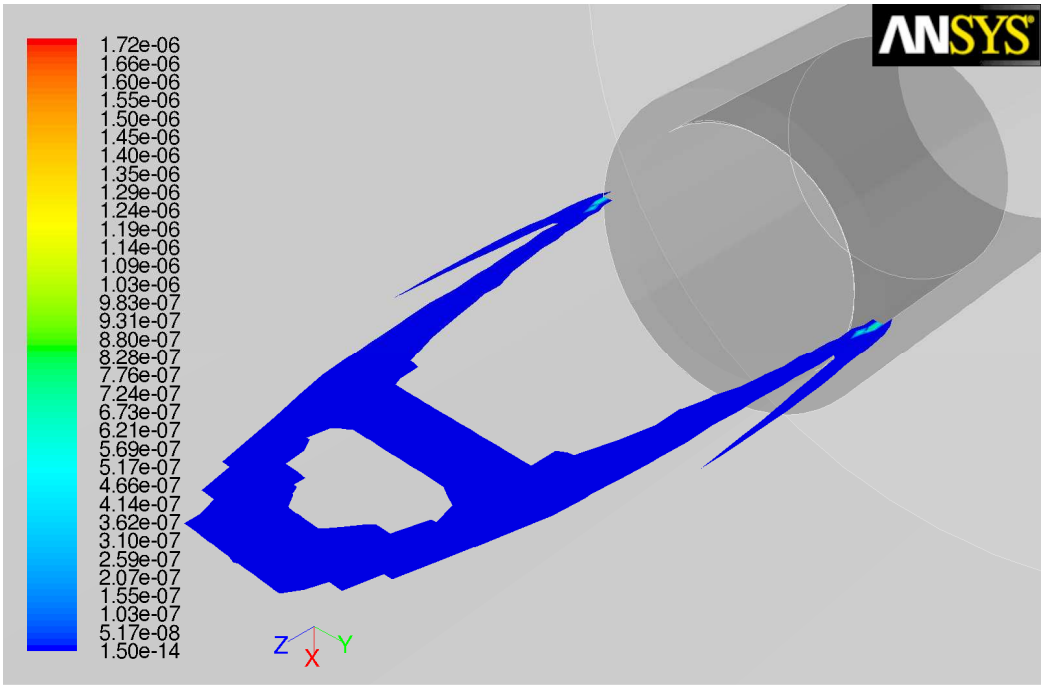


Figure 2 Contours of K_w in the vicinity of the acid inlet nozzle. 1ns time step

Using the pressure based coupled steady state solver in FLUENT however opens up a new realm of simulation possibilities. In this approach the solution is internally progressed towards a steady state using a time step back calculated from amongst other things the local Courant number. Fundamentally the pressure based coupled solver approach avoids problems with main flow changes due to the sudden change in chemical reactions possibly encountered when switching from a steady state to a transient solver. Equations 5.3 through 5.9 reveal the procedure of calculating the local time step. [2]

$$\Delta t = \frac{2CFL \cdot V}{\sum_{face} A_{face} \lambda_{face}^{\max}} \tag{5.3}$$

$$\lambda_{face}^{\max} = \max(u, u' + c', u' - c') \tag{5.4}$$

$$u = \vec{v} \cdot \vec{n} \tag{5.5}$$

$$u' = u(1 - \alpha) \tag{5.6}$$

$$c' = \sqrt{\alpha^2 u^2 + U_r^2} \tag{5.7}$$

$$\alpha = \frac{1 - \beta U_r^2}{2} \tag{5.8}$$

$$\beta = \frac{\partial \rho}{\delta P}\bigg|_T + \frac{\frac{\delta \rho}{\delta T}\Big|_P}{\rho C_p} \tag{5.9}$$

5.2 Humidifying dry heated air using a water droplet jet

In simulating the humidification of dry heated air using a water droplet jet several interesting phenomena were discovered and studied.

When injecting droplets of the size distribution outlined in figure 4.6 d) particle parcels containing few but large droplets $(>19\mu m)$ to a great extent survive long enough in the high temperature domain in order to eventually impinge the pipe wall. At the same time almost no parcels with diameter $10\mu m$ survive past the first section following the injection nozzle. The reason fundamentally lies in the evaporation rate as described by equation 3.9. Examining said equation noting that $k \propto d_{part}^{-1}$ and $A_{part} \propto d_{part}^2$ reveals that $\dot{m}_{part} \propto d_{part}$. Forming the life expectancy of a droplet in the system $\tau_{life} = \frac{m_{part}}{\dot{m}_{part}}$ and noting that $m_{part} \propto d_{part}^3$ reveals $\tau_{life} \propto d_{part}^2$. This hence explains the over representation of large droplets in the pipe bend. A droplet of size $20\mu m$ simply has a 100 times longer life expectancy than a droplet of size $2\mu m$. Each particle parcel with droplets in the size range $>19\mu m$ however due to limited size distribution mass contains less than 10 droplets each resulting in a slight visual misrepresentation in figure 4.6. In said figure it looks like a large portion of the droplets impinge the wall whereas in reality only a small portion of the total droplet mass escapes evaporation. This is reflected in the extremely thin wall film having been formed in figure 4.11.

Continuing the discussion regarding mass transfer from the droplets to the surrounding fluid the formulation in equation 3.6 is empathized. Said equation states that the change in droplet diameter is proportional to the transferred mass but inversely proportional to the square of the droplet diameter. The formulation is hence by definition stiff since the rate of change in droplet diameter rapidly progresses towards infinity as the droplet diameter is increased. This gives rise to the question why the droplet diameter is the parameter of choice and not the droplet mass which would result in a mathematically much less complex system since there would be no tendency towards division by zero. To be able to answer this question one needs to realize the manner in which a particle exists in FLUENT. From the time the particle parcel is born at the injection nozzle till it dies due to evaporation it contains a constant number of droplets. The parameter instead being changed due to evaporation is the particle diameter which will decrease to 1nm at which time the particle is considered fully evaporated. Just before being fully evaporated the droplet mass will hence be in the vicinity of 10^{-25}kg at which time the evaporation rate is of the same numerical order. By subtraction the removed mass might be close enough to the droplet mass that the resulting mass is either positive or negative ~10^{-60}kg. This machine precision induced numerical issue gives rise to an oscillatory Stiff ODE Suite solution behaviour which will significantly add to the computational cost. Instead the much larger parameter droplet diameter is modelled giving rise to a stiff system of ODEs easily solvable using the Stiff ODE Suite.

In simulating the droplet jets a sub cooling effect of the jet core was noticed. This effect stems from the acceleration of the fluid caused by the rapid transfer of momentum from the injected particles hence an interesting effect arising from the two way coupled multiphase system. Examining the energy equation 2.17 the term $P\frac{\delta U_j}{\delta x_j}$ acts either as a sink or a source depending on if the fluid is accelerated or decelerated. It should be noted that the simulation was performed using a compressible formulation of the fluid density hence $\frac{\delta U_j}{\delta x_j} \neq 0$ is possible. Running a short trial simulation with a incompressible fluid density formulation removed the temperature effect hence proving the hypothesis. The sub cooling of the jet core was the main reason why the droplet inlet temperature had to be chosen arbitrary high ie 325K. By doing so the simulation ran no risk of unphysically predicting droplet temperatures below the freezing point of water in which case the enthalpy of melting would have to have been added.

In the evaporation simulations the near wall region is somewhat falsely predicted the reason for this being the need of a very fine computational grid close to the wall in order to resolve large gradients. The goal of the simulation is however to resolve bulk evaporation which renders the near wall fluid region less important. The savings in terms of computational cost is however greatly reduced by the reduction of computational cells. In all the mesh contains roughly 2 million cells which, considering the geometric dimensions of the domain, is a very fine mesh leaving only the near wall domain unresolved.

The near wall region using the particle distribution outlined in figure 4.6 d) is however of interest to the particle description. As covered in the theory section the FLUENT internal wall film model was used however augmented with a slight modification. In order to avoid the limited applicability of the FLUENT internal evaporation models the Stiff ODE Suite was used to describe evaporation from the pipe walls. By doing so droplets can be evaporated at an arbitrary rapid rate at temperatures far above the boiling temperature of water. Figure 4.12

depicts the source terms of water vapour close to the wall as generated by evaporation. The wall film model description of the formation and movement of the wall film and the probability of droplet reflection and wall separation is however kept unaugmented. Using the Stiff ODE Suite evaporation formulation in the entire interior domain of the pipe segment however imposes an error on the description of evaporation from the wall. Since the Stiff ODE Suite evaporation formulation, according to equation 3.4 to 3.9, assumes droplet evaporation the formation of a wall film is neglected and instead wall particles are treated as any other particles in the domain. Film evaporation assuming an adiabatic surface is inherently slower than the evaporation of dispersed droplets causing the augmented film model to somewhat over estimate the mass transfer from the wall. Since the large droplets impinging the wall are already subjects to slow evaporation the global time frame of the simulation reduces the effect of the overestimated evaporation rate. It should be noted that the internal FLUENT evaporation model from a wall film, with the assumptions outlined in the theory section, is a vastly simplified approach best used with fuel sprays in vertical cylinders as originally intended.

Further work needs to be done implementing a general wall film model into FLUENT. The mathematical description of the formation and movement of a wall film is beyond the scope of this discussion but a general basis for further work is hereby presented. By coupling the Discrete Phase Model with the Volume Of Fluid model it is theoretically possible to form a wall film whilst still tracking particles in the fluid bulk. In this approach the Stiff ODE Suite is well suited for describing evaporation and condensation to/from the VOF modelled wall film.

6. Conclusion

Stiff systems of autonomous ordinary differential equations, typically on the form $y' = ky \quad k >> 1$, are a common occurrence in nature being found in fast equilibrium reactions, catalysis, evaporation and incineration. In an attempt to solve stiff systems of ODEs encountered in CFD applications the software package Stiff ODE Suite was developed, coupled to FLUENT v12.1 and validated.

The Stiff ODE Suite is based on a variable order Backwards Differential Formulation algorithm with adaptive time stepping capabilities. The algorithm introduces a versatile stiff or non-stiff ODE solver with numerical performance comparable to MATLABs ODE15s. The Stiff ODE Suite functions as a standalone module and in coupling it to FLUENT it vastly increases said software´s applicability. The coupling also makes use of the benefits introduced by multi processing – an advantage of great use in modern day multi million mesh grids.

The Stiff ODE Suite was used to resolve fast equilibrium reactions in FLUENT where neutralization of hydrochloric acid using a sodium carbonate water solution in a pipe segment was simulated. In using the Stiff ODE Suite the extremely fast water auto proteolysis was predicted with acceptable accuracy hence enabling the resolution of the mixed solution pH. Furthermore the Stiff ODE Suite enabled the simulation of rapid humidification of dry heated air using a water droplet jet in a pipe segment. In the simulations two way coupled momentum, mass and heat transfer was successfully modelled realistically predicting the evaporation time of water droplets.

Fundamentally the introduction of the Stiff ODE Suite enables the CFD engineer to use time steps much larger than the time scale of the stiff system of ODEs needing to be solved. It also extends the applicability of FLUENT since the user is no longer limited to the descriptive capabilities of the graphical user interface. Concluding imperfect or faulty FLUENT internal models can be augmented and changed to better suite the problem formulation at hand.

7. Bibliography

1. C. Crowe, M. Sommerfeld, Y. Tsuji, *Multiphase Flows with droplets and particles*, CRC Press, 1998

2. ANSYS FLUENT v12.1 Documentation, Software Guide - bundled with ANSYS FLUENT v.12.1, Accessed on 2010-02-01

3. B. Andersson. R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, B. Wachem, *Computational Fluid Dynamics for Chemical Engineers*, 2009

4. Marek M. Stabrowski, *An efficient algorithm for solving stiff ordinary differential equations*, Simulation practice and theory, volume 5, Issue 4, pp. 333-344, 1997

5. J. Welty, C. Wicks, R. Wilson, G. Rorrer, *Fundamentals of Momentum, Heat, and Mass Transfer, 4th edition, John Wiley & Sons, 2001*

6. R. Bolz, G. Tuve, Handbook of Tables for Applied Engineering Science, 2nd edition, CRC Press, 1976

7. Carbon Dioxide and Carbonic Acid-Base Equilibria, http://www.chem.usu.edu/~sbialkow/Classes/3650/Carbonate/Carbonic%20Acid.html, Accessed on 2010-03-20

APPENDIX A – Determining reaction kinetics

With reference to the reaction rates shown in table 3.1 and 3.2 the generation and decay of each component can be expressed as shown in equations A.1 through A.8.

$$\frac{d[HCl]}{dt} = -r_1 \tag{A.1}$$

$$\frac{d[H_3O^+]}{dt} = r_1 + r_2 - r_3 - r_4 + r_5 - r_6 + r_7 \tag{A.2}$$

$$\frac{d[Cl^-]}{dt} = r_1 \tag{A.3}$$

$$\frac{d[OH^-]}{dt} = r_2 - r_3 \tag{A.4}$$

$$\frac{d[H_2O]}{dt} = 2(-r_2 + r_3) + r_4 - r_5 + r_6 - r_7 \tag{A.5}$$

$$\frac{d[CO_3^{2-}]}{dt} = -r_4 + r_5 \tag{A.6}$$

$$\frac{d[HCO_3^-]}{dt} = r_4 - r_5 - r_6 + r_7 \tag{A.7}$$

$$\frac{d[H_2CO_3]}{dt} = r_6 - r_7 \tag{A.8}$$

In all instances of the derivation of the reaction rates the fastest step in the equilibrium is set to $k = 10^{20}$ yielding extremely fast equilibrium reactions. The slower reaction constant is derived by back calculating from the actual equilibrium which results in expressions for all rate constants according to equation A.9 through A.20. [7]

$$k_1 = 10^{20} \tag{A.9}$$

$$\frac{r_3}{r_2} = 1 = \frac{k_3[H_3O^+][OH^-]}{k_2[H_2O]^2} \tag{A.10}$$

$$[H_3O^+][OH^-] = K_w = 10^{-14} \tag{A.11}$$

$$k_3 = 10^{20} \tag{A.12}$$

$$k_2 = \frac{k_3 K_w}{[H_2O]^2} \tag{A.13}$$

$$\frac{r_4}{r_5} = 1 = \frac{k_4[CO_3^{2-}][H_3O^+]}{k_5[HCO_3^-][H_2O]} = \frac{k_4}{k_5}\frac{K_{A2}}{[H_2O]} \quad \text{(A.13)}$$

$$K_{A2} = 4.69 \cdot 10^{-11} \quad \text{(A.14)}$$

$$k_4 = 10^{20} \quad \text{(A.15)}$$

$$k_5 = \frac{k_4 K_{A2}}{[H_2O]} \quad \text{(A.16)}$$

$$\frac{r_6}{r_7} = 1 = \frac{k_6[HCO_3^-][H_3O^+]}{k_7[H_2CO_3][H_2O]} = \frac{k_6 K_{A1}}{k_7[H_2O]} \quad \text{(A.17)}$$

$$K_{A1} = 2 \cdot 10^{-4} \quad \text{(A.18)}$$

$$k_6 = 10^{20} \quad \text{(A.19)}$$

$$k_7 = \frac{K_{A1} k_6}{[H_2O]} \quad \text{(A.20)}$$

APPENDIX B – Determining evaporation kinetics

Derivation of equation 3.8

Derived from fundamental kinetic gas theory condensation of water in terms of water film pressure can be expressed according to equation B.1 to B.4. [BB BENGT Reference]

$$\frac{\dot{n}_{cond}}{A} = k_a P_{H_2O}\left[\frac{mol}{m^2 s}\right] \tag{B.1}$$

$$k_a \propto T_b^{-1/2} \tag{B.2}$$

$$k_a = 1434\left[\frac{mol}{Pa \cdot m^2 s}\right], T = 373K \tag{B.3}$$

$$k_a = \frac{0.27695}{T_b^{1/2}} \tag{B.4}$$

Also derived from kinetic gas theory evaporation of water can be described according to equation B.5 and B.6 [BB BENGT Reference]

$$\frac{\dot{n}_{evap}}{A} = k_d e^{-\frac{E}{RT_p}}\left[\frac{kmol}{m^2 s}\right] \tag{B.5}$$

$$k_d \propto 1.5 \cdot 10^9 \left[\frac{kmol}{m^2 s}\right] \tag{B.6}$$

$$E = 43000 \frac{J}{mol}$$

Expanding equation B.1 and expressing P_{H_2O} in terms of an ideal gas renders equation B.7.

$$\frac{\dot{n}_{cond}}{A} = k_a C_i R T_b \left[\frac{mol}{m^2 s}\right] \tag{B.7}$$

In studying the growth and decay of a particle water vapour film due to evaporation and condensation equation B.7 is formed. In equation B.8 it has been assumed that the volume of the water vapour film is an equivalent of 1% of the particle volume. The assumption is in essence unimportant since the inverse of the particle diameter, shown in equation B.8, will cause a_{film} to grow rapidly causing adding stiffness to $\frac{dC_i}{dt}$.

$$\frac{dC_i}{dt} = evaporation - condensation = a_{film}\left(1.5\cdot 10^{9} e^{\frac{-43000}{RT_p}} - 0.27695 R T_b^{1/2} C_i\right)\left[\frac{mol}{m^3 s}\right] \quad \text{(B.8)}$$

$$a_{film} = \frac{A_{part}}{V_{film}} = \frac{6}{d\cdot 0.01}\left[\frac{1}{m}\right]$$

In terms of kmol/m^{3}s equation B.7 is rewritten and the final formulation is produced in equation B.9.

$$\frac{dC_i}{dt} = evaporation - condensation = \frac{600}{d}\left(1.5\cdot 10^{6} e^{\frac{-43000}{RT_p}} - 0.27695 R T_b^{1/2} C_i\right)\left[\frac{kmol}{m^3 s}\right] \quad \text{(B.9)}$$

Derivation of equation 3.6

The change in diameter of a particle with constant density due to the change in mass is expressed in equations B.10-B.11.

$$\frac{dm}{dt} = \frac{d}{dt}\left(\frac{\pi d^3}{6}\right) = \frac{\pi d^2}{2}\frac{dd}{dt} \quad \text{(B.10)}$$

$$\frac{dd}{dt} = \frac{2\frac{dm}{dt}}{\pi d^2} \quad \text{(B.11)}$$

Additional property data

The following section contains additional material data entered to augment FLUENT default values.

The diffusion coefficient of the water vapour air system was set according to Bolz and Tuve. In equation B.12 the temperature is specified in Kelvin. [6]

$$D_{H_2O,air} = -2.775\cdot 10^{6} + 4.479\cdot 10^{-8} T + 1.656\cdot 10^{-10} T^2 \quad \text{(B.12)}$$