MODELLING OPEN CHANNEL FLOW

By Stephen Hopton, B.Sc. (Hons), M.Sc.

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

September 2010

Abstract

The study of open channel flow and dam breaking is not a new topic in computational fluid dynamics. However it has only recently started to gain significant attention from researchers using meshless methods, i.e. numerical modelling techniques which do not rely on the use of a mesh to discretise the domain. The research presented here is an attempt to use the meshless method known as smoothed particle hydrodynamics (SPH) to simulate the flow of water down a channel.

Hydra, a pre-existing SPH code designed originally for astrophysical simulations, was converted to simulate water flow and then applied to the problem of dam bursting and flow over a weir. The conversion of the code to its new purpose was verified by simple code tests and then extensive validation was performed via the modelling of multiple dambreaks. The validation process can be split into three broad categories: 1) Comparison against the published data gained from other numerical methods both meshless and traditional. 2) Comparison against physical experiments performed by the author. 3) Comparison against independent experimental data found in the literature. Hydra in its newly converted form was satisfactorily applied to the majority of the tests presented to it and the same level of accuracy was achieved as with any of the other codes tested. A limit to the SPH method for performing this type of simulation was proposed based on particle number, smoothing length and initial conditions. A formula for the calculation of the number of ghost particles required to prevent spurious boundary pressures was also proposed. An analysis of various kernels used by different SPH researchers was presented and it was discovered that a relatively simple cubic spline kernel proved sufficient and that increasing complexity did not provide an increase in solution accuracy. The flow of water over a weir was presented next and results compared to published data which utilised a leading mesh based fluid simulation package. Results gained from Hydra simulations showed good downstream water level prediction but overestimation of upstream levels. A steady

state solution was achieved within a similar timeframe compared to the grid based method.

It was concluded that use of the SPH method and the Hydra code in particular can provide solutions to problems involving water flow down a channel and accuracy on the dambreak tests was equal to any rival codes/methods tested. However when the complexity of the boundaries involved in the model increased there was some evidence that the CFX simulation package could be used to achieve a more accurate solution than Hydra. Suggestions for continuation of research into Hydra as a water flow modelling code are presented in addition to recommendations for improving the experimental methods used.

Publications

The work presented in this thesis was performed by the author, with advice from supervisors. Where material has been taken from literature explicit references are given.

In addition to this thesis some of the material in chapter 3.2 has been published in:

Tasker, E., Brunino, R., Mitchell, N., Michielsen, D., Hopton, S., Pearce, F., Bryan, G., Theuns, T. (2008). "A test suite for quantitative comparison of hydrodynamics codes in astrophysics". Mon. Not. R. Astron. Soc. Volume 390, Issue 3, pp. 1267-1281.

Material in section 4.4 is currently under preparation for publication in:

Hopton, S., Pearce, F., Morvan, H. and Wright, N. (2010). "Smoothed Particle Hydrodynamics for simulating wave propagation from a dambreak". Journal of Hydraulic Research.

Acknowledgments

The author would like to thank the following people:

- Dr Frazer Pearce for all his help on this project and his seemingly endless knowledge of SPH and computer programming.

- Prof. Nigel Wright for offering me opportunity to study for a PhD and all his help and encouragement during the course of it.

- Dr Herve Morvan for all his help and encouragement over the project and his huge knowledge of all things CFD.

- All the others in both the Civil Engineering department and the Astronomy department who have helped me over the years and made studying at Nottingham such fun.

The author would also like to thank:

- Bethan for all the encouragement to keep going and finish this rather large document.

- My parents for everything they have done for me over the years.

Table of Contents

List of Figures

Chapter 1

Introduction

In today's fast paced world of technological progress Computational Fluid Dynamics (usually referred to as CFD) has become essential to a great many fields of science and engineering. Indeed it has become an essential part of several industries such as automotive, oil and gas, aerospace, manufacturing, ventilation and marine along with many others. Essentially what CFD does is simulate on a computer a problem that involves fluid flow. This allows the engineer to devise a way of solving a problem without the need for practical work which may be difficult, expensive or sometimes even impossible to do. Indeed, several attempts may be needed to provide the best solution and this is much more practical on a computer than in reality. Many companies now have CFD sections within their research and development departments so that ideas can be tried out before manufacture even takes place. In the past this element of the design process would have been reserved for high tech laboratories that could afford the computer power but nowadays many simulations can be performed on a cluster of, or even a single, desktop pc.

Traditional CFD works by dividing up the region of interest (i.e. the area being simulated) into a set of discrete elements or cells. This is known as the grid or the mesh and it is made to fit onto the geometry of the problem at hand. Fluid flow is described mathematically by a set of partial differential equations called the Navier-Stokes equations. These are rewritten for the computational process and used to convey information about the fluid such as temperature or density between cells. The properties of a fluid inside a cell affects the properties of the fluid inside neighbouring cells (though of course they may be the same) and this is used to simulate motion. This means that the resolution of the simulation is, at least partially, limited by the cell size so it is better, or more accurate, to have a large number of small cells rather

than a few big ones. Having more cells does increase the computational power required to run the simulation however so usually a compromise between the two is reached. Another common method is to switch from a rigid grid to an adaptive one in order to reduce costs without losing accuracy. Even with large advanced CFD abilities however, traditional CFD methods can not solve everything with a desired level or economy or precision. Flood Hydraulics and wetting/drying simulations for example have been studied for years within CFD and it could hardly be said that perfect solutions exist for these problems.

The governing equations of fluid dynamics can be considered in two frames of reference: the Eulerian frame which is stationary and where the fluid moves past the observer and the Lagrangian frame which moves with the fluid. Typically grid-based methods are based on an Eulerian frame of reference whilst particle-based methods are based on a Lagrangian frame.

For the Eulerian frame of reference various methods can be used to discretise the governing equations: Finite Difference Method (FDM), the Finite Element Method (FEM) and the Finite Volume Method (FVM). Between them they form the bulk of all CFD simulations carried out since CFD's inception in the middle of the 20th Century (though others certainly exist). There are several differences between the three, and these are explored more in chapter 2, but key differences revolve around how the governing equations are discretised and the kind of grid used. Several codes have tried to combine methods together to take advantage of each ones strengths (Combined Lagrangian and Eulerian Grids). For the purposes of this research however, the important thing to remember is that they all rely on a grid or mesh to define the domain and evaluate the fluid properties.

Understanding the flow of water, whether it be in a river, over a floodplain, along the coast or as a result of a burst dam or levee failure is a challenging problem for engineers and has been an obvious case for CFD analysis for many years. Serious and/or large scale flooding events have become an increasingly concerning problem across many parts of the world for both ordinary people and their governments. In the last half-century alone there have been several

flooding events leading to serious loss of life and massive property damage and these have occurred in both developed and developing countries. This ranges from the North Sea storm surge in 1953 in the UK, the Netherlands, Belgium and Germany to the Boxing Day Tsunami in 2004. In the UK alone recent events have demonstrated the destructive power of flood water and the importance of flood defences and a planned emergency response. This is clearly shown from the partial destruction of the village of Boscastle, Cornwall and the flooding which occurred across wide stretches of the country in 2007.

In the EU, currently over 12% of the population of the United Kingdom live on fluvial flood plains or areas identified as being at risk of coastal flooding. This equates to approximately 7.2 million people. Approximately half the population of the Netherlands lives below mean sea level (~8.25 million people). In Hungary about 25% of the population lives on the floodplain of the River Danube and its tributaries (~2.5 million people) (floodrisk.org). It is very important that we are able to simulate flood inundation events with accuracy because if we know how and where, for example, a river will flow when it has burst its banks or broken through a dam we can improve our flood risk mitigation strategies such as evacuation plans.

This project is designed to study the possibilities of using the Hydra fluid simulation code, which uses the Smoothed Particle Hydrodynamics (SPH) method, to model open channel flow and flood inundation events. SPH is a Lagrangian computational simulation method created for and used mainly in the field of astrophysics where it has enjoyed considerable success since its inception in 1977. The key point to note about SPH is that it is different to many of the more commonly used CFD simulation methods because it is “meshless”. This means that it is not restricted by a grid allowing it to model complex structure more easily than a grid based code. Instead it uses a collection of particles to represent the fluid body. The hypothesis of this research is that when an SPH code can be converted from astrophysical gas based modelling to Earth based water modelling it will enjoy similar successes. This provides the main aim of the research, to convert an astrophysics based SPH code and modify it to simulate water flow on Earth. If reliable Hydra simulations can be achieved (by

comparison with and testing against other proven simulation methods) then this research could show that there is a future for Hydra in water flow modelling and flood defence design. SPH has been suggested as a means of simulation in a wide range of fields including this one with some promising initial results. This work will collect together information on these previous efforts and attempt to expand upon them with regard to flooding and open channel flow. In addition to converting a SPH code for flooding simulations and channel flow, experiments and traditional CFD methods will be investigated for comparison and validation but also as research tools in their own right.

Chapter 2

Theory and Literature Review

2.1 CFD

2.1.1 Introduction

CFD is the process of using numerical methods to provide computationally derived solutions to problems involving fluid flow. CFD has been around since the 1960s and methods have been continually updated and refined since that time. The advent of better, and considerably cheaper, computer hardware in the 1990s saw CFD techniques really take off. The basic principle is to create a simplified model of a real physical problem and then gradually apply more and more complex physics until it represents the real thing as closely as possible (though it will always be an approximate solution). The challenge is to combine numerical accuracy with modelling precision and achieve the best possible solution at an acceptable cost. CFD solutions have some advantages over experimentation in that they are always repeatable, modifiable and safe to the researcher.

CFD typically begins with the Navier-Stokes equations. These are the non-linear partial differential equations, derived from Newton's second law, which provide the fundamental definition for any single phase fluid flow. They do not provide a solution directly for the flow but rather provide a relationship between the rates of change of the variables. The Navier-Stokes equation for compressible flow in vector form for the conservation of momentum accompanied by the corresponding continuity equation can be expressed as:

$$\rho \frac{Dv}{Dt} = -\nabla P + \nabla \bullet T + f \tag{1}$$

$$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho v) = 0$$

Where ρ is density, v is velocity vector, P is pressure, f represents body forces (often just gravity) and T is a 2nd order stress tensor which describes the stresses throughout the fluid. In continuum mechanics the stress tensor is usually comprised of a pressure term and a deviatoric term $\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ which relates it to the distribution of local fluid velocity.

A good review of the complete derivation of the Navier-Stokes equations can be found in Versteeg and Malalasekera (1995).

The typical method of simulating fluid flow is to discretise the domain into cells, which can be regular or irregular, then solve the equations of motion (Euler for inviscid, Navier-Stokes for viscous). This process involves three stages:

Pre-processing

- Domain is discretised
- Geometry is defined
- Volume occupied by fluid is divided
- Physics is defined
- Boundary conditions are defined.
- Steady state or transient simulation defined

Solver

- The equations are iterated over and over by the computer till the simulation end time has been reached

Post-processing

- Results are visualised
- Required analysis is performed

2.1.2 Discretisation

Probably the largest consideration when performing a CFD simulation is how to handle the discretisation process of turning a continuous fluid into a mesh or grid. This is important because by forming a set of discreet, finite elements instead of having a continuous volume allows the problem to be solved by the application of partial differential equations. Typical CFD solutions involve not only a mesh, but a discretisation scheme and a time marching scheme as well. There are several schools of thought on how to best accomplish this although several are notably very similar to each other, three main ones will be discussed briefly here. Much of the following description is a summary of a more detailed description in Versteeg and Malalasekera (1995).

The Finite Difference Method (FDM) uses a fixed rectangular grid and discretises the equations using Taylor Series expansion. This method was used historically due to its ease of programming and accurate results but tends to rely on a fairly regular mesh. It therefore does not handle large deformations or complex problems well. It is rarely used in modern codes though it is still sometimes used today in specialised codes for solving certain problems.

The Finite Volume Method (FVM) discretises the domain into a number of finite volumes and integrates the governing equations over each of these. This method is popular amongst fluid dynamics researchers because integrals are applied separately within each volume. One applies the conservation principle (volume integration) and exploits the Gauss Green theorem to turn a volume problem into a surface one; the rate of change of one property inside a control volume can now be assessed by the computation of the property fluxes at the CV

boundaries. A structured grid is not required when using this method giving it an advantage due to the effort saved.

The Finite Element Method (FEM) divides the domain up into elements usually in the form of basic geometric shapes. The numerical solution is also determined by integration albeit a weighed integration and shape functions are also used to express the value of a property continuously as a combination of the cell nodal values. An attractive feature of FEM is that it is well suited to handling complicated geometries and is generally considered to be very robust however conservation is more difficult to enforce and this technique is not as commonly seen in CFD.

2.1.3 Turbulence

Turbulence is a phenomenon in fluid dynamics where interactions within the fluid cause chaotic and hard to predict changes. Instabilities in the flow cause energy to be dissipated from the system through the generation of eddies in the fluid. A common example to demonstrate this is smoke rising from a cigarette. It is ordered and smooth at first but then becomes unstable. A non-turbulent fluid is known as laminar and is usually characterised by a low Reynolds number (Hughes and Brighton, 1999). There is ongoing research into this problem as it represents one of the biggest challenges in fluid dynamics. When all length scales (i.e. including the eddies) can be resolved by the grid Direct Numerical Simulation (DNS) is possible. This relies on the mesh being fine enough so that it is below the Kolmogorov length scale (Easom, 2000). This would be good in an ideal world but in practice is unachievable for most simulations. The number of grid cells required for this would be beyond currently available computer power. Two commonly used techniques are Reynolds Averaged Navier Stokes (RANS) and Large Eddy Simulation (LES). In the former, equations are produced which approximate the NS equations which introduce new stresses called Reynolds stresses to simulate turbulence. LES is considered somewhere in-between

DNS and RANS in that it filters out the smallest eddies and models them separately on a sub grid. Larger eddies are simulated on the main grid in the normal way. This makes LES more computationally expensive than RANS but nowhere near as costly as DNS.

There are many different codes which have been optimised over the years to study fluid dynamics using the above principles. There are also many other less well known methods, for example the Boundary element method (Cheng and Cheng, 2005) and new research is still continuing on such things as an extended version of FEM (XFEM) (Moes, Dolbow and Belytschko, 1999). Several complete CFD solution packages are currently available commercially using all of the methods described in this section and sometimes combine several of them together (e.g. CFX, Fluent). The key point to all of them is that whether they use a Lagrangian or Eulerian point of view, they discretise their domain in the form of a mesh or grid.

2.1.4 Meshless Methods

Meshless (or meshfree) methods are a class of numerical simulation methods which limit, or do away with entirely, the need for a spatial grid to be formed. Particles themselves form the viewpoint from which the solution is reached. This type of discretisation is a substantial alteration from the traditional way in which CFD is performed. There are several meshless methods currently being used, some more successfully than others. The most commonly come across ones being the Moving particle semi-implicit method (MPS) (Koshizuka and Oka, 1996), the Element-free Galerkin method (EFG) (Belytschko, Lu and Gu, 1994), Radial Basis Functions (RBF) (Stevens, Power and Morvan, 2008) and Smoothed particle hydrodynamics (SPH) (Monaghan, 1992). Meshless methods have perhaps enjoyed more widespread use in astrophysics research (SPH especially) but they have proven valuable in fluid dynamics as well. The major advantages most often cited they have over mesh based methods are mesh distortion insensitivity and natural resolution adaptivity. A more detailed

review of the differences in SPH and grid methods (from an astrophysics point of view) can be viewed in Agertz et al (2009). SPH will be discussed in considerable detail later in this chapter and in Chapter 3.

2.2 SPH

Smoothed Particle Hydrodynamics was conceived in 1977 by Gingold and Monaghan (1977) and independently by Lucy (1977). It was one of the first meshless methods of CFD. That is a hydrodynamical fluid simulation method that does not rely on grids or meshes to function. It is a fully Lagrangian particle simulation method where the particles are tracked as they move over time. Originally designed to solely model astrophysical fluids, i.e. gases, plasmas and stars in space; it has been modified over the years to cope with a wide variety of problems and can still show some advantages over today's state-of-the-art Adaptive Mesh Refinement (AMR) codes. The main modifications that have made SPH so adaptable over the years are the abilities to alter the equation of state (EOS) of the fluid (i.e. turn the gas into liquid or solid) and to be able to include solid boundaries. This is essential for a lot of the modern (non-astrophysics) research which involves processes done in a liquid phase in bounded domains. However it is still also being improved every year for its original purpose in astrophysics' simulations.

Where SPH differs from more traditional CFD methods that use a grid is that the fluid is represented solely by a collection of lagrangian particles which are acted upon by forces. Each particle represents an interpolation point with a smoothing length. The equations which represent the fluid properties (e.g. energy, momentum) become sets of differential equations used at the particle positions. Each particle contributes to the forces acting on, and properties of, its neighbouring particles. The further away a particle is from another the less its effect on it up to 2 times the smoothing length. Beyond this length the contribution is zero (see Figures 2.1 and 2.2). This is similar to the RBF method. The particles' properties are smoothed over

this distance by a smoothing kernel such as a spline function. A smoothing kernel can be thought of as basically a normalised weighting function. It is from this "smoothing out" of properties over a defined range (usually given the label *h*) from which SPH gets its name. Another way of looking at it is that SPH treats each gas particle as an extended cloud with the smoothing length representing the spatial extent of each cloud. SPH's use of particles allows it to improve the resolution of areas where the fluid is dense, i.e. where there are a lot of particles. The smoothing length will be very small in these areas while it will be large where there are few particles, e.g. voids. Usually this length will be set based on the local density on the previous timestep. Typically the smoothing length will be set to alter so that it includes approximately the thirty nearest neighbours to the particle in question. This high resolution in dense areas – low resolution in diffuse areas means that the method does not simulate large areas of space where there are no particles and concentrates on the regions of interest, unlike some mesh based methods which have the same resolution (set by the size of the grid) at all times. This is beneficial in reducing the cost of the simulation. This could be considered analogous to the adaptivity present in modern mesh-based CFD solutions via adaptive mesh refinement (AMR) techniques.

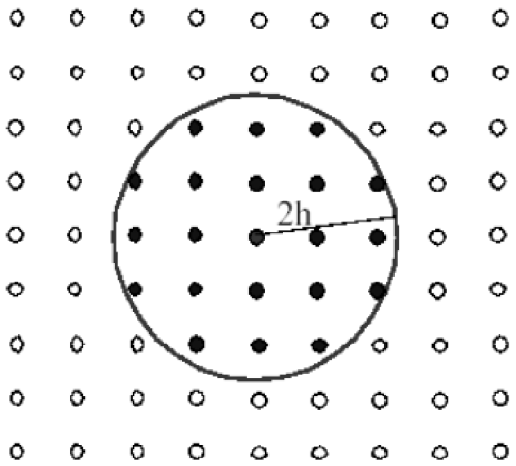


Figure 2.1. Demonstration of the smoothing length. All particles in bold are within range of the central one and so contribute an amount to its properties dependant on their distance away (Vignjevic, 2004).

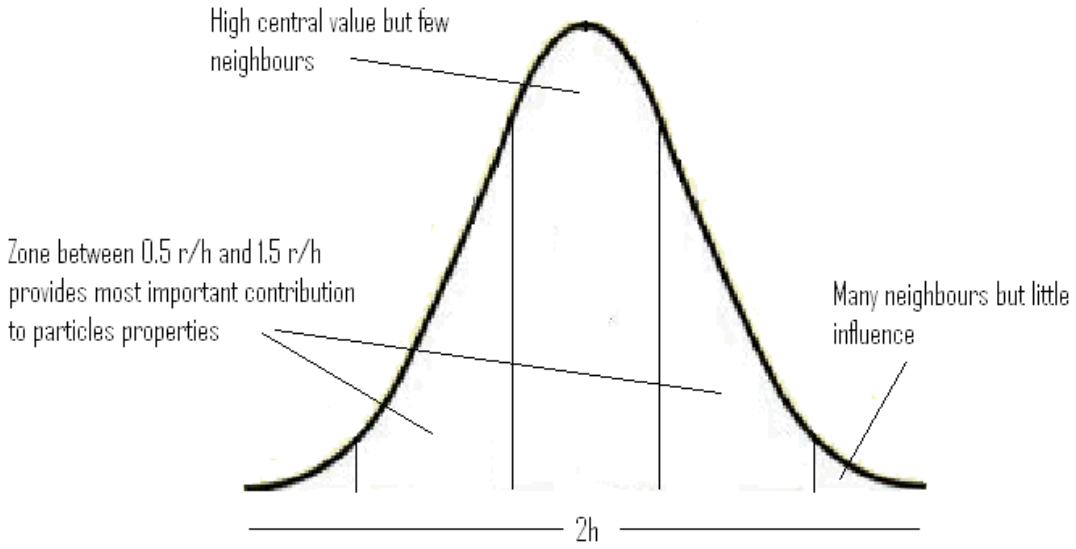


Figure 2.2. Picture showing how particles properties are smoothed by a kernel. In reality (no smoothing) this would take the form of a single, thin tall spike in the centre of the curve where the particle is located.

An SPH simulation is essentially a 2 stage process (per timestep). In the first pass neighbouring particles are found and local density and pressure found using the kernel. An index of all the current particles is made. In the second pass the application of forces is considered and such things as acceleration and temperature changes are made. This is different to other particle methods such as P^3M (explained in more detail in chapter 3) where only one pass is used because of the presence of a smoothing length.

2.2.1 Advantages of SPH

Due in a large way to its particle based nature SPH has several advantages over mesh based CFD methods. Some of these are listed below:

- It can handle very large density variations – simulations have been known to have a density range in excess of 6 orders of magnitude.

- Voids require no special treatment.
- It can handle large deformations easily. This is useful in, for example, the simulation of an explosion. A grid would find it difficult to deform by a sufficient amount but SPH of course has no such problem.
- SPH codes can create complex geometry. This is due in no small part to the lack of a grid and the fact that the physics is described simply by using differential equations. There is also no limit on the amount which the system can evolve from the initial conditions.
- Locations of interfaces between different fluids or vacuums are clear. There is no need for explicit interface tracking.
- It is relatively simple to make an SPH code 3-Dimensional. Indeed many SPH codes are naturally 3D and were designed that way.
- Individual particles can be tracked throughout the evolution of the system and traced back to their origin.

SPH is very good if you are interested in finding out where the mass is in a simulation although it does rely on a good local density estimate. The density estimate is always crucial to an SPH simulation.

2.2.2 Disadvantages of SPH

It would be remiss if it were not pointed out that SPH does have some problems. Certain areas of simulation are hard to model using an SPH code. These are highlighted below. Of course efforts are constantly being made to try and improve SPH's handling of such matters:

- Interfaces. While a free surface boundary is easy to track in SPH due to its particle nature, interfaces such as gas/void or free surfaces such as water/air can still be problematic in SPH. This is due to the sudden change in the density, some of the

particles smoothing lengths will extend over the interface causing the densities of both sides to be smoothed.

- Shocks and contact discontinuities. A shock is a very steep change of density propagating through a medium. A similar problem to the one described above occurs here, the shock is smeared out by a small amount due to the particles either side of the shock front searching for neighbours and influencing their properties. This will be looked at later on in this thesis.
- Artificial viscosity has always been a problem for SPH codes. It is necessary to add in viscosity to a simulation but getting it to match up with real viscosity is very hard.
- Turbulence remains a big problem to truly accurate modelling. To achieve a realistic simulation of turbulence in, say, an open river channel requires enormous accuracy and even so we are not entirely sure of all the physics concerning the energy dissipation involved in turbulence. Turbulence is clearly present in a scenario such as a dambreak and although not represented directly in this research it is considered and discussed in chapter 4.

One of the important (and surprising) things to note about these disadvantages, especially with regards to the first two listed, is that it is the *smoothed* nature of the *smoothed* particle hydrodynamics method that causes the problems! Sharp changes in density are blurred, or smoothed, out over a small distance due to the smoothing length finding the nearest thirty neighbours regardless of sudden changes. The size of this effect is dependent on particle number in the simulation – the more particles the less the effect will be noticed. It is impossible however to completely eradicate this effect.

One other final point that should be noted here is the robustness of the SPH method. Under normal circumstances having a robust method is a big positive point; however it can occasionally be a double edged sword. SPH codes have a habit of being so robust that they work in any situation even if a mistake has been made in the code. This means that sometimes the code may not break down when something goes wrong but just carry on to

produce an incorrect result. To combat this, the researcher must always check the results by visualisation to see that they make sense and look physically "real" before accepting them as the result.

2.3 Open Channel Flow

Any channel whether it is naturally occurring, such as a river or ditch, or artificial, such as a canal or sewer, can be considered to be an open channel as long as it is has a free surface open to atmospheric pressure. This distinguishes it from a pipe flow which has no free surface. Whilst any fluid can flow along an open channel, water is by far the most typically seen and therefore modelled fluid. In an open channel flow the forces usually considered are gravity, inertia and viscosity. One complication of naturally occurring open channels (and sometimes artificial ones) is that their cross-sections are often arbitrary shapes instead of round or square as is typical in pipes. The position of the free surface in open channel flow tends to vary with time and space. Depth of water is often variable as well as is the slope gradient which in turn affects the discharge of the channel. Uniform flows are almost never seen in nature, unsteady flows are much more common. The flow characteristics of the channel have to be found from solving both continuity and momentum equations.

Continuity says that for incompressible flows the mass flow rate entering a channel (or section of a channel) must equal the mass flow rate leaving the channel, i.e.

$$\rho Q_{entering} = \rho Q_{leaving} \tag{2}$$

where ρ is density and Q is the volume flow rate usually measured in m^3/s.

As the volume flow rate depends on the area normal to the flow direction and the average velocity of the flow passing through that slice it can be said that,

$$Q_{entering} = u_{entering} A_{entering} \quad \text{and} \quad Q_{leaving} = u_{leaving} A_{leaving} \tag{3}$$

where A is the cross-sectional area of the channel at that point and u is the average velocity.

The average velocity is used because the actual velocity of an open channel flow varies across A. The velocity of the flow is actually zero at the wetted boundary of the channel where it touches the three solid walls. The flow velocity increases to a maximum at the centre of the channel and just below the surface. The region near the wetted boundary is called the boundary layer, and the most dramatic change in velocity takes place across this region.

By considering the continuity equation and Newton's second law which states that force (F) is the rate of change of momentum it can be shown that the momentum equation can be expressed as,

$$F = \rho Q(V_{leaving} - V_{entering}) \tag{4}$$

where V is velocity assuming velocity is uniform across A.

In real flows a coefficient β must be used to account for the fact that velocity is not uniform over A making,

$$F = \rho Q \beta (V_{leaving} - V_{entering}) \tag{5}$$

$$\beta = \frac{\int \rho u^2 dA}{\rho V^2 A} \tag{6}$$

Energy between the upstream and downstream cross-sectional slices is related by the Bernoulli equation

$$\frac{P_1}{\rho g} + \frac{u_1^2}{2g} + z_1 = \frac{P_2}{\rho g} + \frac{u_2^2}{2g} + z_2 = const. \quad (7)$$

where P is pressure, g is gravity and z is the height of the geometric centre of the slice. Subscript letter 1 denotes entering slice, subscript 2 denotes leaving slice.

Other considerations that must be taken into account when considering open channel flows are whether the flow is supercritical (controlled from upstream) or subcritical (controlled from downstream). This is determined by the Froude number

$$Fr = \frac{u}{\sqrt{gL}} \quad (8)$$

where d is depth and L is hydraulic radius.

If Fr is >1 the flow is supercritical.

The Reynolds number is used to measure the ration of internal forces. A laminar flow is characterised by a low Reynolds number while a turbulent flow is characterised by a high Reynolds number.

$$Re = \frac{\rho u L}{\mu} \quad (9)$$

where μ is the dynamic viscosity of the fluid.

2.4 Theory of SPH

Presented in this section is a review of the major aspects of SPH theory. Several good reviews of SPH already exist such as Monaghan (1992); Monaghan (2005); Issa PhD thesis (2004); Crespo et al (2007); Gomez-Gesteira and Dalrymple (2004) to name a few authors.

Notation used by SPH researchers varies; this present work uses the same notation as usually used by Monaghan (Monaghan, 1994).

2.4.1 Basic Equations

At the heart of SPH lies the interpolation method which allows any function to be expressed in terms of its values at a set of disordered points – the particles. The integral interpolant *A(r)* is defined as:

$$A\ (r) = \int A(r')W(r - r', h)dr' \tag{10}$$

This integration is performed over the entire space, where h is the smoothing length and W is an interpolating kernel. The smoothing length automatically adjusts itself so that the nearest 32 neighbours (in the code used here) are contained within the smoothing length. In discrete notation, equation 10 becomes

$$A(r) = \sum_b m_b \frac{A_b}{\rho_b} W_{ab} \tag{11}$$

Where m is mass, ρ is density, subscript letters denote particles and W_{ab} denotes $W(r_a\text{-}r_b, h)$.

The kernel has the properties;

$$\int W(r - r', h)dr' = 1 \tag{12}$$

i.e. it is normalised, and

$$\lim_{h \to 0} W(r - r', h) = \delta(r - r') \tag{13}$$

In the early days of SPH it was usual for the kernel to be a Gaussian function unless specified otherwise. These days a cubic spline function is a much more popular choice of kernel though others do exist. The kernel essentially plays the role of a weighting function and thereby determines the level of contribution a particle has on a neighbour based on their separation. Modern kernels (such as the one used by Hydra) have the advantage of compact support, limiting the range of the forces (out to 2*h*) and hence the neighbour search length to a small fraction of the volume. Hydra uses a cubic-spline kernel, based upon that developed by Monaghan & Lattanzio (1985), which has been normalised in 2 dimensions in equation 14.

$$W(r,h) = \frac{5}{14\pi}\begin{cases} 4 - 6x^2 + 3x^3 & 0 \le x \le 1 \\ (2-x)^3 & 1 \le x \le 2 \\ 0 & else \end{cases} \tag{14}$$

where $x = \frac{r}{h}$ is the ratio of the particle separation to the smoothing length.

Kernels will be examined in more detail later in this thesis. A full mathematical proof of the normalisation process for the kernel used by Hydra is contained within Appendix A.

Density is estimated everywhere by the following function:

$$\rho(r) = \sum_b m_b W(r - r_b, h) \tag{15}$$

where ρ is the local mean density and m is the particle mass.

If smoothing length is constant the density estimate in equation 15 can be integrated to give

$$\int \rho(r) dr = \sum_b m_b = M \tag{16}$$

where M is total mass.

If the smoothing length is allowed to vary during the simulation this is no longer implicitly true but the errors will remain very small. This is due to the possibility of a slight difference between sum (m) and the integral (m*W) when the smoothing length can extend outside the simulation boundaries. Provided no particles escape a simulation, mass is always conserved due to the number of particles and the mass-per-particle being fixed.

2.4.2 The Equations of Motion

Central to the simulation are the equations of motion (momentum, continuity and thermal energy equations) which control the movement of the particles. Using the interpolation framework presented above, these can be formulated reasonably simply. Beginning with the momentum equation, the pressure gradient is estimated as follows;

$$\rho_a \nabla P_a = \sum_b m_b (P_b - P_a) \nabla_a W_{ab} \tag{17}$$

where P is the pressure.

Equation 17, however, is not acceptable because momentum is not exactly conserved at all times. By utilising the fact that

$$\frac{\nabla P}{P} = \nabla\left(\frac{P}{\rho}\right) + \frac{P}{\rho^2}\nabla\rho \tag{18}$$

And combining it with equation 11, the momentum equation becomes

$$\frac{dv_a}{dt} = -\sum_b m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right) \nabla_a W_{ab} \tag{19}$$

where v is the velocity.

Equation 19 is sometimes expressed as the acceleration equation and can be considered an SPH formulism of the continuous Lagrangian form of the Navier-Stokes equation for a weakly compressible flow. Another way of formulating this conservation concept is to consider combining the equation used to find the density (equation 15) with the first law of thermodynamics to produce the above which is the SPH equivalent of the standard equation of motion, i.e.

$$\frac{dv}{dt} = -\frac{1}{\rho}\nabla P \tag{20}$$

By using this version of the momentum equation it can be shown that the force on particle a from particle b can be written as

$$F_{ab} = \frac{2m_a m_b}{h^2}\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)(r_a - r_b)\nabla_a W_{ab} \tag{21}$$

The continuity equation

$$\rho_a = \sum_b m_b W_{ab} \tag{22}$$

is automatically satisfied as the mass is carried by an unchanging number of tracers of fixed mass. This has the advantage that another differential equation does not need to be solved. Compressibility can be adjusted by changing the speed of sound, a property which will become very important when the code is to be modified for water simulation.

A brief derivation of the thermal energy equation is presented here for completeness though it should be noted that these are not needed for the simulation of water using SPH and so are

discarded for the bulk of the work in this thesis. The rate of change for thermal energy can be written

$$\frac{du}{dt} = -\left(\frac{P}{\rho}\right)\nabla \bullet v \tag{23}$$

where u is thermal energy per unit mass.

Equation 23 can be written for SPH in the form

$$\frac{du_a}{dt} = \left(\frac{P_a}{\rho_a^2}\right)\sum_b m_b v_{ab} \bullet \nabla_a W_{ab} \tag{24}$$

Or

$$\frac{du}{dt} = -\nabla\left(\frac{Pv}{\rho}\right) + v \bullet \nabla\left(\frac{P}{\rho}\right) \tag{25}$$

Averaging equations 24 and 25 gives us

$$\frac{du_a}{dt} = \frac{1}{2}\sum_b m_b\left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2}\right)v_{ab} \bullet \nabla_a W_{ab} \tag{26}$$

2.4.3 Moving the particles

In standard SPH formulism particles are moved using the simple equation

$$\frac{dr_a}{dt} = v_a \tag{27}$$

However in the present work equation 27 is insufficient so we have made use of the XSPH variant where a velocity averaging scheme is used (Monaghan, 1989).

$$\frac{dr_a}{dt} = v_a + \varepsilon \sum m_b \left(\frac{v_{ba}}{\overline{\rho}_{ab}} \right) W_{ab} \tag{28}$$

where ε is 0.5 and $\overline{\rho}_{ab}$ is the average density between particles a and b.

This variant on standard SPH causes particles to be moved at a velocity that is closer to the average velocity of its neighbours. The idea behind XSPH is that the particles in a weakly compressible flow will move in an ordered way more consistent with liquid flow.

The Equation of state (EOS) of a gas can be represented by several different gas equations, the most common being the ideal gas equation

$$PV^{\gamma} = const. \tag{29}$$

where γ is 5/3.

Equation 29 (or similar) allows for a compressible fluid but does not work for a liquid which is essentially incompressible (Monaghan, 1994). As such the traditional SPH method must be modified for use in the open channel flow simulations described in this thesis. In order to create liquid flow a much stiffer equation of state must be constructed to ensure that the fluid contains enough internal pressure to support itself and maintain volume. In this work the EOS described by Monaghan (1994) and originally proposed by Batchelor (1974) has been used where

$$P = B\left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] \tag{30}$$

Here γ is set to 7 and ρ_0 is a reference density of 1000 kg m^{-3}. B is a coefficient which changes in different scenarios and is constructed thus,

$$B = \frac{C_0^2 \rho_0}{\gamma} \tag{31}$$

where C_0 is the speed of sound. This formula for pressure ensures that the fluid is kept very stiff though not fully incompressible. It is described as weakly compressible. B can be varied by alteration of the speed of sound. In SPH the sound speed is set before each simulation and does not represent the speed of sound in nature, rather the speed at which information can travel. C_0 is set to be large enough so that the density fluctuations are kept less than 1%.

$$\frac{|\delta\rho|}{\rho} \approx \frac{v^2}{c_0^2} \tag{32}$$

where υ is the velocity of waves inside the fluid.

By considering equation 32 density fluctuations of 1% or less can be achieved when v/c <0.1. Therefore at the beginning of a simulation sound speed is set approximately 10 times higher than the highest velocity of physical waves in the fluid. Observations of the simulations presented in this work indicate that overestimating the speed of sound at this stage does not adversely affect the solution but an underestimation can be fatal. This requires a reasonable estimate of the speed of the fluid before the simulation begins though often this is not a problem. For example in a water column collapse the maximum wave speed expected is proportional to the height of the column (Martin and Moyce, 1952) and can be predicted by the formula

$$v = \sqrt{2gH} \tag{33}$$

where g is gravitational acceleration and H is the height of the column (Monaghan, 1992).

The EOS in equation 30 has been extensively tested by several authors (e.g. Monaghan, 1994; Violeau and Issa, 2007) and is consistently used in SPH simulations though it is possible that other formulations of weakly compressible fluid EOS's may also function adequately. This is not the only method of pressure estimation that can be used in SPH to model liquids. Some researchers (e.g. Shou and Gotoh, 2004; Shao, 2006) have chosen to do away with an equation of state and solve a pressure Poisson equation instead. This has the advantage that it is fully incompressible but does add in additional numerical operations and is more complicated mathematically (Violeau and Issa, 2007). This approach is not considered here.

2.4.4 Artificial Viscosity

This is a difficult area in SPH, which has had several groups proposing solutions, none of which have been considered to be fully satisfactory though the introduction of a viscous force is most commonly used. Introduction of artificial viscosity is necessary in SPH simulations to stabilise ringing created by shocks and to prevent interpenetration of streaming flows. Whilst common in astrophysics these phenomena occur less in water simulations but the subsequent loss of stability is balanced by use of the XSPH variant described in section 2.3.3. Presented next is the most common way of expressing artificial viscosity in SPH (Monaghan, 1992) and is the formulation that has been used in this thesis. The momentum equation (equation 19) is re-written as equation 26:

$$\frac{dv_a}{dt} = -\sum m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a W_{ab} \tag{34}$$

In this example;

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha \bar{c}_{ab} \mu_{ab} + \beta \mu_{ab}^2}{\bar{\rho}_{ab}} & v_{ab} \bullet r_{ab} < 0 \\ 0 & v_{ab} \bullet r_{ab} > 0 \end{cases} \tag{35}$$

and

$$\mu_{ab} = \frac{h v_{ab} \cdot r_{ab}}{r_{ab}^2 + \eta^2} \tag{36}$$

where α and β are constants (chosen as 0.01 and 0 for the research presented in this thesis) while η is a small number (a fraction of the smoothing length) included to prevent singularities.

The linear term produces a shear and bulk viscosity while the quadratic term is present to handle high Mach number shocks and is roughly equivalent to the Von Neumann-Richtmyer used in finite difference methods (Monaghan, 1992). This form of artificial velocity is beneficial since it conserves momentum and is Galilean invariant.

2.5 Uses of SPH

In the years immediately after its creation in 1977, SPH remained a simulation tool for astrophysicists wishing to model stars and gas clouds and other gaseous objects in deep space. However in recent years it has started to come into focus as a viable technique for simulation in many other fields. Thus, it has become an alternative to traditional CFD. Astronomers remain at the forefront of the SPH field due to their experience with the codes and

astrophysical simulation continues to be the most common use of SPH and provides the largest and most complex simulations. Nevertheless, several groups of researchers have turned SPH into a valuable tool for many applications based on Earth, beginning with Solid dynamics in the early nineties, and quickly afterwards, fluid dynamics (Randles and Libersky, 1996; Monaghan, 1992). The research being described in this Thesis for example is an attempt to take an originally astrophysics code and modify it to be of use within a civil engineering context (specifically water flow). In this section, there is a list of main areas of research and industry that have made use of SPH codes and a brief description of these fields and how SPH has fitted in. References are given as examples but should not be considered an exhaustive list. Some industries have been using SPH for CFD purposes for some years now while others are only just coming to realise its potential.

2.5.1 Astrophysics

There are several different types of problem that have been attempted using SPH and the main ones are outlined in this section. The variety and range of astrophysics' simulations that can be performed using SPH is an indicator in itself of SPH's robustness and versatility. As astrophysicists have been using the method for around thirty years now, quite a few different codes have been released and several coding "tricks" to improve and speed up simulations have been found. This is made apparent when comparing current state-of-the-art astrophysics simulations to many other fields. Many astrophysics simulations can be seen to have far greater particle numbers and much more visually realistic looking effects.

2.5.1.1 Black Holes, Neutron Stars and Accretion Disks

When a massive star has reached the end of its life and gone supernova it will sometimes implode on itself to form an extremely dense object called a neutron star or possibly an even denser object known as a black hole. When there is something near to, say, a black hole, such

as a binary star companion, material is pulled toward the black hole and dragged into a rotating disk around the object as it spirals into the actual hole. This is called the accretion disk. Black hole mergers could be the key to discovering gravitational waves which would be a powerful test of Einstein's general theory of relativity (Cnet.com article). Several scenarios have been modelled by various researchers round the world, such as black hole collision (Dotti, Colpi and Haardt, 2007), neutron star collision (Faber and Rasio, 2002) and black hole/neutron star collision. Black hole simulations can also be said to include the modelling of supermassive black holes (extremely large black holes that are found in the centre of galaxies) and these have been studied in some detail as well. Also simulated in recent years is the formation of accretion disks and how matter reacts when it is pulled into the intense gravity of such an object (Gerardi, Molteni and Teresi, 2005; Belvedere and Lanzafame, 2002).

2.5.1.2 Galaxy Mergers and Gas Cloud Interaction

Astronomers want to understand the physics behind galactic mergers and collisions in order to gain a better understanding of why our universe and the galaxies within it look the way they do. This topic has been made even more interesting since the discovery that galaxies contain supermassive black holes in their cores and these will naturally play a large part in determining the fate of both galaxies in the event of a collision. The simplest way to approximate a galactic collision is to replace the galaxies with very large clouds of gas. This is a much less daunting prospect than modelling an entire galaxy in detail. It should also be considered that the interaction between two large gas clouds includes some interesting physics of its own and provides some challenges to the SPH researcher. Examples of this include the need for the code to be able to handle large shocks as one cloud hits another at high speed something which has been a topic of research in its own right, for example Pfrommer et al (2006) and McCarthy et al (2007). Such interactions have been studied by many SPH researchers for years with Navarro and Benz (1991) and Barnes and Hernquist (1991) to name but a few.

Simulation of a complete galaxy with each star represented by a SPH particle is a goal of several researchers but it is not yet within our ability. The addition of the supermassive black hole to any galaxy simulation does increase the realism of the model but does also increase the complexity as well (Matsui, Habe and Saitoh, 2007).

2.5.1.3 Star Formation

Star formation occurs when large clouds of molecular gas (hydrogen and helium) become unstable and begin to collapse. This causes the gas to heat up into plasma. When sufficient density and temperature have been reached nuclear fusion begins in the centre of the plasma. Star formation has been simulated in some detail using the SPH method, for example Gittins, Clarke and Bate (2003) or Kitsionas, Whitworth and Klessen (2007).

2.5.1.4 Supernovae

When a star nears the end of its life, it begins to swell up to far greater than its former size due to the fusion of heavier and heavier elements in its core. Once the star runs out of fuel the outer layers of the star are blown off and the core collapses down to a white dwarf star. If however the star is very large, then the core can implode at a much faster rate and this in turn causes an immense explosion called a supernova. These explosions are some of the largest in the Universe and of great interest to astronomers who use simulations to recreate the events in order to explain the physics behind them (Wiersma et al, 2009).

2.5.1.5 Whole Universe Simulations

Since its inception SPH simulations have been increasing in scale and size as computer processing power has increased. In recent years, computer power has developed sufficiently to set up simulations of the entire Universe. These simulations are some of the largest SPH

simulations ever carried out involving over a billion SPH particles. These simulations start out in a very young universe and progress through billions of years allowing the string like structure that we observe from the galaxy clusters and superclusters in the universe today. The particles can clearly be seen to clump together to form clusters of galaxies in the manner in which galaxies have been observed by astronomers. These simulations have been carried out on some extremely powerful supercomputers around the world and have been done in both "dark matter only" form and also in gas plus dark matter form incorporating the physics of dark matter as best we understand it. These simulations are collectively known as the millennium simulations and performed by researchers from the Virgo Consortium (Virgo Consortium).

2.5.2 Water

Originally invented to study non-spherical stars and nuclear fusion reactions within them, it was quickly realised that SPH could be used to study a wide range of astrophysical fluids (all compressible gases). It took some time, however, before SPH was considered as a means of modelling incompressible flow here on Earth (Monaghan, 1992). Any liquid can be modelled as an incompressible flow but generally it is the simulation of water that is considered most often.

2.5.2.1 Underwater Explosions

With SPH being considered as a valuable simulation method for fluid mechanics on Earth and for shaped explosions on land, a natural step was to see if the method could be used to simulate explosions underwater (Liu and Liu, 2003). This is very much an area still to be researched and not much seems to have been completed in it. Nevertheless underwater explosions seem to be a valid problem to be tackled by SPH because they require large deformations of the fluid and a moving boundary between heated gas and water to be

accurately tracked (Liu et al, 2003). Once a simulation of sufficient size has been created the explosion itself causes problems. The explosion causes a strong shock to move through the water in all directions causing drastic changes in water current. This being said, the actual event could be considered similar to a Sedov blast (Sedov, 1959) which is considered an important test of a hydrodynamical code (Tasker et al, 2008).

2.5.2.2 Dam Bursts

A dam burst is a good way of testing a CFD code and has been modelled many times and performed in laboratory experiments (Monaghan, 1994; Colagrossi and Landrini, 2003; Gomez-Gesteira and Dalrymple, 2004; Violeau and Issa, 2007 to name a few authors). As well as being a good code test it is also important to know how and where water will flow in the event of a dam or levee breach so that proper flooding contingency plans can be devised, e.g. so you would know where not to construct homes/buildings. In this respect it is a very important topic to be able to simulate accurately. Considerable time has been devoted to this very problem by the current author and the results of this research are detailed in chapter 4 of this thesis along with comparisons with the results of the authors named above.

2.5.2.3 Waves and Tsunami

The modelling of waves crashing and breaking and the resultant splashing of water is something that traditional mesh based methods have trouble with (Gomex-Gesteira et al, 2004). Accurate modelling of ocean waves and how they impact on structures can provide useful information for storm defence builders and ship designers alike (Crespo, Gomez-Gesteira and Dalrymple, 2007a). Tsunamis (or "harbour waves") are a different matter. SPH has become a valuable tool in simulating these large solitary waves and since the recent deadly Tsunami on Boxing Day 2004 research into this phenomenon has increased (e.g. swri.org article). Tsunami can be caused by undersea earthquakes, landslides or volcanic eruptions. SPH's versatility in being able to simulate all of these events makes it a powerful

tool in helping to understand Tsunami which in turn might help us develop a better warning system for them.

On a more historic side, SPH simulations performed by J. Monaghan, one of the founders of SPH, have given strong credence to the theory that the ancient Minoan civilisation was destroyed by a Tsunami (Monaghan and Kos, 1999). 3500 years ago the ancient civilisation was thriving on the Greek island of Crete when it was abruptly torn apart. This occurred at the same time as a large volcanic eruption over a hundred kilometres away. The SPH simulations showed that a tsunami could have been produced which would have headed in the direction of the Minoans. The damage caused from that event is believed to be responsible for the downfall and eventual extinction of the civilisation.

2.5.3 Manufacturing Industries

The term manufacturing can be applied to a great many things but what is common to all companies/organisations that are involved in it is that they all are constantly striving to cut their costs in the manufacturing process. Researching how to manufacture a new product can prove expensive to do and if a new technique does not work you will have wasted time and money in testing it. Being able to simulate a process allows a company to accurately predict what will happen in a new technique without having to actually do it or stop manufacturing in the old method.

SPH has had its most notable and widespread success in the industries that involve liquid moulding or die casting where liquid metal is injected into a pre-made mould to make a large number of identical solid metal parts. SPH has also been put to use in high performance vehicle part's modelling.

2.5.3.1 Vehicle Parts

- Fuel Tanks

A BAE Systems/McLaren Mercedes partnership has recently started researching several new technologies in order to improve the performance of the McLaren Mercedes Formula 1 (F1) racing car's performance on the track. Among them is SPH modelling of fluid motion within the fuel tanks and fuel pipes (Engineeringtalk.com).

Using SPH to accurately model the fuel flow through pipes in the engine and sloshing in the tank could help the car designers to better calculate the fuel remaining in the car and also help optimise the car's centre of gravity with a goal to decrease the car's lap time around the F1 circuit.

This research is a continuation of BAE Systems' work on SPH modelling of aircraft fuel tanks where they have had some success. The motion of fuel in a jet aircraft's fuel tank would not be all that different to that in a F1 car, except that the forces and accelerations generated from the aircraft's motion would be more extreme due to the jet moving considerably faster. The modelling of bird strike on an aircraft engine is also an area of interest to BAE systems who plan to use their proprietary SPH code to simulate this as well (BAE systems report).

On a similar theme in BAE systems/McLaren Mercedes' partnership, there is current research (begun 2005) to use the techniques learnt with the fuel tank simulations to model oil flow through pipes in the gearbox. They have used SPH to study how the oil moves and to use that information to try and improve the design of the gearbox for the F1 car again with the intention of improving circuit performance. Improvements designed from SPH simulations were included in the 2006 World Championship McLaren Mercedes F1 car (BAE systems report 2).

- Tyres

Yokohama Rubber Company, one of the world's leading manufacturers of rubber products including vehicle tyres, have begun to use SPH simulations to analyse water displacement of tyres in their product development department, the first such company to do so (Yokohama Rubber Co. Ltd., 2002). With conventional finite element methods it is virtually impossible to track water flows which change constantly with tyre rotation. SPH methods allow analysis of water flows within tyre grooves and accurate modelling of the direction of water sprayed from those grooves during tyre rotation (Yokohama press report).

2.5.3.2 High Pressure Die Casting

High Pressure Die casting (HPDC) is a versatile process for producing engineered metal parts by forcing molten metal under high pressure into reusable steel moulds. Metallic parts made using a die casting method can be seen in products ranging from cars to plumbing tools to toys. As the flow of liquid metal (usually aluminium, zinc or copper though some alloys are also used) into the cast is done at very high pressure into a structure of, sometimes complex, boundaries there is a lot of splashing. This is an area that SPH can simulate well (unlike many CFD methods). The mould needs to be filled quickly and uniformly and certain methods/entry points may prove better than others for this. SPH simulation can reduce costs by helping the manufacturers predict what the best way of creating their products will be using the HPDC method without the hassle and expense of actually performing the process. For an excellent introduction to HPDC and SPH's valuable role within it the best place to go is the CMIS/CSIRO collaboration website (www.cmis.csiro.au/cfd/sph/index.htm) or see Cleary et al (2004) or Ha et al (2003).

2.5.4 Shaped Explosions

Explosives are used in many areas of both modern industry and the armed forces. In the last few decades great advances have been made in the field of shaped charges or explosive devices that channel their energy in a particular desired direction. These types of explosive devices are used in a variety of applications. For example they may be used in military technology, breaching walls, building demolition and oil well perforation. It is important that we are able to simulate these events accurately because a shaped explosion is most likely going to require a precise detonation, i.e. enough charge must be used to complete the objective but not cause large amounts of collateral damage. Another good reason to be able to simulate an explosion is to be able to model an accidental or deliberate detonation in a built up environment. SPH is ideal for such simulations due to its absence of a grid. Codes which use a mesh often have difficulty modelling multidimensional problems with large deformations, something which a large or powerful explosion is bound to have (Liu and Liu, 2003).

Also, a good place for more information is the ANSYS (the developers of the popular CFD simulation code CFX and recently Fluent) website and look at their AUTODYN code pages (ANSYS, 2007). AUTODYN incorporates SPH into it, combines it with other CFD methods and specialises in modelling non-linear dynamics whether it be with gases, liquids or solids.

2.5.5 Solid Body Impact with Water

A solid body impacting with water (or any liquid) is still considered a difficult problem to simulate even if it is simplified to a simple geometric shape impacting on a still, infinite body of water (which is rarely the case in reality). There is not much in the way of analytical solutions to these types of problems, due to the sheer number and variety of different possible combinations of factors that may occur, but SPH simulations tend to agree well with

experiments. Examples include avalanches into water (Monaghan, Kos and Issa, 2003), ship's hull hitting the ocean (Groenenboom, 2008), various items (e.g. a car) sliding into ocean/river/lake off an edge or cliff (Jones and Belton, 2006).

2.5.6 Armour Research

Recent research indicates that it is possible to study the effectiveness of different types of armour (for a person or machine) via SPH simulations. A search through the literature reveals SPH simulations in the armour research field, specifically dealing with the impact of 5.56mm calibre bullet of the kind used by the NATO armed forces on ceramic/Kevlar body armour (preliminary report, university of Lisbon).

2.5.7 Deformation and Impacts

- Traditional

Impacts are not something that has been often considered when using an SPH code but recent attention has been given to the simulation of what are known as hypervelocity impacts. This could be a bullet being fired from a gun, involving gas flow, solid body motion at high speed and shock problems (Das and Cleary, 2008) to even higher speed problems of impact perhaps involving a small body coming from orbit (a hypervelocity impact).

- Moon-Forming Impact

A popular theory regarding the formation of the Earth's moon is that when the earth was very young it was struck by another proto-planetary object about the size of the planet Mars. This immense collision probably happened at an incident angle (i.e. not head-on) and would have

caused tremendous damage to the other planet and substantial damage to earth. It is believed that the debris blown off of the Earth's crust that was created by this collision was sent up into orbit around the earth and this eventually coalesced to form the moon. SPH simulations (Wada, Kokubo and Makido, 2006) have shown that such a collision could provide Earth with a moon of the correct properties.

2.5.8 Multiphase Flow

Multiphase flows are very common in nature yet they can prove to be very tricky to simulate on a computer. SPH has shown some promise in simulating multiphase flow and this remains an active area of research.

Examples of multiphase flow include (but are not limited to) volcano eruption leading to lava flow, a river carrying sediment, droplets of oil in water or air, and a "lava-lamp" (Muller et al, 2005) and galaxy formation (Ritchie and Thomas, 2001; Hu and Adams, 2006).

2.5.9 Medical Research

Recently the idea has been put forward that various medical, surgical or biological processes could be simulated on a computer using SPH (Allard et al, 2007). This would have the advantage that surgeons, for example, could see what effect some procedure might have or that the flow of blood through an artery could be modelled (Muller, Schirm and Teschner, 2004). This research is still in the early stages but has had some positive initial results. There are, of course, a wide range of complexities in trying to simulate a biological process or medical procedure such as dealing with multiphase flows, boundary conditions and deformations (sometimes to materials that are "spongy" and not quite solid).

2.5.10 Turbulence

Turbulence is a very difficult physical phenomenon to model and even today parts of it are quite poorly understood even though it occurs all over nature. Indeed, some have labelled it one of the most difficult problems of the last century. Nevertheless turbulence modelling has started to be considered by researchers using SPH methods beginning, to the best of the author's knowledge, with Violeau, Piccon and Chabard (2001). Recently a new approach has been proposed by incorporating the Alpha turbulence model into SPH by Monaghan (2002). This model is compatible with SPH and research is currently underway to include turbulence effects in SPH. There have also been moves to introduce a k-ε turbulence model into SPH by researchers at EDF energy (Violeau and Issa, 2007).

2.5.11 Entertainment Industry

The SPH method has been modified for use within the entertainment industry. Most notably in the special effects that are seen in feature films, TV adverts and computer games. The ability to realistically model fluids (especially liquids) is a valuable commodity in these fields. The codes used in the entertainment industry tend to be custom designed for the project that is being simulated and often require complicated boundaries and deformable scenery.

Several major Hollywood motion pictures have used SPH (or a derivative of it) in their computer generated special effects. For example:

- Water flow through New York City in "The Day After Tomorrow".
- Lava flow in "The Lord of the Rings – The Return of the King".
- Water flow in "Ice Age 2".

More information about these and a wide variety of movie clips that use SPH techniques as part of the simulation process can be found on the website of a company called Next Limit (www.nextlimit.com) who specialise in the creation of realistic fluid motion simulations. Several movies exist here for both scientific and entertainment purposes made using their proprietary software RealFlow, which is the leading 3D simulation and rendering software in the world.

Chapter 3

Hydra: Conversion and Verification

3.1 Hydra

3.1.1 Development

The computational code that has been identified for conversion from astrophysics based to water based is called Hydra (Couchman et al, 1997). Hydra has the advantages of considerable previous testing and years of refinement (see section 3.1.3 for more details). The version of Hydra as used in this research is coded in the Fortran programming language and implemented in serial. The development of Hydra began with looking at N-body (dynamic systems of particles under physical forces) simulation methods. The earliest such codes used a variety of methods, the ones of interest here being the Particle-mesh (PM) method, the Particle-particle (PP) method and the treecode (TC) method.

In the PM method particles were converted into a mesh often weighted according to their distance to the grid points (e.g. by cloud in cell method). The particle mass is turned into a density in the cell. This reduced mass distribution to matrix form where Poisson's equation

$$\nabla^2 \Phi = 4\pi G\rho \tag{37}$$

where G is the gravitational constant, ρ is the density, Φ is gravitational potential.

can be used to find the potential energy of the cells and the forces applied to each particle based on the cell it has been mapped onto.

The PP method is perhaps the simplest and instead relies on direct summation to calculate the forces between two particles. This may provide accurate results but is very computationally expensive.

In The TC method the volume is divided up into cells as before but only particles inside nearby cells are considered while cells far away from the region of interest are combined to form a single large area with a common centre of gravity. Typically the eight neighbouring cells to the one in question are used. This saves time and cost when simulating. This method has been successfully combined with SPH (Tree-SPH) for use in astrophysics simulations, for example in the Gadget2 code (Springel, 2005).

Hydra has been developed along a different route to Gadget2 however. Taking the PP and PM methods and combining them, the Particle-particle-particle-mesh method (P^3M) is created (Efstathiou & Eastwood, 1981). This is considerably more efficient than either PP or PM as it combines the best qualities of both. Still the number of interactions to be computed when there is significant clustering of particles still scales as N^2 so computational cost is high. A solution to this problem was devised by Couchman (1991) with the introduction of adaptive meshing where subgrids are allowed to form in denser regions speeding up the direct summation of nearby neighbours. This method is known as Adaptive particle-particle-particle-mesh method (AdP^3M). AdP^3M is sometimes (erroneously) thought of as a meshless method due to it being particle method, but in fact it still maps particles onto a grid (figure 3.1).

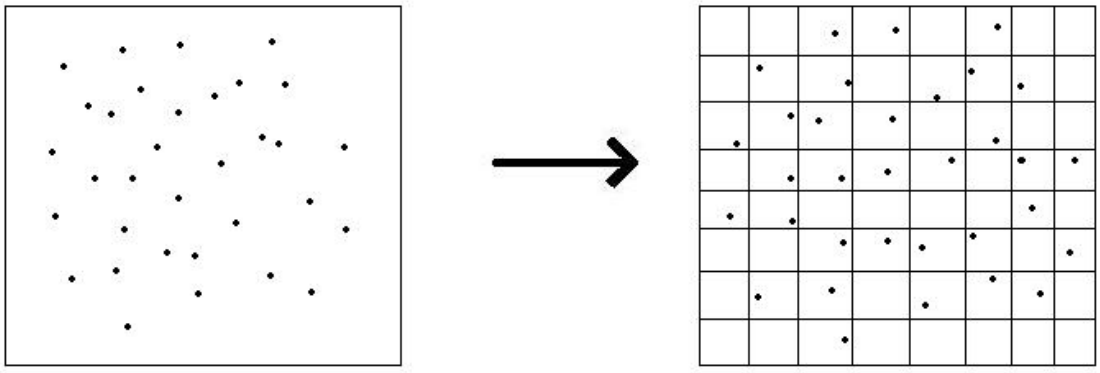


Figure 3.1. Mapping of particle positions onto a mesh.

However it remains different to a pure grid based code. An Eulerian grid code would not have the mapping step described above; instead it just has cells with values of properties defined for each cell. If density values were to be considered in this (simplified) example there would be fluid flow over time as the values are changing with time (figure 3.2).

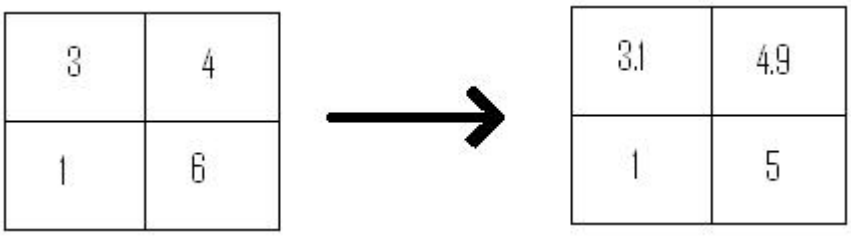


Figure 3.2. Information exchange between grid cells at timestep and timestep 2 simulating flow over time.

This means it must exchange information between cells – and solve the continuity equation to ensure mass conservation. A particle method has no need to do this (as long as no particles are lost from the simulation).

It should be noted though that in a particle method such as AdP^{3}M when particles are mapped onto the grid:

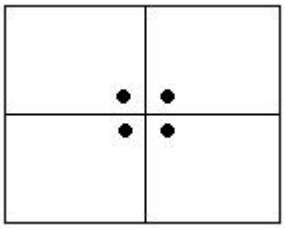


Figure 3.3. Particle mapping inaccuracy.

Is considered identical to

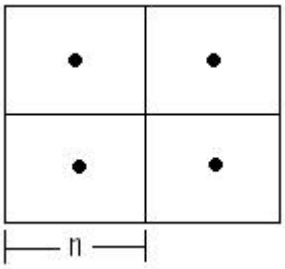


Figure 3.4. Particle mapping inaccuracy (cont.).

even though there is a subtle difference in the real world. The size of the cell is equal to the Nyquist length (n) and could be considered analogous to an accuracy or resolution limit. When this limit is breached errors can be generated in the solution due to small scale information loss.

In the same vein that Gadget2 is a combined Tree-SPH code, Hydra is an AdP^3M combined with SPH (AdP^3M-SPH) code (Couchman, Thomas, & Pearce, 1995). SPH is a completely meshless method but AdP^3M is not so the implication of this is that Hydra may be considered technically not to be completely meshless – though it is still of course a Lagrangian particle code not a grid code. Large clouds of gas, dark matter, stellar clusters and galaxies are all vast objects in space and have their own gravitational fields but on Earth gravity is a simple and constant force downwards. The astronomical version of Hydra contains modules for calculating the self gravity of the particles (Counchman, 1991). These routines are not necessary for the water code so are not implemented for the main research of this thesis.

Hydra uses a fairly conventional implementation of SPH and as is common nowadays uses a process called linked lists in order to speed up neighbour searches. All SPH particles are affected by the ones around them and so when a code is working out what forces a particle is experiencing it must find all of its neighbours within the smoothing length (Hydra tries to search for the nearest 32). This is a computationally expensive part of the simulation so instead of checking every particle to see if it is within range it only considers those close by. The job of the linked list subroutine is to map the particle positions onto a grid in order to eliminate the ones far away. Essentially:

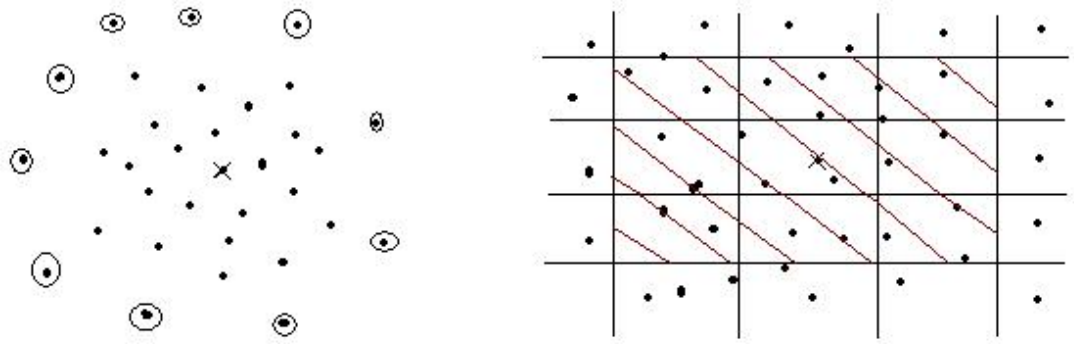


Figure 3.5. Example of neighbourhood searches in standard form and with linked lists respectively.

All of the circled particles (of figure 3.5) are too far away to have an effect on the particle in question (the cross) but are calculated anyway wasting resources. With linked lists employed (right of figure 3.5) only the particles own cell and the eight surrounding cells (shaded) are considered to contain potential neighbours as these are the ones which have the greatest effect on the particle and probably contain sufficient numbers anyway. This results in a huge computational saving reducing the number of operations from N^2 to N log N (though technically introduces a grid back into Hydra).

Hydra does have the significant disadvantage that even though massively parallel versions exist (Pearce & Couchman, 1997), the publicly available version used in this research is not a parallel implementation and so this code cannot, as released, be used for very large

simulations. Akin to Gadget2, the original Hydra uses an entropy conserving implementation of SPH (entropy is recast as the state variable instead of density), but as temperature is not used in the water version of Hydra this is no longer the case and so this property is not considered in this thesis. Unlike Gadget2, Hydra does not have fully adaptive individual timesteps. Although the timestep adapts automatically from one step to the next (i.e. they are completely variable) all the particles move in lockstep. The size of timestep is tied to the maximum acceleration experienced by any particle as shown in equations 38 and 39.

$$\Delta t \sim \left(\frac{\Delta s}{a_{\max}} \right)^{\frac{1}{2}} \tag{38}$$

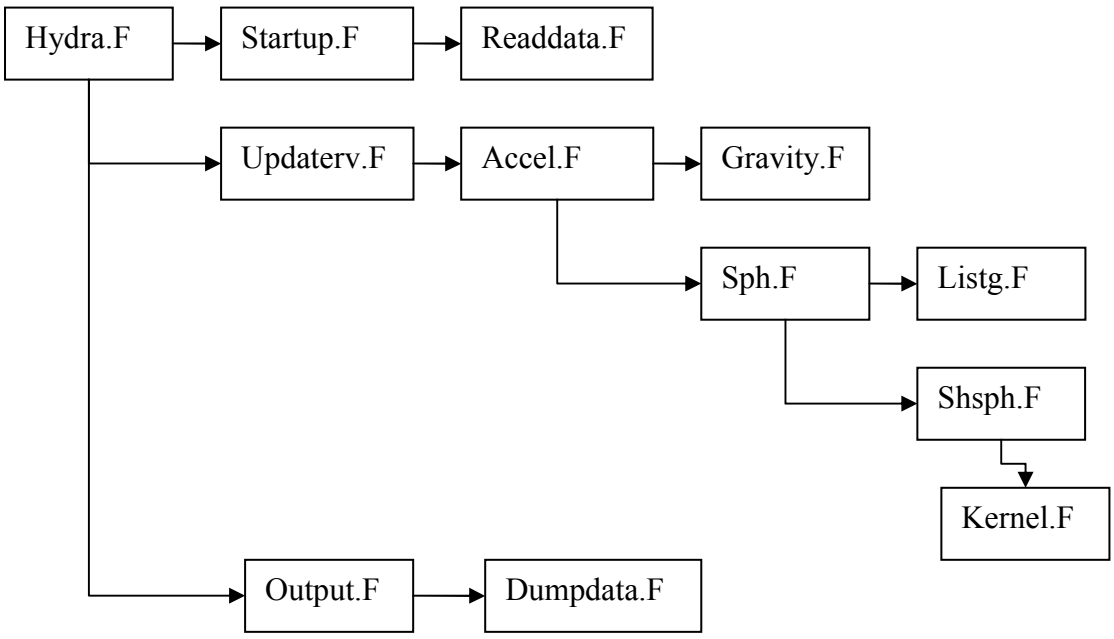
$$\Delta t \sim \frac{\Delta s}{v_{\max}} \tag{39}$$

where Δt is the timestep, a is the acceleration, v is the velocity and s is the distance.

It is important that the timestep be properly chosen. If it is too small then the simulation will be too costly and take too much time to run. If it is too large then the risk is run that a particle will "jump" through a boundary between one step and the next. Short timesteps mean that a particle cannot travel too far even if accelerations they experience are large and so boundary forces have a chance to act. A compromise must be reached here which is done automatically by Hydra. It should also be noted that Hydra in its water version fixes the smoothing length at the start of a simulation due to the fact that water has a constant density and it is not required to expand and contract every timestep. This results in a computational saving.

3.1.2 Detailed subroutine layout of Hydra

Figure 3.6 is a look at the code structure of Hydra with the subroutines laid out in the order they are called by the solver. Hydra is written entirely in FORTRAN and is initiated by the Hydra.F subroutine. The other subroutines are repeatedly called according to the diagram until the simulation time exceeds the highest required output time. For diagram clarity a few very minor subroutines are not included here.



Includes

Psize.inc
Rvarrays.inc
Pinfo.inc
Itype.inc
Units.inc

Makeflags

Figure 3.6. Layout of Hydra.

In figure 3.6 the main subroutines are in boxes whilst the extra files not part of the calculation loop but still vital to simulation as they contain key values are in the top right corner. An arrow indicated that this subroutine is called by the one above it.

The basic function of each of the subroutines is laid out in figure 3.7:

Subroutine	***Type***	***Function***
Hydra	Fortran	Initiates the simulation and calls all the other subroutines in the correct order.
startup	Fortran	Provides the initial conditions (IC's) of the simulation. Also checks simulation parameters and initialises output times.
updaterv	Fortran	Calls accel.F and keeps track of particle energies.
output	Fortran	Creates the data files and updates the log.
dumpdata	Fortran	Saves results files for later analysis.
readdata	Fortran	Reads in any data file required, e.g. particle positions for IC's.
accel	Fortran	Evaluates the accelerations and controls the timestep.
infout	Fortran	Writes the energy log file.
gravity	Fortran	Describes the gravitational force.
sph	Fortran	Decides when to apply sph forces.
shsph	Fortran	Calculates the sph forces for all the particles.
listg	Fortran	Creates the linked list.
kernel	Fortran	Describes all the kernels which are available.
makeflags	Control	Tells the compiler which Fortran files to include.
psize	Include	Contains the essential parameters of the simulation.
rvarrays	Include	Data arrays containing the physical parameters of the system.
pinfo	Include	Lists commonly used parameters.
itype	Include	Contains a list of particle types and their reference numbers.
units	Include	Provides the units relative to the simulation.

Figure 3.7. List of Hydra subroutines and their function.

3.1.3 Previous use of Hydra

Hydra has been exclusively used in the field of astrophysics research for several years and has been successfully used in many published works. A range of these are listed below although this should not be considered an exhaustive list.

Simulation code tests and comparisons (Tasker et al, 2008; Agertz et al, 2007; Kay et al, 2000), Galactic structure, Galaxy and star formation (Pearce et al, 2001; Kay et al, 2002; Gazzola et al, 2007; Evrard et al, 2002) and Effects of Cooling and preheating on X-ray properties (Muanwong et al, 2002).

3.2 Riemann Shock Tube

3.2.1 Sod Shock

Significant time has been devoted to studying the well known Riemann Shock Tube problem (often described simply as a Sod Shock) (Sod, 1978) which has proven to be an important test for a computational simulation code. Here the Sod shock will be modelled using the SPH method implemented in Hydra. The Sod Shock is a very powerful test of a computational code because it allows the user to test three difficult to model principles simultaneously. These being the shock wave, the sound wave and the contact discontinuity which resides in between them. Studying how various physical properties change over these regions of interest (ROI's) and how SPH handles the modelling of them is the main aim of this section. This may seem far removed from the floodwater Hydraulics modelling that is at the heart of this thesis but the principle of this set up can be used in water simulations later on.

A Sod shock is essentially where two regions of gas are placed next to each other within a box or tube. One of these regions is hot and dense while the other is cold and diffuse and there is a thin impermeable membrane placed in between them. The regions of gas should be as close to a smooth, uniform state as possible with the particles being unclustered yet without any grid alignment. This state is called a glass. The starting point from which the particles are let go and tracked is marked by the removal of the membrane. In the absence of external stimuli gas will expand to fill any volume with uniform density. As soon as the gases are released the hot, dense region starts propagating into the cold, diffuse one. A shock wave will propagate into the diffuse area while simultaneously a sound wave will propagate into the dense region. The particles in either region will not begin moving until the respective wave reaches them and delivers the information that there is a density imbalance. Once a shock has been simulated the data concerning the density, temperature, entropy, pressure, velocity and shock front can be analysed and plotted graphically.

To begin with, a cubic grid of particles was initialised and each particle was superimposed with a small random displacement. This was accomplished by incorporating a random number generator telling the particles how much to move off from their set locations in the grid. This cube was then left to run for around 10,000 steps in Hydra just by itself in order for it to settle down into the smooth glass required. This process was then repeated for another cube of the same size but with four times fewer particles inside it (i.e. four times less dense). After two glasses had been constructed the left hand side of one was cut out and the right hand side of the other was cut out and they were placed together. This is a good example of one simulation providing the initial conditions for the start of another. The shock simulation was then begun.

Performed first was a standard Sod shock with a 1x1x1 cube containing approximately 250,000 particles made from two glasses (one containing approximately 400,000 particles, the other approximately 100,000). The dense and diffuse sides each take up exactly half the cube, i.e. the discontinuity is a 2D sheet beginning at x=0.5. The dense side is 4 times denser

than the diffuse side and is 1.4 times hotter at the start. The ratio of pressures is 1:0.1795. The shock runs parallel to the x-axis and wraps in every direction meaning the simulation is completely periodic with no boundaries present. Therefore there is a second shock coming in the opposite direction but the simulation was stopped before the 2 shocks are allowed to interact. The initial set up is displayed in figure 3.8 with black being dense and white diffuse:

Figure 3.8. Initial conditions of the standard Sod shock.

Figure 3.9 is a sequence of pictures which depict the shock described above. The pictures are taken along the z-axis (perpendicular to the shock) and show density in a logarithmic scale where yellow is most dense and blue the most diffuse. The sequence is a series of snapshots in time (from left to right) showing how the shock propagates quickly throughout the cube and the rarefaction wave travels in the opposite direction. The first picture is just after the simulation has begun so the shock has had only a fraction of a second to travel. Nevertheless it can still be seen as a red line. By the end the second shock coming in from the left hand edge (due to the simulations periodicity) can be seen to have almost caught up with the main sound wave travelling from the centre towards the left.

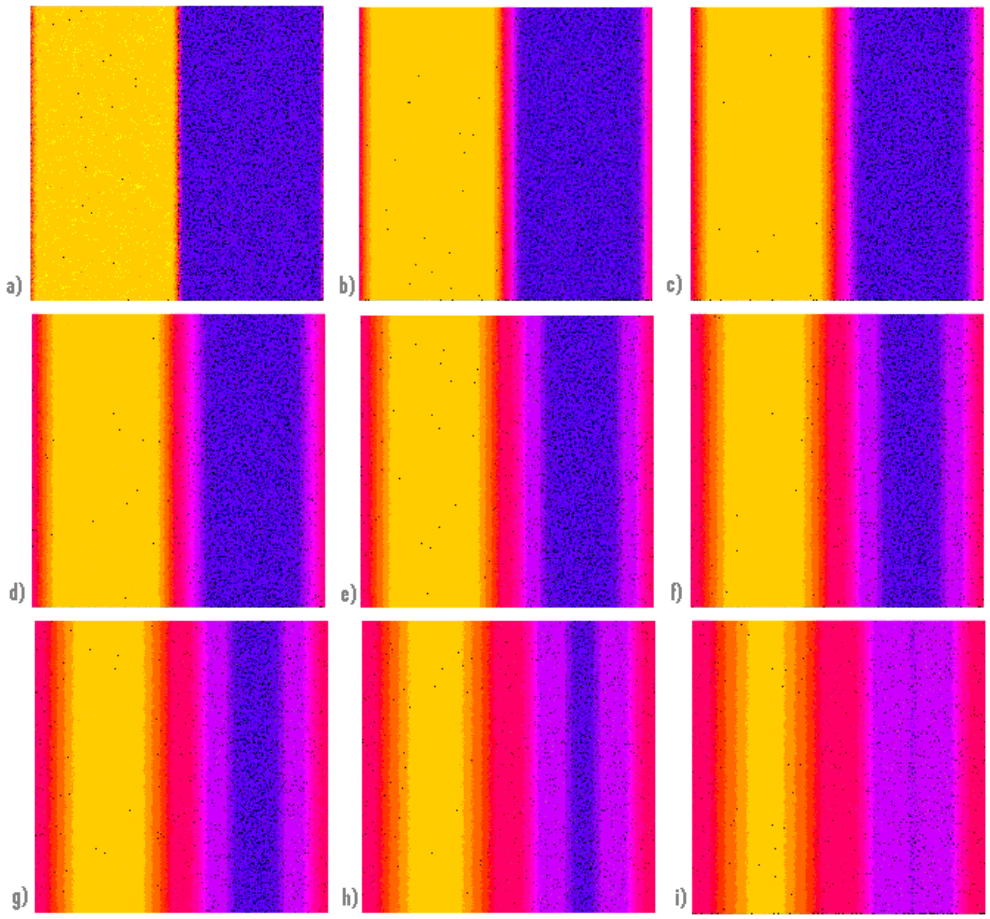
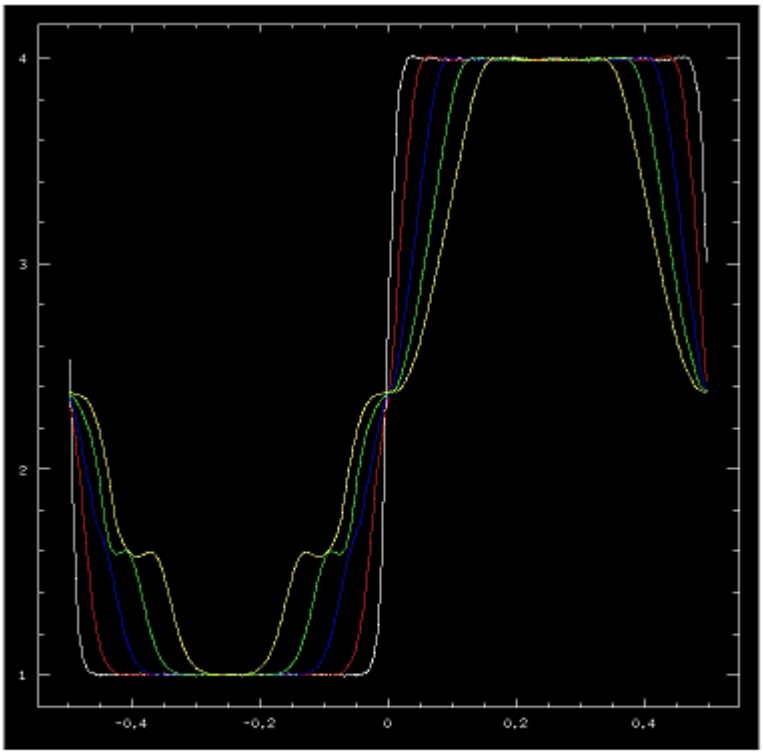


Figure 3.9. Density map sequence (yellow is 4 times more dense than blue going via red) showing the propagation of the straight shock (going into the blue) through the whole simulation. Times are a) = 0.005s, b) = 0.03s, c) = 0.06s, d) = 0.09s, e) = 0.12s, f) = 0.15s, g) = 0.18s, h) = 0.21s, i) = 0.24s.

The visualisations of the simulation indicate that it has been successful but proper analysis of the physical properties is required. A technique called “binning”, where thin slices through the simulation are sampled, was used to collect the information required from a Hydra output file. This allows physical properties to be plotted in the form of a graph. Figure 3.10 is a plot showing how the density of the simulation changes with time.



time = 0 (white) time = 0.05 (red)
time = 0.1 (blue) time = 0.15 (green)
time = 0.2 (yellow)

Figure 3.10. Density profiles taken perpendicular to the straight shock showing how the density around the shock front, rarefaction wave and discontinuity change with time.

The dense side is on the right while the diffuse side (4 times lower) is on the left. The region separating them is the region of interest. The second shock will shortly meet the centre one after time t=0.2 (the approaching second shock causes curving at each side). The region of interest is divided into three parts, the shock front (lower portion), the sound wave (upper portion) and the contact discontinuity in the middle which represents the boundary between the two sets of particles. Note how the gradient of the shock gets lower with time as the rarefaction wave (sound) broadens.

The time picked to analyse the Sod Shock was 0.15 seconds after the start of the simulation. This was chosen because it represented a time when the shock had covered most of the way across the cube but had not yet interacted with the second shock coming in the other

direction. Using the density, temperature and velocity information, pressure and entropy can also be worked out. Figure 3.11 shows are the plotted profiles of the Sod Shock.

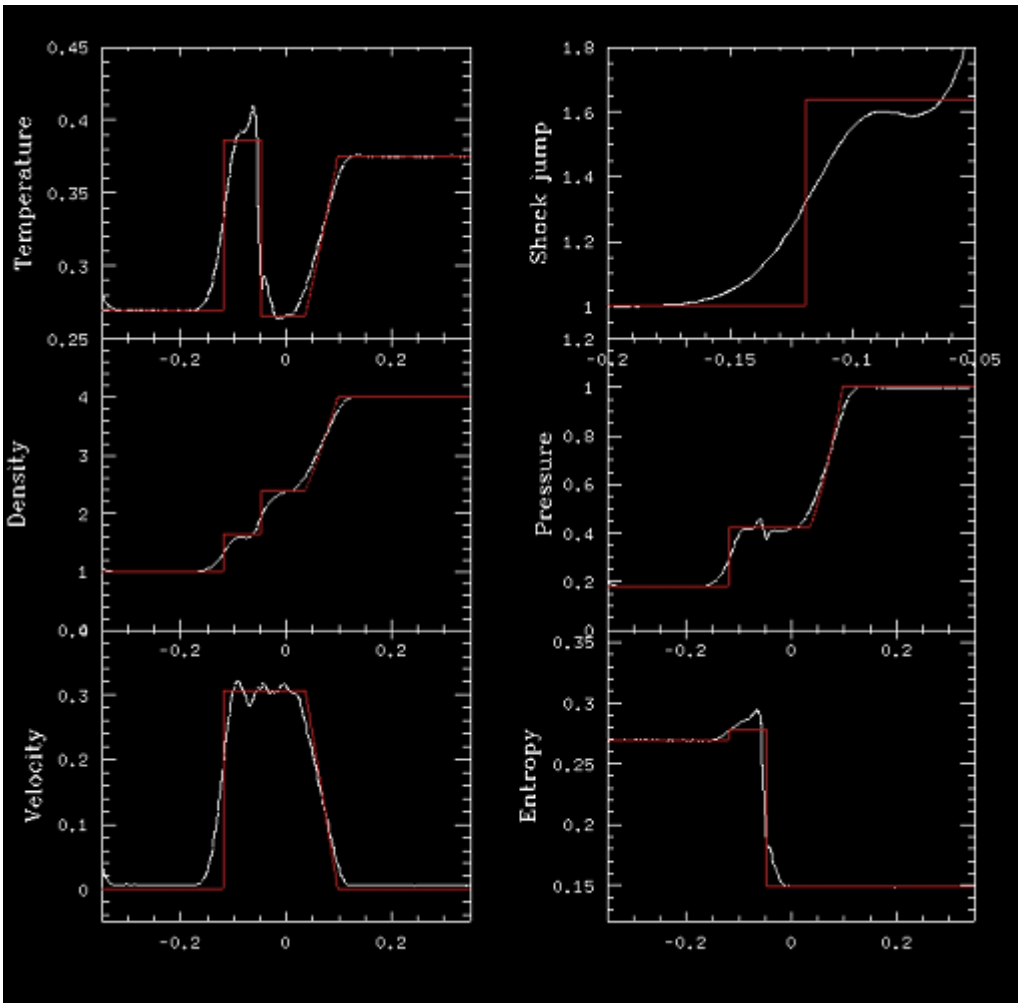


Figure 3.11. Profiles taken perpendicular to the straight shock at time t=0.15. Units should be considered to be unitary and relative to each other, i.e. density on 1 side of the shock is set 4 times higher than the other at the start and pressure on the dense side is simply set to 1.

The profile shows 6 graphs which display the temperature, density, velocity, shock jump, pressure and entropy as they change around the shock area. The graphs are all in line with each other with the exception of the shock jump. Looking at the density profile (middle left) the shock, contact discontinuity and rarefaction wave can clearly be seen. The shock jump graph represents a blow up of the shock area of the density graph. The simulation results are the smooth white lines. They have been overlaid with the analytical result (i.e. a perfect

shock) in red. The mathematical solution of the analytical result was first calculated by Sod (1978) and that solution is employed here for comparsion. The closer the match the simulation has to the analytic result the better. Notice when the shock approaches how the Hydra results curve slightly into the jump instead of being sharp right angles like in the analytical model. This is a good example of the smoothed nature of the SPH method. The particles which have just been hit by the shockwave are showing a density much higher than the ones before them; but, because of smoothing, they can still influence the properties of their neighbouring particles. The edge of the shockwave artificially increases the density of the "soon to be shocked" particles by a fraction of the density inside the shocked section creating a smooth gradient instead of a sudden shock. The sudden appearance of a density jump introduces a local source of entropy into the simulation and causes the code some difficulty in fully resolving the contact discontinuity. This effect is manifested in the spikes which can be seen at the location of the discontinuity. These results are in agreement with previous work done by Springel (2005). In order to highlight the improvement due to higher particle numbers in such a simulation the whole process was repeated but using four times fewer particles, i.e. each glass had four times fewer making the overall shock four times less. Analysis was done in an identical manner to the 250,000 particle shock and produced the following profiles in figure 3.12:

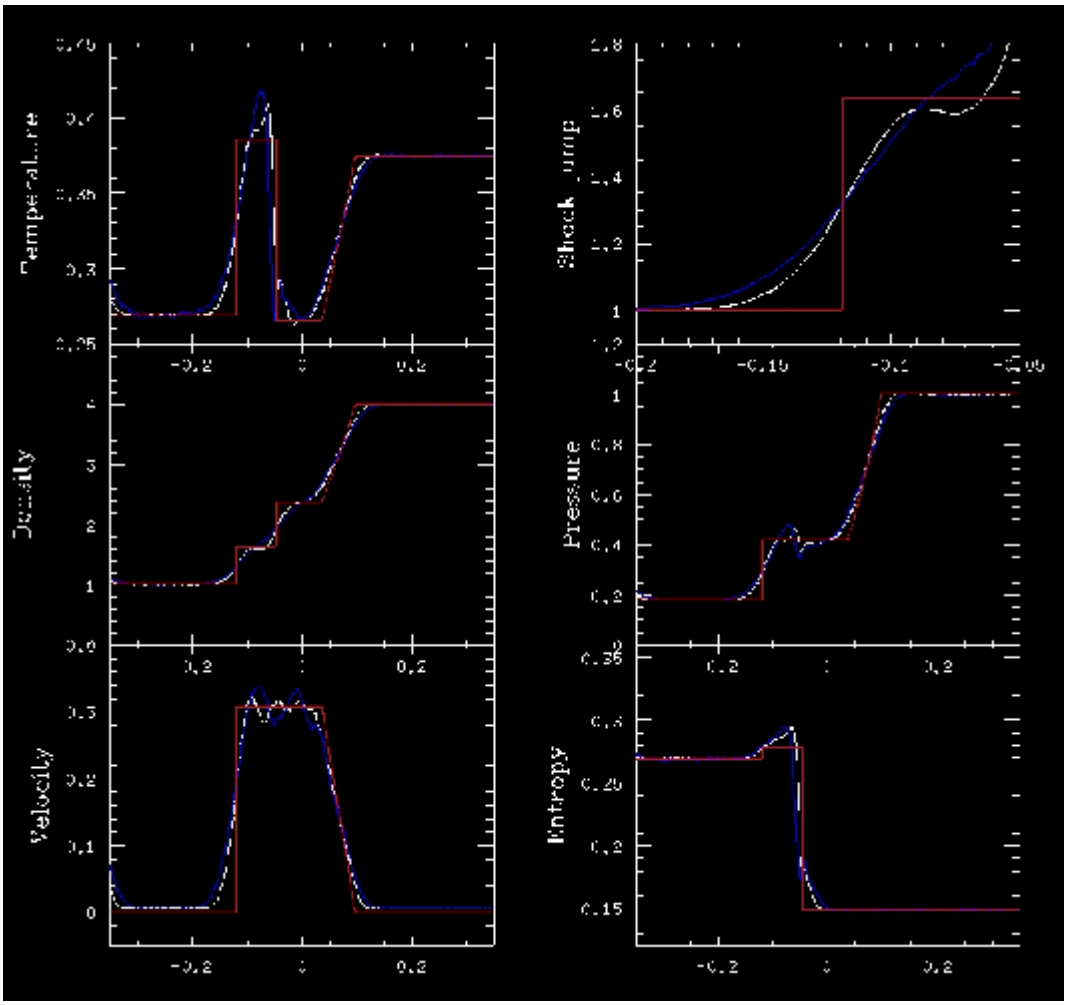


Figure 3.12. Lower resolution profiles of the straight shock at time t=0.15.

The profile in figure is the same as the original but with the lower resolution profile overlaid in blue for easy comparison. The features can be seen to be the same but with a less close match to the higher resolution plots (e.g. shock jump is less defined). In the temperature, pressure and entropy plots a small spike is visible around the contact discontinuity and this problem seems to get worse with decreasing resolution. Noise is also larger in the lower resolution graphs, best seen by the wavy lines in the velocity graph. This means that a simulation with a higher number of particles than the 250,000 should achieve an even closer match to the analytic result. Some of these results and profiles of a shock with 4 times more particles than the original one (1 million) can be viewed in Thacker et al (2008). There is also a comparison with a wide range of alternative numerical techniques. Hydra and Gadget2 are

both SPH codes, whilst Enzo and Flash are both Eulerian AMR codes. The density and shock jump profiles can be seen in figure 3.13.

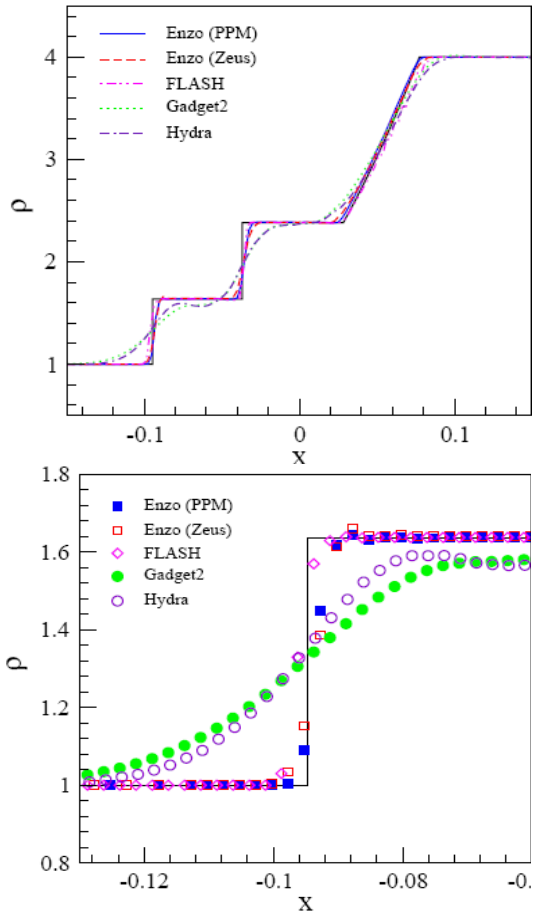


Figure 3.13. Density profile and shock jump comparison from 5 different codes (Thacker et al, 2008).

The estimation of density can be seen to be accurate in all codes but the smoothing effects of the SPH method around steep density gradients can clearly be seen versus the AMR grid codes in the shock jump profile. The solution accuracy in mesh based methods is also dependent on cell number as can be seen when Enzo is run as a static grid with various different cell numbers and compared to Hydra and the analytical solution (figure 3.14).

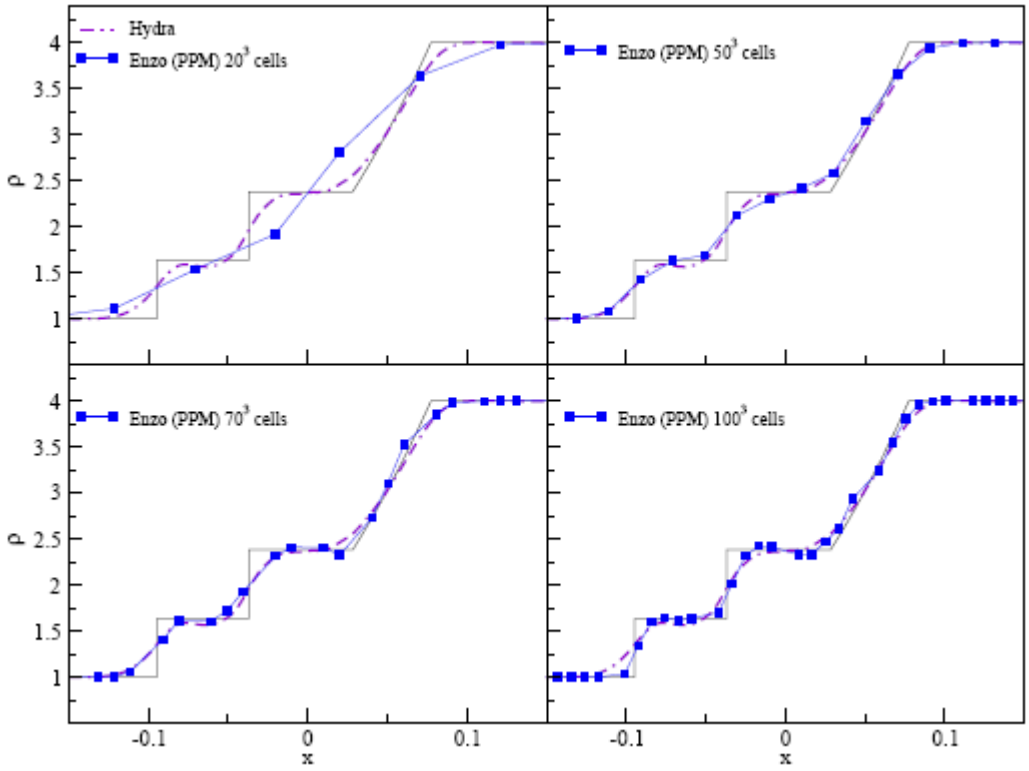


Figure 3.14. Density profiles demonstrating how increasing cell count improves accuracy across all three ROI's (Thacker et al, 2008).

3.2.2 Oblique Shock

At times it may become necessary to simulate a process when the orientation of the simulation is not at an easy angle; real events in nature rarely occur like ideal test cases. In the previous examples the shocks occurred exactly aligned to the grid axes. It is important that the code be tested at other angles so that it is known whether or not it can handle non aligned simulations. To test this, another shock with exactly the same initial conditions as the standard shock (250,000 particles) but with an important difference was simulated. The shock happens at a 45 degree angle to the x-y plane. This was created from two smooth glasses just as in the "straight" shock but the parameters for selection during the cutting out of particles phase were changed. It is important to remember that the box of each glass is a cube of side length one, meaning all the particles had an x, y and z coordinate between zero and

one. For the straight shock every particle which had x-coordinate less than 0.5 in one glass and every particle which had x-coordinate greater than 0.5 in the other was selected. In the case of the oblique shock, every particle which had the sum of its x and y coordinates less than 1 in one glass and every particle which had the sum of its x and y coordinates greater than 1 in the other was selected. There is wrapping in every direction in this simulation as well meaning that stripes of high and low density all at 45 degrees to the perpendicular occur. This will be described in this research as an "oblique Sod Shock" and takes the following shape.

Figure 3.15. Initial conditions of the oblique Sod shock.

This is an important test because it can be used to show that the orientation of the shock is irrelevant when using SPH. Some mesh based codes can have problems when dealing with interactions that are not happening in the same orientation as the grid giving SPH an advantage over them. All pictures are still taken along the z-axis. Figure 3.16 is a snapshot taken from the oblique shock to show how the simulation looked as it progressed with yellow being high density and blue low density:

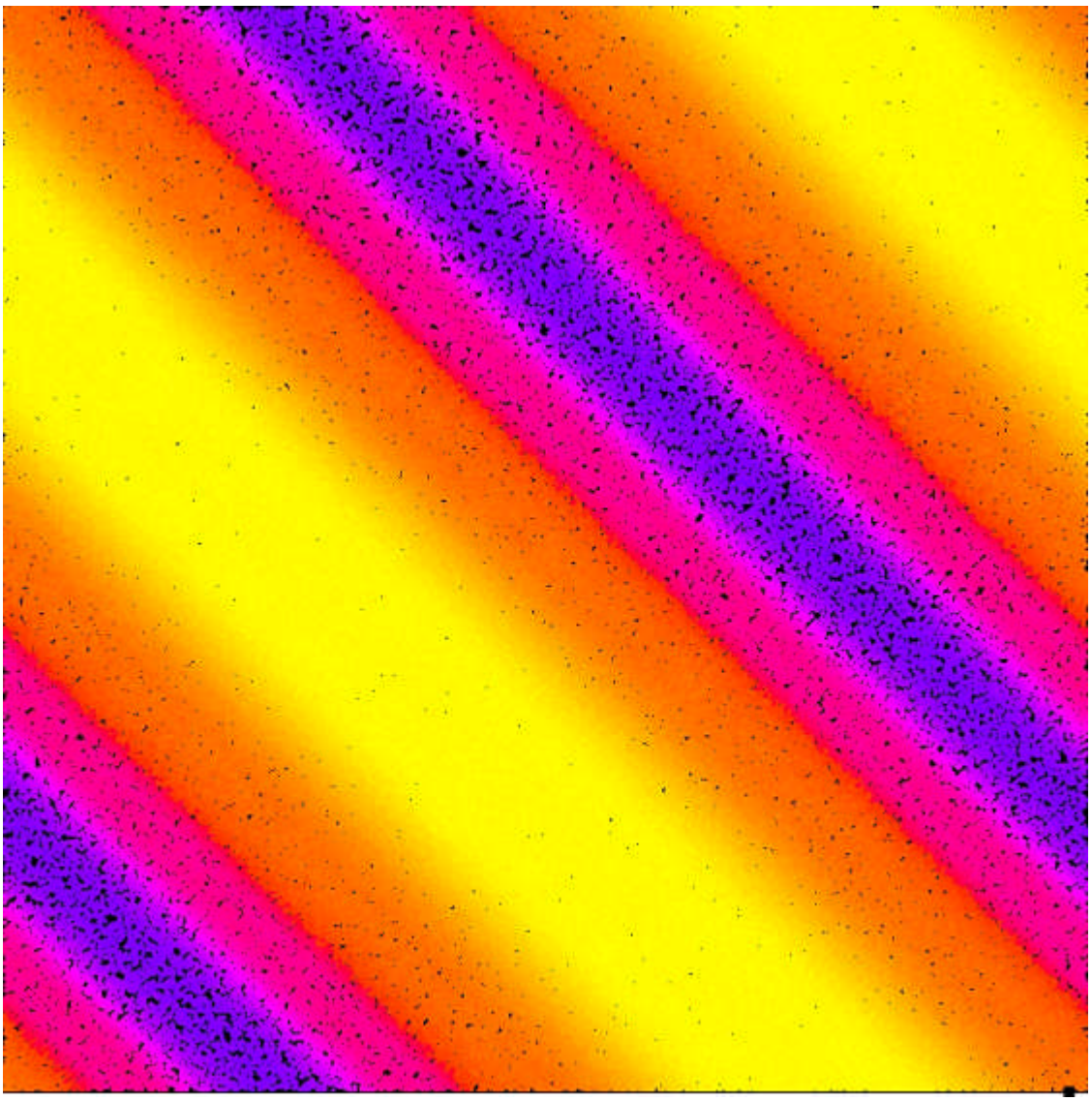


Figure 3.16. Density map of straight Sod shock at time t=0.15.

As before the profiles of the physical properties of the shock as it propagated were analysed. Performing the analysis identically to the original “straight” Sod Shock the two orientations can be compared directly by overlaying one profile onto the other as displayed in figure 3.17.

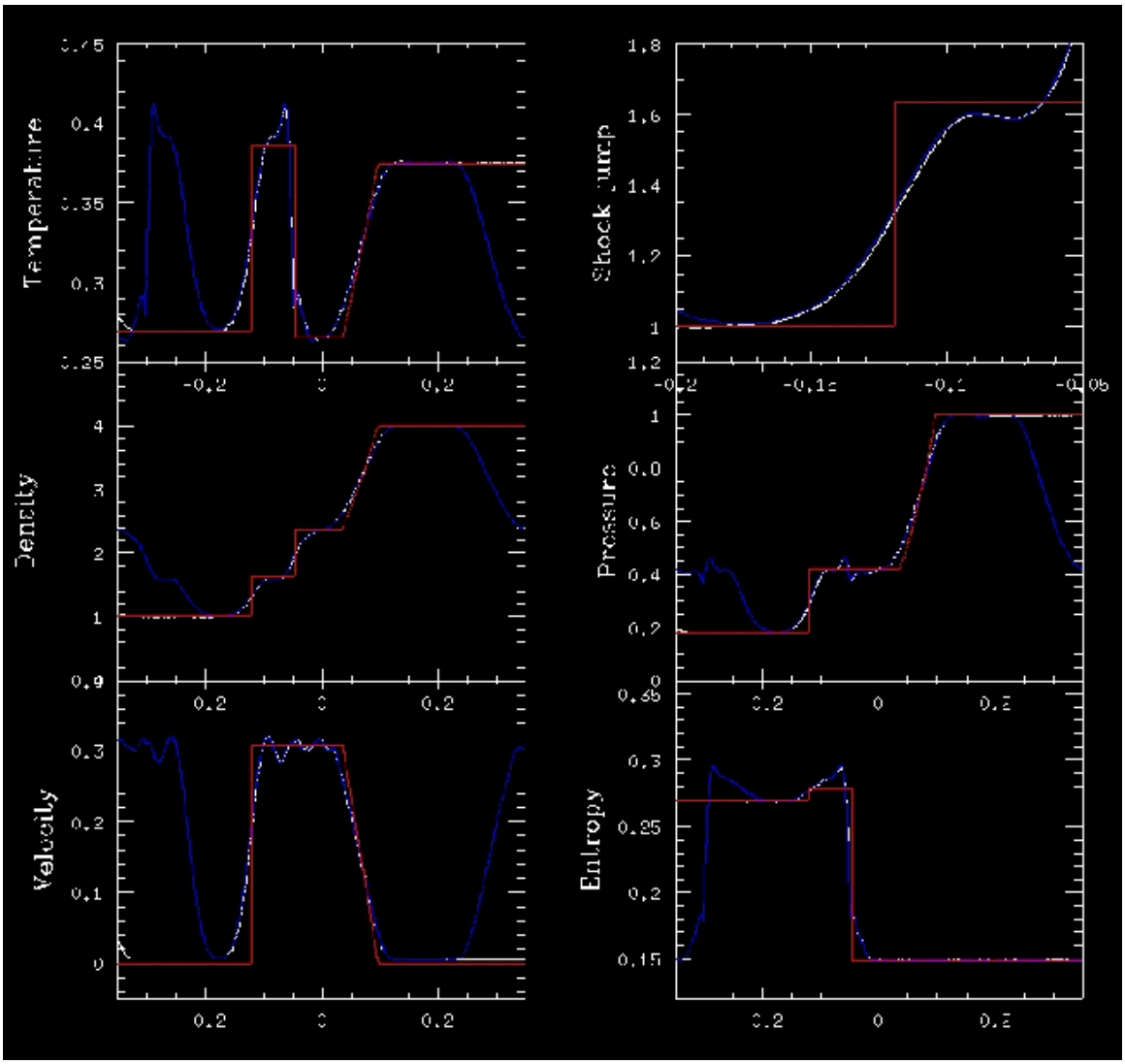


Figure 3.17. Comparison of the straight and oblique shock profiles at t=0.15s. Units should be considered to be unitary and relative to each other, i.e. density on 1 side of the shock is set 4 times higher than the other at the start and pressure on the dense side is simply set to 1.

The straight shock is in white, the oblique shock is in blue and the analytic result is in red. For the whole shock portion the straight shock is almost indistinguishable from the oblique. This indicates that the code is functioning completely independent of the orientation of the grid. The oblique plot curves round for the second shock earlier than the straight because the distance between them at the start is shorter so the second shock approaches the centre one sooner.

Now that a basic gas dynamics code test has been completed successfully using the astrophysics version of Hydra, the conversion of Hydra to model liquid based problems and eventually flood inundation can begin.

3.3 Code Modification to Water

3.3.1 Fluid Properties

In order to modify Hydra as it stands into Hydra (water version) the main aspects of the code that need to be altered are the equation of state, the particle types that are used and the inclusion of physical boundaries. The creation of solid impenetrable boundaries is perhaps the biggest and most complicated challenge of the modification process.

The equation of state used by the fluid was modified according to the equation 22 described in Section 2.3.3. With gamma set to 7 a very stiff fluid was created which closely approximates the density of water and can support itself under gravity. The coefficient B described in equation 23 will alter depending on the chosen speed of sound.

As standard a particle in Hydra can be representative of gas, stars or dark matter. Each particle is assigned to one of these types. The code looks at which type a particle is and applies the properties accordingly. The gas type was kept and used with the new EOS to make the fluid particles. These are the particles which move and represent the water. The dark matter and star types were deleted and replaced with new ones designed for the water version of Hydra.

3.3.2 Boundaries

There were no solid boundaries needed in astrophysics simulations but as they are essential to current research, new particle types had to be created to simulate them. This was one of the most important parts of the code conversion as without correct boundary conditions no simulation could be relied on.

Taking a similar approach to Liu and Liu (2003), Gomez-Gesteira and Dalrymple (2004) and Violeau & Issa (2007), boundary particles have been implemented within Hydra. These act as a repulsive wall with each boundary particle acting to repel fluid particles that closely approach. Essentially they may be considered as point repulsion particles and as such do not contribute to the SPH density calculations. These are called type bound. Improving upon the standard implementation of such particles the repulsive force was only applied in a fixed direction normal to the wall itself (e.g. upwards if the desired boundary is a base layer). This has the advantage of significantly reducing boundary layer friction. There are two different types of "bound" particle these being boundx and boundy which form solid walls horizontally and vertically respectively. A third type, Boundz particles, has also been coded for the implementation of solid walls in a third dimension if needed. The boundary particles used by Hydra have the advantage that they can be simulated in the same program loops as the fluid particles. This results in a reduction in computational cost. The locations of boundary particles are set in the initial conditions but with double the number per unit length compared to fluid particles to prevent fluid particles penetrating the wall. Having double the number of boundary particles to fluid particles also creates a fairly flat boundary pressure zone. The force that boundary particles apply away from themselves rises inversely to distance as a power law so as a fluid particle closes on a boundary particle the force increases considerably. For the simulations performed in this research a forth power law is used. The force applied by a boundary particle is cut off at a short distance away from the particle. This is designed to eliminate the effect of boundaries on particles which are no where near them. The distance chosen should be of the order of, or slightly less than the inter-particle

separation. A particle interacting with a boundary particle under gravity will bounce a few times due to the pressure variations. As was noted in section 3.1 these boundary particles do not guarantee that a boundary cannot be penetrated; they can only succeed if the timestep of a simulation is short enough to prevent a particle jumping right through them. This type of boundary condition is comparable to that outlined by Monaghan (1994) and Monaghan and Kos (1999) where an analogy to the Van der Waal's forces between molecules is drawn.

Hydra includes another type of particles which have been given the name ghost particles. These are placed in layers outside of boundary particles, i.e. not in contact with fluid, with fixed positions. The ghost particles are placed with the same number per unit length as fluid particles, i.e. half the number of boundary particles. Ghost particles participate in the SPH density calculation but do not "feel" any forces and do not move. In this respect they are much closer in physics to the fluid particles than the bound particles. They are coupled to the fluid particles via the XSPH coupling despite not ever having their position updated. They are skipped by the subroutine which updates particle position each timestep. Even though they cannot ever actually move they can be imbued with a velocity. This can be used to create desired motion effects within the water. The ghost particles ensure that SPH returns a correct density estimate for particles lying close to a boundary and prevents spurious boundary pressures. Water does not always provide sufficient nearby particles in the neighbour search (i.e. within the smoothing length) when near a boundary. A particle near a boundary can "see" over the boundary and feel the sudden drop in density due to the smoothing length. Ghost particles can ensure a correct (or at least better) density estimate. The depth of the ghost particle layers is determined by the required number density of fluid particles. In Hydra 32 neighbours are searched for within the smoothing length which is circular. Therefore number density = $32 = \pi h^2$ with h being the smoothing length. So the number of ghost particle layers can be represented by:

$$\sqrt{\frac{\#\rho}{\pi}} \tag{40}$$

where #ρ is the number density.

Using this approach at least four layers of ghost particle are required in Hydra in order to provide a proper density estimate for a fluid particle right on the boundary. When this number is present a fluid particle's smoothing length will not reach the edge of the ghost particles. At least this number will be used throughout the research presented in this report.

This approach is different to the application of ghost particles detailed by Randles and Libersky (1996) who use a system of generating a ghost particle with opposite velocity each time a fluid particle approaches a boundary. No attempt to follow the water surface or to generate ghost particles along the free surface is necessary because the surface is stabilised by the presence of a gravitational field here. The different particle types can be seen in figure 3.18, where fluid particles are represented in blue, boundary particles are represented in red and ghost particles are represented in green.

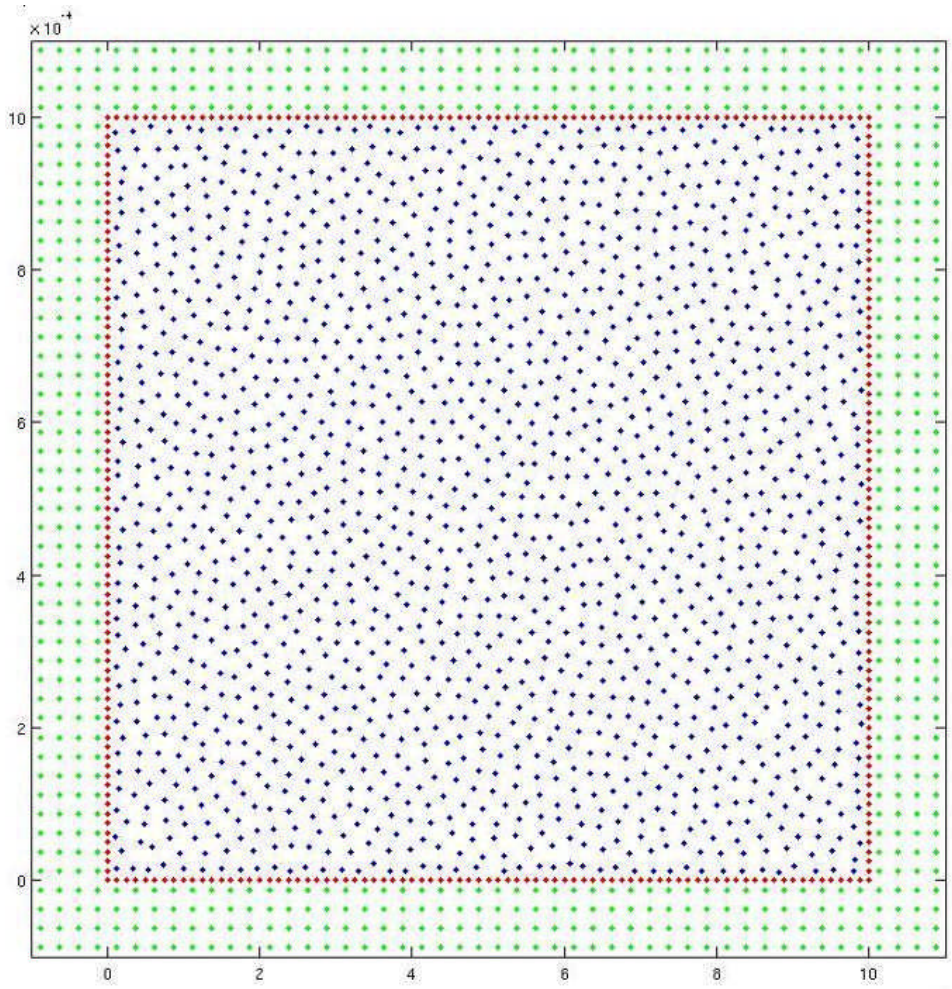


Figure 3.18. An enclosed container filled with water demonstrating fluid particles (blue), boundary particles (red) and ghost particles (green). The fluid has settled into a glass state.

There is no limit to the number of different types of particle that can be defined (except perhaps computer power) and different ones can operate with different physics. This allows the possibility of multiphase flow simulations.

Most researchers use a staggered grid to place fluid particles on when generating initial conditions. That approach is not favoured here as the initial conditions to a simulation because a grid is not a natural state in nature. Instead a grid start is used initially but then the fluid particles are allowed to settle into a more randomised fluid particle layout called here as a glass state. This state is different to a completely random distribution because there is still order in the layout of particles. The density estimate produced from a glass state is good

whereas there is no guarantee of that if, say, a Poisson distribution were used. The particles in the glass are then imported into the main simulation to act as initial conditions.

3.4 Initial Verification and Validation simulations

In this section there are a few basic tests to prove that the code is functioning properly in that it can create and lay down particles of water in different shapes and dimensions, define boundaries and complete simulations satisfactorily. This is mainly a verification process for the Hydra code (water version) but can be considered part validation as well. Verification is the process of confirming that the code has been converted successfully to simulate water and is behaving correctly. Validation is the process of comparison against measurable results as a check for solution accuracy. Validation tests will be performed in much more detail later in this thesis. Together these tests will form the initial confirmation that Hydra has been transformed from a gas based astrophysics simulation code to a fluid dynamics code designed to simulate water flow in an Earth environment.

3.4.1 Lid Driven Shear Cavity

Beginning with a simple common code test known as a 2D lid driven shear cavity where an enclosed box is completely filled with fluid. The lid of the box (which is assumed to be infinitely long) is kept at a constant speed moving from left to right. The motion of the lid causes a drag effect within the water. This slowly causes effect within the entire box and eventually a steady state recirculation pattern emerges.

1600 fluid particles were placed inside a square box of side 1mm made of 320 boundary particles. The easiest way of creating the effect of constant lid motion was found not to have

the boundary particles actually move themselves but impart a velocity factor of 1.5m/s in the x-direction on them creating a fluid with a Reynolds number of 100. This way no changes to the code would be necessary, as boundary particles are not updated each timestep (preventing their movement) but they still have x,y,z velocity components set within the startup.F subroutine. The fluid particles would feel the forces on them as if they were moving but no position change would be required. As the box was entirely filled and the test was designed to check stability and the effectiveness of the boundaries no other external forces were needed. Gravity therefore was turned off for this simulation. This was attempted in 3D as well as 2D in order to test the code worked in 3 dimensions (figures 3.19, 3.20, 3.21).

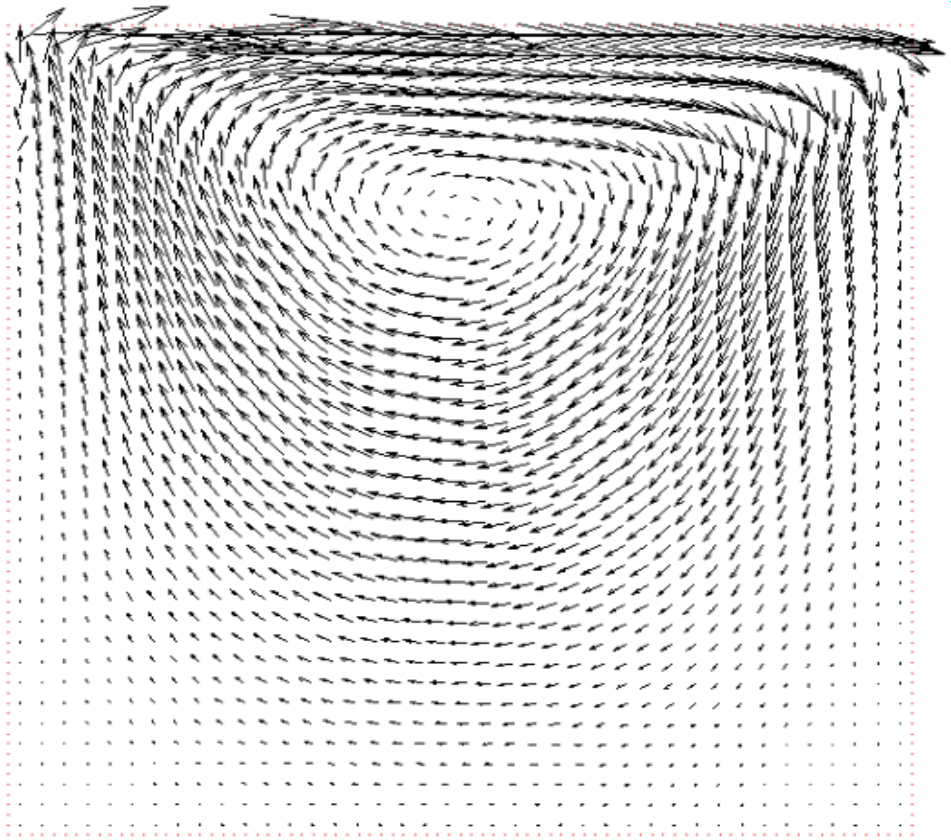


Figure 3.19. 2D lid driven shear cavity showing stable water circulation.

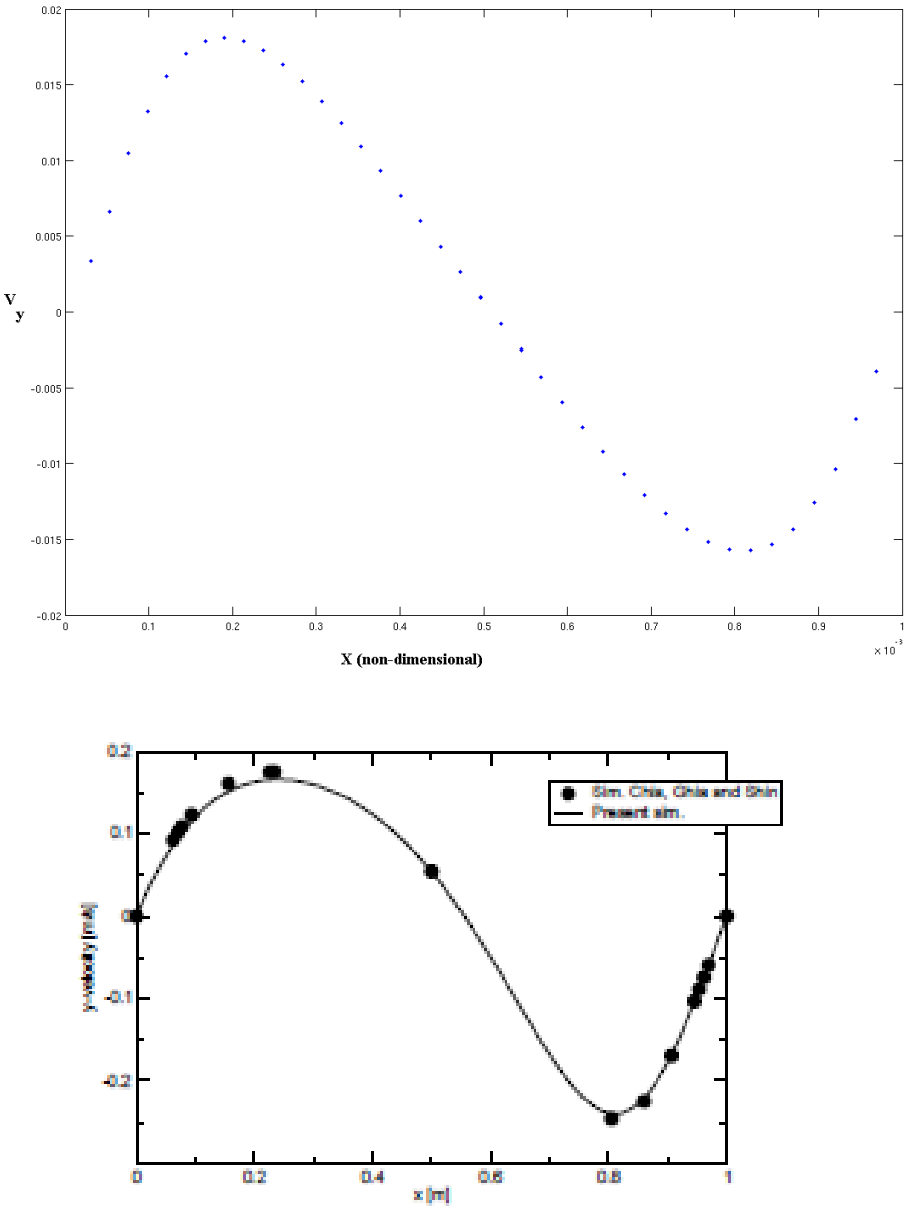


Figure 3.20. X vs Y velocity for 2D shear cavity and a comparison with published data. Hydra (above) agrees well with results from Ghia, Ghia and Shin (1982) (below, dots) and Dauptain (2007) (below, solid line).

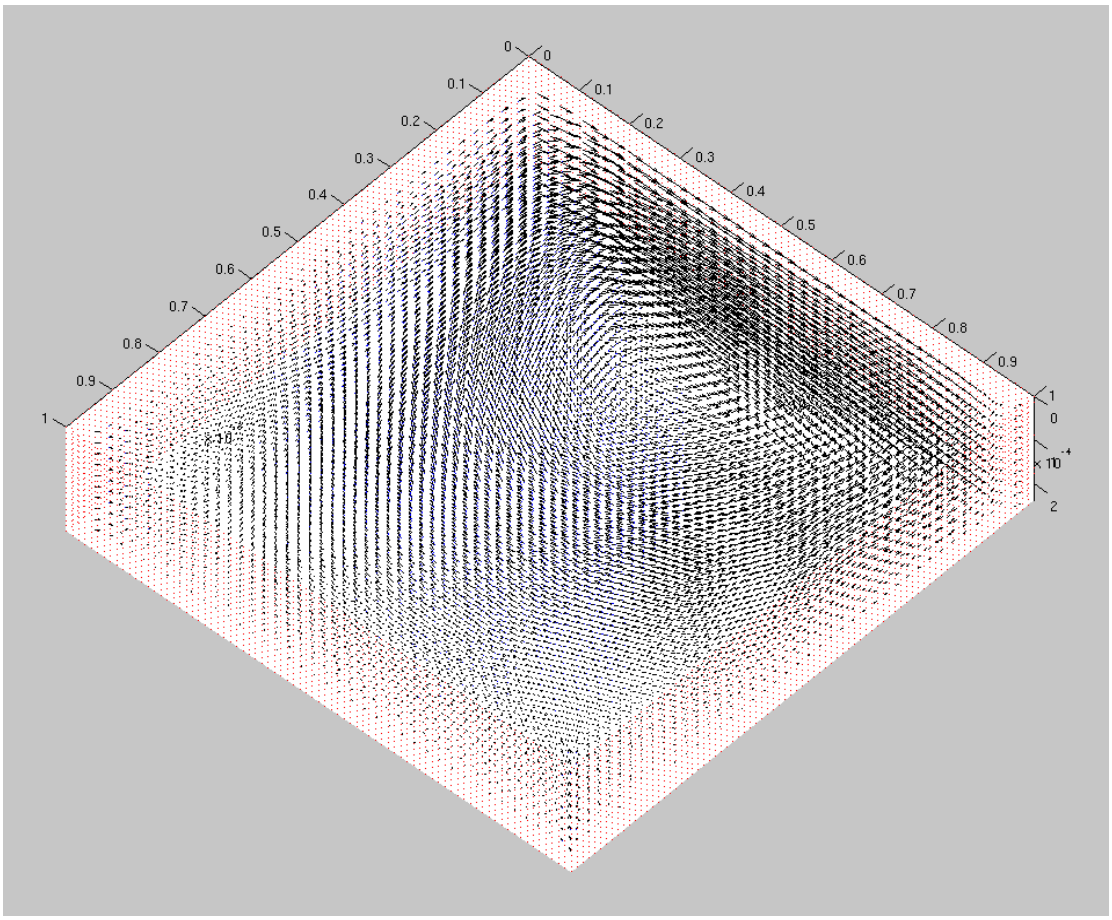


Figure 3.21. Slice through the 3D lid driven shear cavity showing stable circulation.

The water is stable and shows no sign of irregular motion or blowing up in 2D and in 3D. There is slight particle clustering in the top right hand corner but this does not lead to any wall penetration. Overall results are consistent with results from Ghia et al (1982) and Wright and Gaskell (1994).

3.4.2 No Gravity Water Square

This test involves creating a domain in Hydra and placing 2500 water particles inside. There are no walls, containers or boundaries of any kind and the domain is periodic. The particles are initially placed in a square shape (20cm by 20cm) on a regular lattice and left to evolve.

There is no gravity whatsoever in this simulation. It is shown in figure 3.22. The aim of this test is to provide a basic confirmation of the stability of the simulated water without the added complication of boundaries or external forces. This will provide a stepping stone towards more complicated simulations.

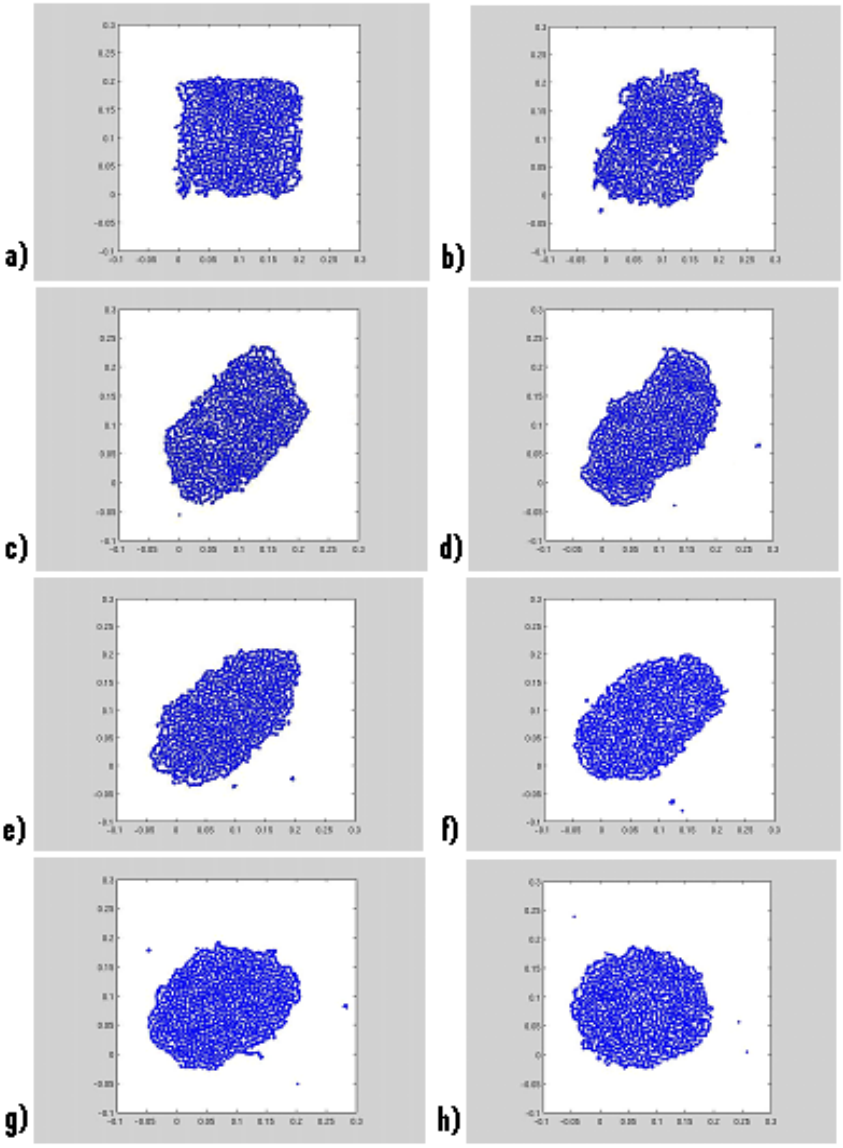


Figure 3.22. Snapshots showing the evolution of a square of water without container in a zero-gravity environment. Snapshots begin at 0s and occur every 0.2s. Units are in metres.

The water particles sit comfortably and stably next to each other throughout the simulation. The square evolves reasonably quickly into a circle and retains a circular shape thereafter

despite no external forces being present. This may seem initially surprising as the obvious result of this simulation was to remain in an unchanging square. The way a circle is formed demonstrates the way that Hydra has a naturally built in partial surface tension model. A circle (or sphere in nature) is a much more natural shape for a droplet of water to be in than a square (cube).

3.4.3 Settled cup

Taking that initial 2500 square of water particles and now placing them into a cup made up of 350 boundary particle surrounded by 925 ghost particles. Standard Earth gravity of -9.81 m/s^2 in a downward direction (y-direction) was now switched on. After some initial settling into a lower energy state than the grid lattice the particles should sit in the cup indefinitely. This test (shown in figure 3.23) builds upon the no-gravity water square test and with the addition of boundaries will fully test the stability of the water and whether or not Hydra is capable of being given real problem to solve.

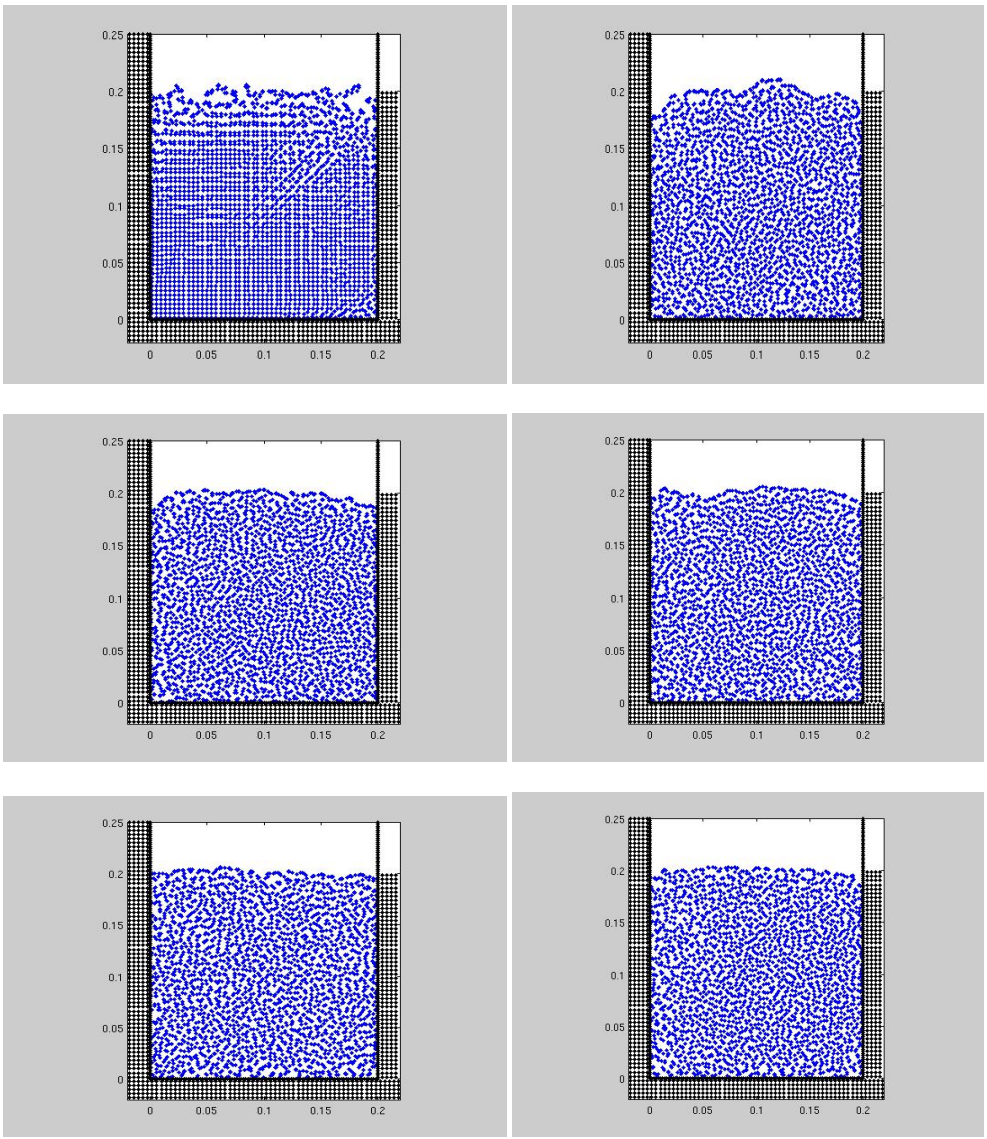


Figure 3.23. From a grid start – snapshots 0.2 seconds apart, starting at t = 0.05 seconds, showing a 20cm by 20cm container of water resting in a glass.

This is exactly what happens – proof of stable water is the result of this test. The water sat stably in the final state for as long as could be simulated. It may be interesting at this point to see the actual settling from a grid start to a glass like state in detail. It is also prudent to increase the number of particles used as 2500 is fine for a test but too few for a realistic simulation. A new open container was created from 650 boundary particles surrounded by 1625 ghost particles. A 1m by 2m rectangle of water was placed inside constructed on a grid using 6000 fluid particles. In figure 3.24 the way the particles drift into more randomised positions of the glass state can be seen.

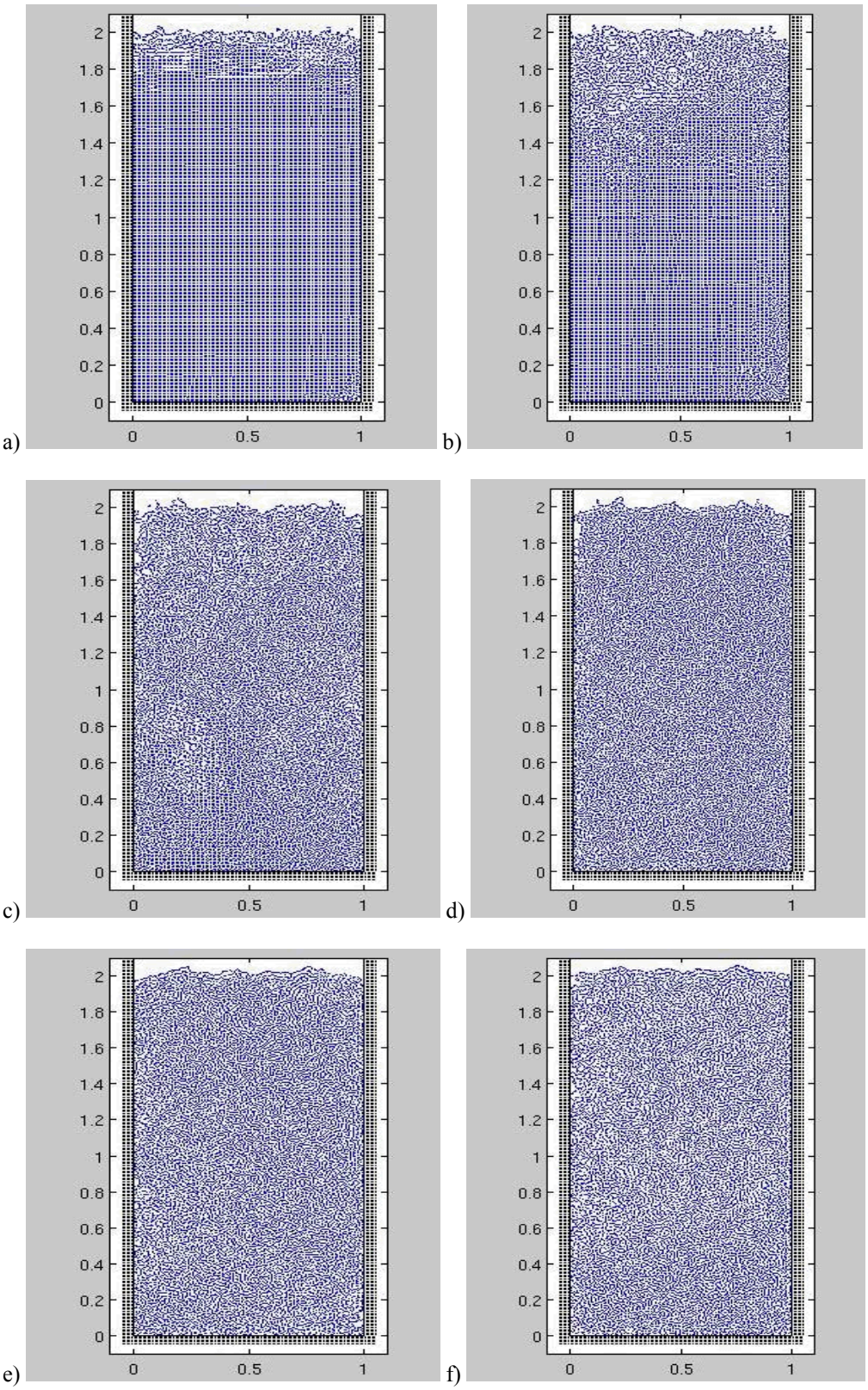


Figure 3.24. Snapshots of a 1m by 2m box of water starting as a grid and settling into a glass state to be used as the initial conditions of a new simulation. Snapshots are 0.01s apart starting at 0.01s.

This process only takes a few tens of milliseconds but can still be seen clearly. The particles at the top of the water react first. This is due to the fact that the information of the change in forces that causes the settling can only travel down at the speed of sound (set in the initial conditions). The information that the surface has appeared ripples down the container at the chosen speed of sound. So in a way for the first three snapshots in figure 3.24 the bottom line of fluid particles is not aware that the simulation has really begun and not experienced any forces yet. In the final snapshot a glass has been created and the data regarding the particle positions and velocities could be used as the initial conditions of a new simulation if required.

3.4.4 Wall Hole

Next there comes a more substantial test of the code. It must prove to be stable after free motion of the fluid under gravity and violent impacts have disturbed the free surface. The wall hole test will also prove that the mass, volume and density of the fluid are all conserved during simulation. This is a key step in demonstrating that reliable solutions can be achieved through use of Hydra with respect to water flow.

The wall hole test consists of a 1m by 3m open topped container. This is as the second settled glass test but with additional height demonstrating how one simulations final state can provide the initial conditions for a different simulation. There is a further containment vessel outside of this one making a total box of size 3m by 3m. A hole is made 1m up the separating wall (of size 30cm) and the water allowed to flow out freely under gravity. Beginning with a 1 by 3 metre area (pre-generated in the manner described in the previous example), the water should become a 3m by 1m area by the end with zero particles escaping the container walls. Figure 3.25 shows a diagram of the initial conditions.

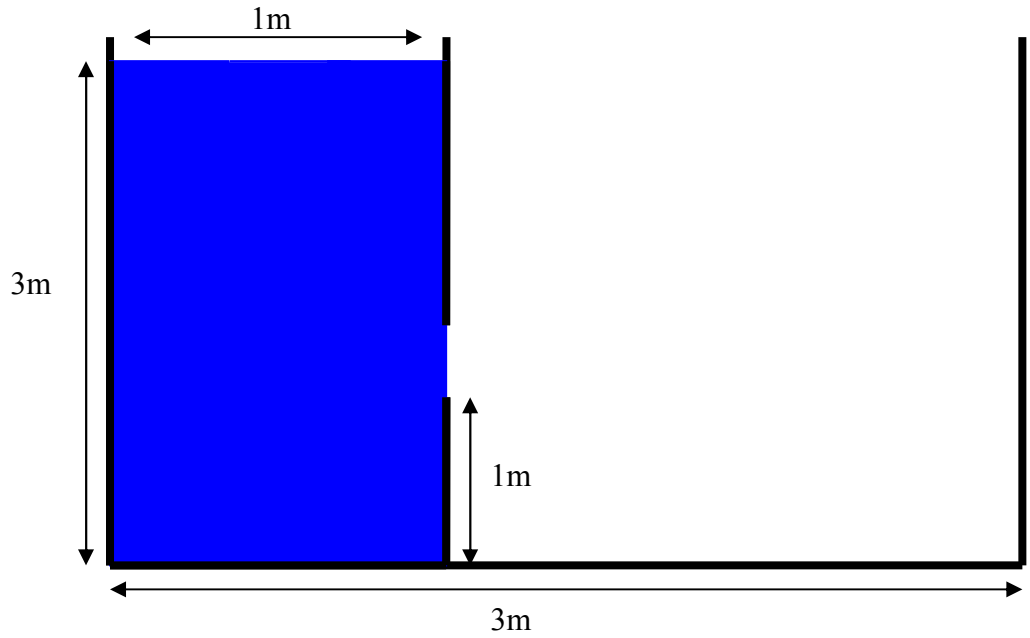


Figure 3.25. Initial conditions of the wall hole test.

Water is allowed to flow freely under gravity of -9.81m/s^2. This test is performed twice in Hydra using a different number of particles so as to be sure that mass, volume and density conservation is in no way affected by the resolution. Beginning with a lower particle number of 10800 (Figure 3.26) and setting the sound speed for a 3 metre deep water column collapse the particles are prepared. Note that for the purposes of this test, as no quantitative analysis is being done, merely visual checks, that the particles are placed on a grid structure. There is no need for the extra cost of making them into a glass.

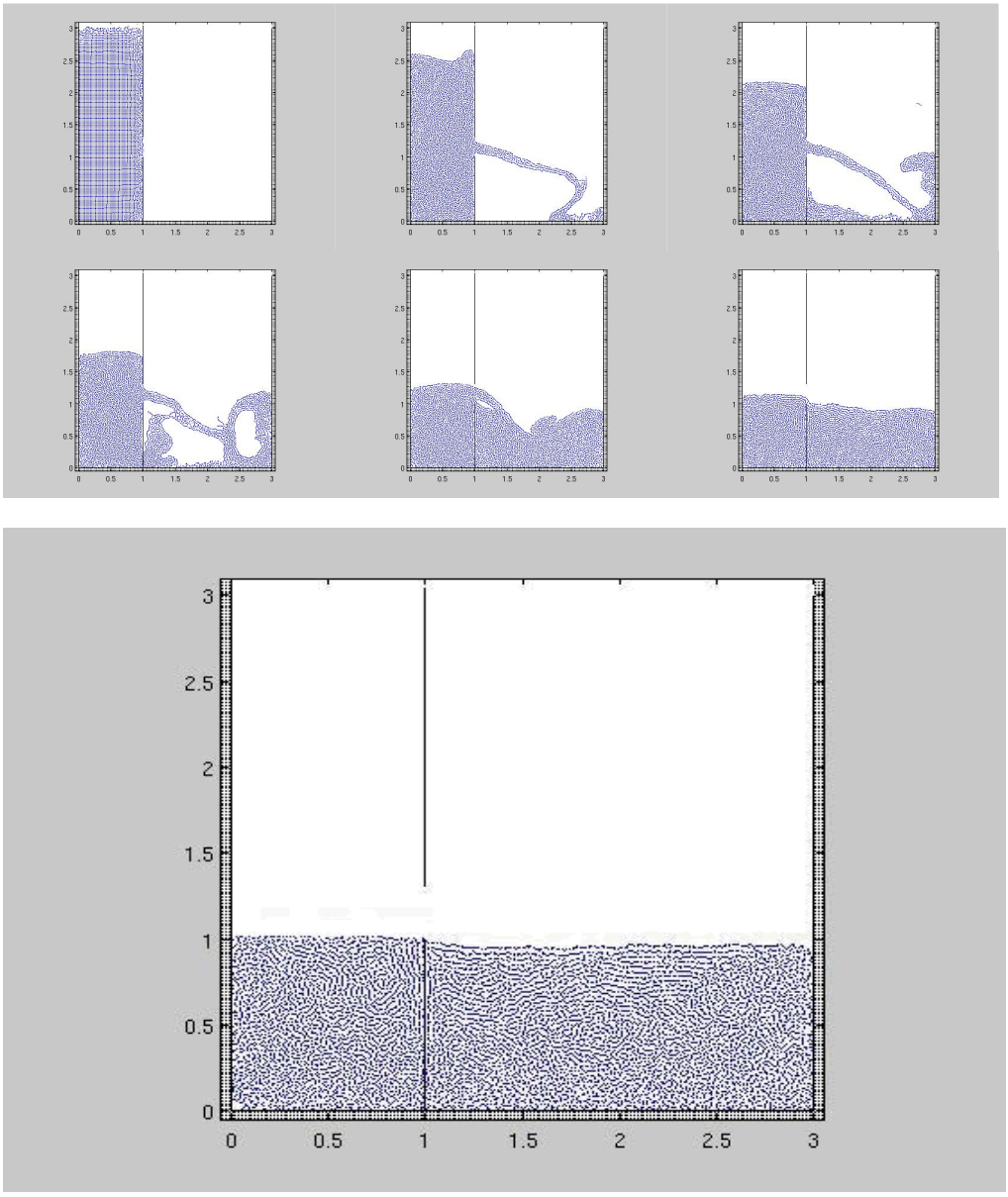


Figure 3.26. Snapshots of the lower resolution wall hole simulation at times t = 0s, 0.15s, 0.3s, 0.45s, 0.75s, 4s and 10s (large picture).

The simulation of 10800 fluid particles settles into the 3m by 1m rectangle as it is supposed to confirming volume conservation even after a violent jet like splash against the right wall. Particles fall freely under gravity when they are ejected into the right hand side of the container and the rate of ejection slows as the initial column decreases in depth (and therefore the pressure is reduced on the particles near the hole). No particles are lost from the simulation, the boundary particles backed up by ghosts provide an impenetrable barrier, and

due to Hydra's coding there is therefore mass conservation. Density is constant throughout. The simulated water takes 10 seconds to completely drain from the left hand side of the tank and allow the sloshing to stop in the right hand side which compares favourably with a theoretically calculated draining time for this problem of 11.8 seconds. A full derivation of the calculation used to produce the theoretical draining time can be seen in appendix B. The simulation is repeated using 43200 fluid particles (figure 3.27). Note that every other parameter is kept identical.

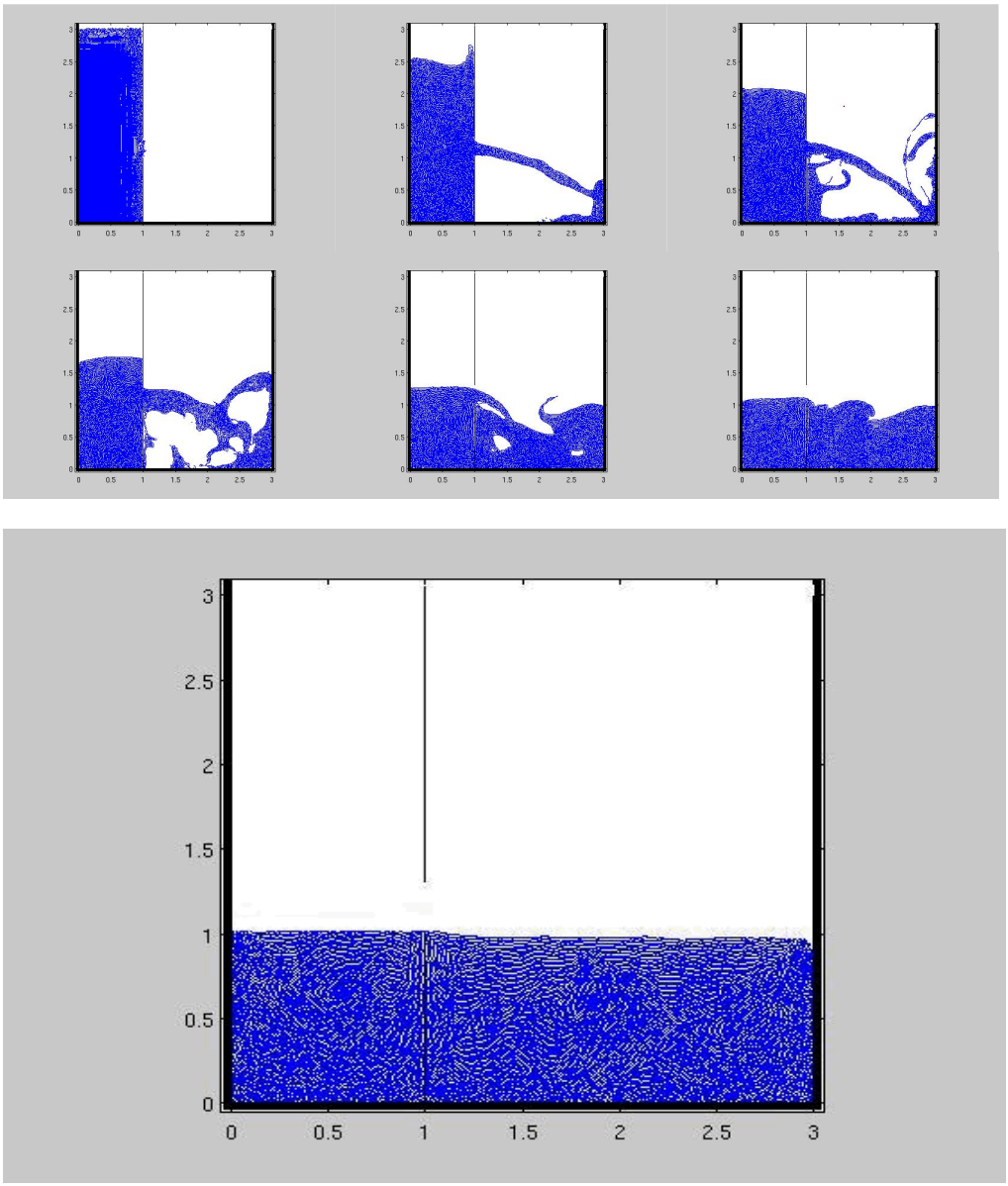


Figure 3.27. Snapshots of the higher resolution wall hole simulation at times t = 0s, 0.15s, 0.3s, 0.45s, 0.75s, 4s and 10s (large picture).

The simulation of 43200 fluid particles settles into the 3m by 1m rectangle as it is supposed to, confirming volume conservation. There is no penetration of the container walls and timing to rest of 10 seconds is unaffected. This demonstrates resolution independence.

Chapter 4

Dambreak Simulations and Validation

The study of the waves caused by the failure of a dam has attracted significant interest, e.g. Fread (1980), Goodwin et al. (1989), Morris (2000) or Frazão et al. (2007) to quote but a few authors. This can be attributed to the significant consequences of dam failure particularly in cases where dams are located upstream of large conurbations. In many cases such population centres have grown considerably since the dam's original construction. There have been fortunately few dam failures, but notable ones include the St. Francis dam failure near Los Angeles, California in 1928, the Malpasset dam failure in France in 1959 and the Teton dam failure in Idaho in 1976. Failures of more minor dams, often where there are not as many inhabitants living down the river channel, occur more regularly with several reported in the past few years leading to some loss of life, for example the Shakidor dam in Pakistan in 2005 and the Situ Gintung dam in Tangerang, Indonesia in 2009. A near failure of the Ulley dam in Yorkshire, England in 2007 highlighted potential dangers closer to home as well. Along with the significance of the consequences, the challenges of adequately capturing the physics of the problem and the difficulty of solving the associated equations mathematically have attracted the attention of researchers. A variety of typical techniques were used in the CADAM study (Morris, 2000) such as one-dimensional solutions of the St. Venant equations or solution of the two-dimensional shallow water equations. Mathematical techniques used include schemes such as the Preissmann scheme (Preissmann, 1961) or Abbott-Ionescu scheme (Abbott & Ionescu, 1967) for the one-dimensional equations as used in most commercial software. Some researchers (Alcrudo & Garcia-Navarro, 1993; Sleigh et al., 1998; Sanders 2001) have used more mathematically appropriate techniques based on techniques such as those based on the work of Godunov (Godunov, 1959) which allows for correct treatment of transitions such as Hydraulic jumps. The various authors of mentioned

here as well as others have had different objectives in their research hence the variety of techniques employed. Some were more interested in the water extent and amount of flooding whereas others were studying flow mechanics. All of these techniques make use of the shallow water assumptions that the streamlines in the flow are parallel and that therefore the vertical accelerations are negligible. This is valid for long waves where the horizontal distance is much greater than the depth. In cases where there is rapid variation in the flow, both the spatial and temporal scales are much smaller and the horizontal scale is of the same order of magnitude as the depth. Such cases have traditionally received less attention and are not so commonly studied using a numerical approach in engineering practice although this is changing. Some researchers have used the mesh-based Volume-of-Fluid (VoF) approach of Hirt & Nichols (1981) to solving the three-dimensional Navier-Stokes equations (Hargreaves, Morvan and Wright, 2007). VOF is a numerical technique designed to locate and track the free surface of a fluid. This approach has shown some benefits, but the resolution of the free surface is neither straightforward nor entirely accurate. Increased accuracy may be obtained using adaptive meshing, but at a significant cost. This style of dam failure (where the horizontal scale, i.e. channel length, after the dam is comparable to the depth of water) has recently attracted the attention of researchers using meshless methods and SPH in particular.

4.1 Tall Dambreak onto a Dry Bed

We begin with a simulation of a complete dam failure onto a dry channel. This is the simplest case to set up, though definitely non trivial as it deals with complex flows and highly disturbed free surfaces, and has been used by several researchers as a measure of code performance so therefore it is an ideal starting point to showcase Hydra's potential. A rectangular section of water is set up 1 metre wide by 2 metres tall and placed inside a box of side length 4 metres. The dam failure is considered total as there is no support to hold the water in place and it freefalls under gravity. Some researchers have also described this as a water column collapse.

Figure 4.1 is the initial set up taken from Violeau and Issa (2007).

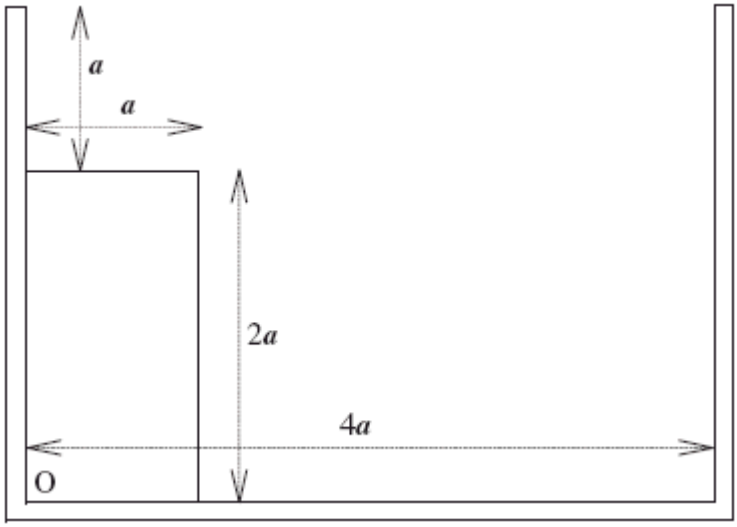


Figure 4.1. The initial conditions of the Dambreak, *a* is chosen to be 1 in this case.

This set up was chosen because it has been studied previously by Violeau and Issa (2007) at EDF Energy using their proprietary SPH code which includes a k-ε turbulence model. It has also been used in Crespo, Gomez-Gesteira and Dalrymple (2007) and is the 1st test case in the SPHysics code package (SPHysics ver. 1.4).

This sort of simulation onto a dry channel bed is designed not only to prove that Hydra provides a realistic water model visually but tests its handling of forces such as gravity and tests whether or not the interaction with solid boundaries is realistic both in continuous contact (sliding) and violent impacts. The accurate prediction of the flow's position down the channel, the location of the free surface and the behaviour of the flow upon impact with a solid object could all be crucial pieces of information to planners when deciding where to build downstream of a dam. Based upon the reports of previous researchers (Martin and Moyce, 1952; Tingsanchali and Rattanapitikon, 1993) it is entirely reasonable to consider these types of problems to be 2 dimensional and as such the width of the channel is not considered here.

Hydra was used to create the geometry (or domain) of the problem and set up the initial conditions which can be seen in Figure 4.2. This set up is a progression on from the wall hole test and settled cup test from chapter 3.4. The right hand containing wall has been removed completely after the initial conditions have been generated from a settled cup test. The basic idea is also consistent with the Sod shock simulations described in section 3.2 where a barrier separating gases of different densities was removed. Here a barrier separating liquid from a void is removed. The Sod shock test results therefore give us some confidence that Hydra is a capable tool for providing a sensible solution to this problem. To give a reasonable comparison of Hydra's performance the number of fluid particles was set to 20000, exactly the same number as used in Violeau and Issa (2007). This number should provide accuracy while keeping the simulation time reasonable; approximately 18 hours on a PC. All of the settings on Hydra were kept as indicated in Chapter 3 whilst the sound speed was adjusted to 10 times higher than the maximum expected speed inside the flow based on the depth of water at the initial release point. The maximum velocity inside the flow for this kind of event can be estimated using equation 25 in Section 2.3.

Note that the particles have been allowed to settle into a glass like state in a previous simulation and their positions read in to the main one instead of being placed directly onto a grid.

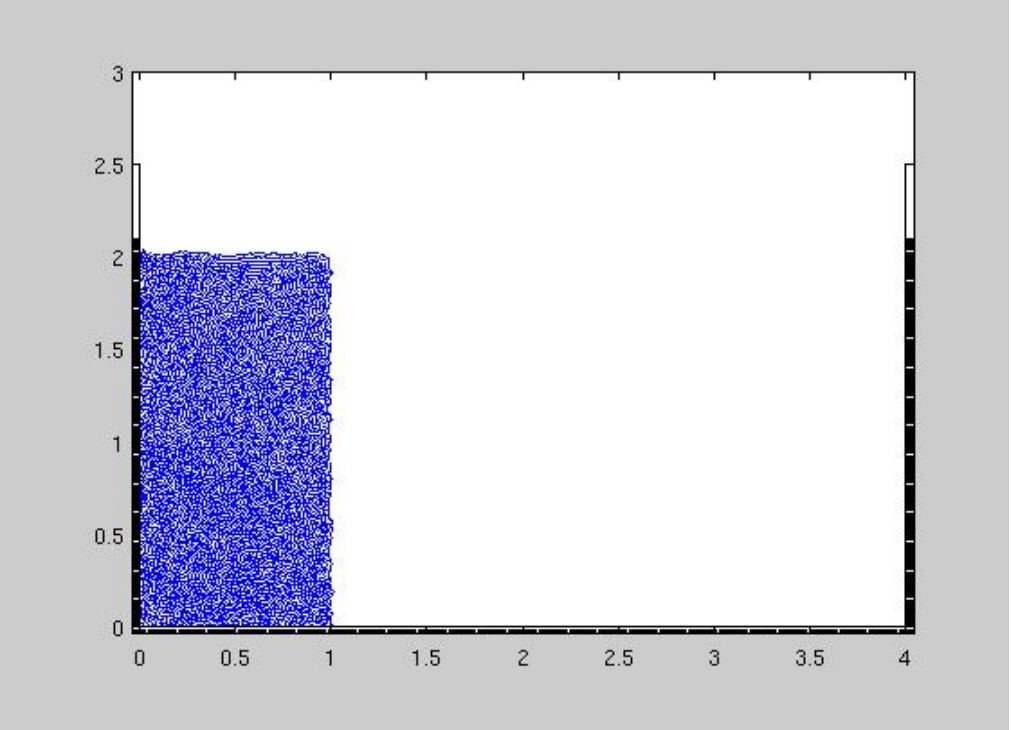


Figure 4.2. The initial conditions. From this point the particles were released under gravity. These conditions were generated inside a settled cup.

Presented in figure 4.3 are several snapshots from the simulation showing the flow of water in the initial stages of the column collapse through to the impact with the wall and the thin sheet of water which forms up the right hand side wall and finally the rebounding of the wave onto the left hand side wall.

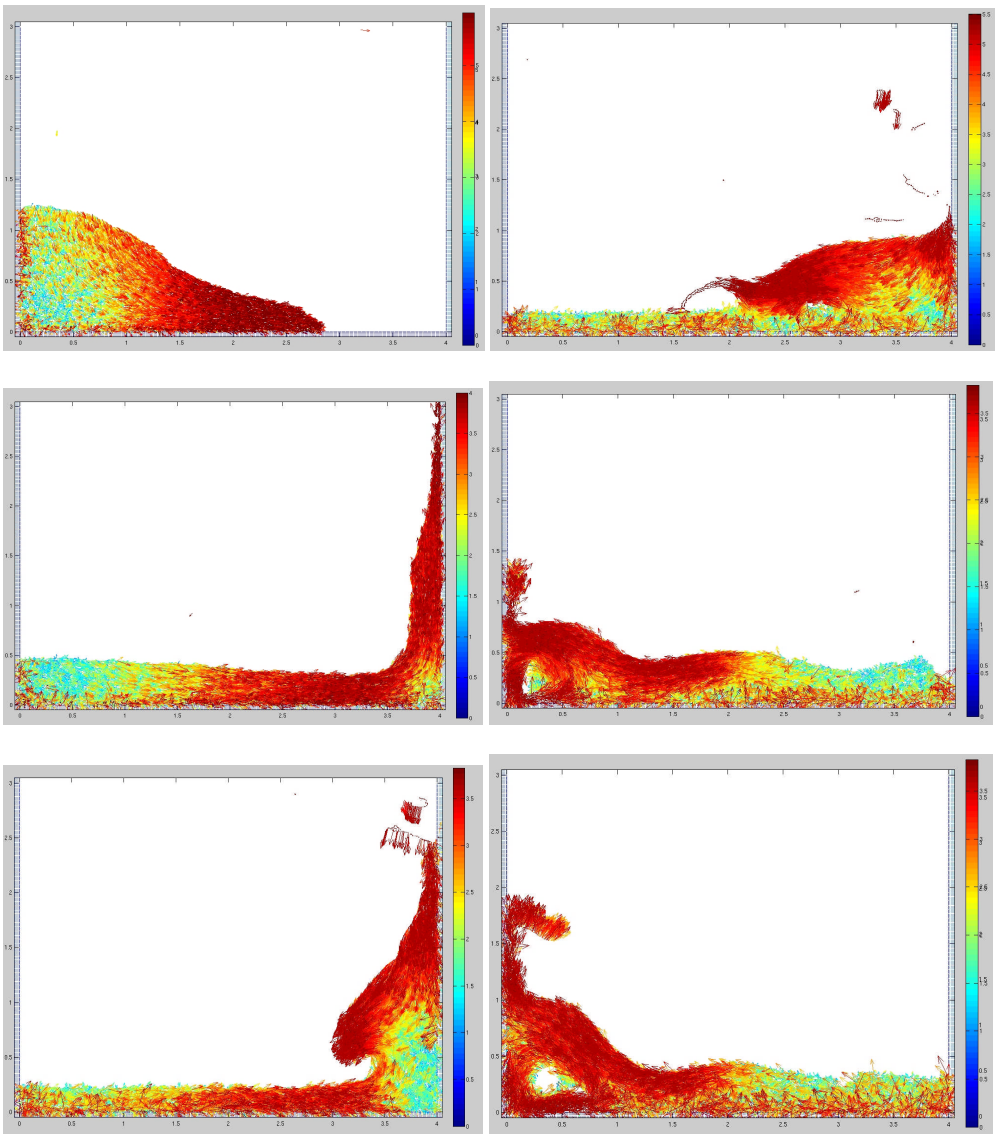


Figure 4.3. Snapshots taken from the Hydra simulation at times 0.5s, 1.1s, 1.8s (down left column), 2.1s, 2.7s and 2.9s (down right column). Note that the colour coding scheme is different in each picture with red indicating the fastest particles in each frame.

The absolute velocities measured in figure 4.3 match extremely well with the ones reported in Violeau and Issa (2007). In an identical simulation using a different SPH code Crespo, Gomez-Gesteira and Dalrymple (2007) reported that at t=0.8s the wave front had collided with the wall, at t=1.1s water had formed a thin sheet up the right hand wall and at t=1.8s the water starts to fall over. Results from Hydra show agreement with all of these statements.

A qualitative comparison can now be made between these snapshots and those originally published in Violeau and Issa (2007) (Figure 4.4).

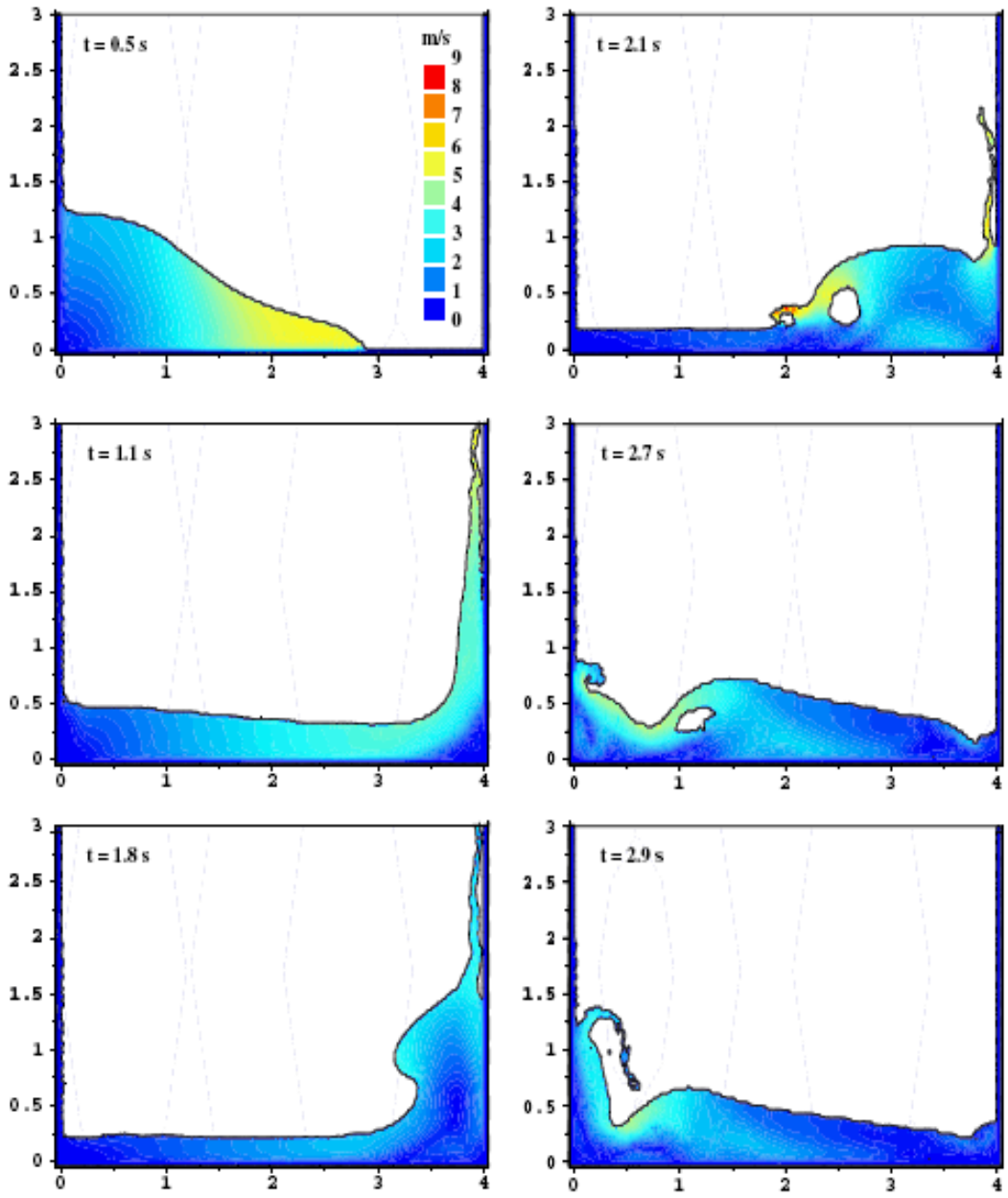


Figure 4.4. Snapshots of the simulation in progress at the identical times taken from Violeau and Issa (2007). The velocities of the water particles are colour coded according to the scale in top left picture.

The simulation was repeated using the latest publicly available version of the SPHysics code (v1.4). To maintain a fair comparison 20000 particles were again used and the settings for the code were kept as recommended by the codes publishers Figure 4.5.

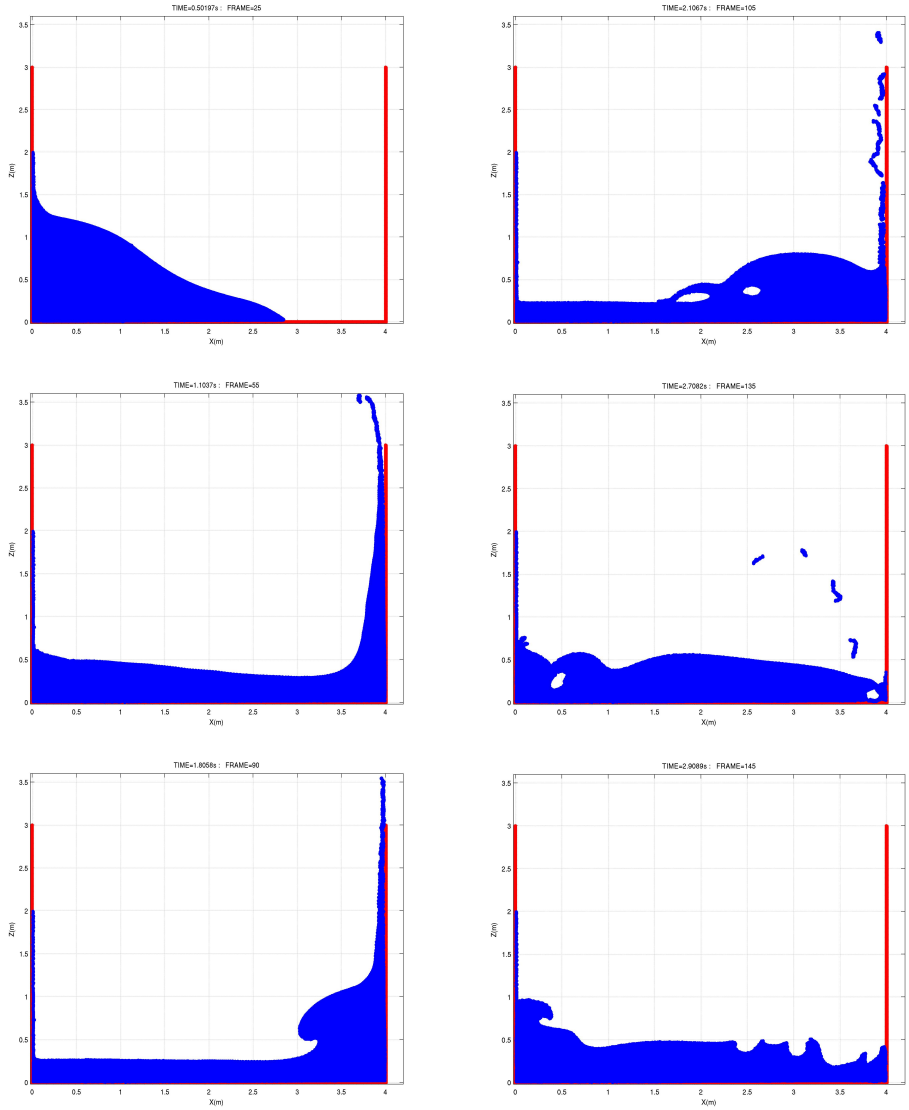


Figure 4.5. Snapshots of the simulation in progress at identical times (0.5s, 1.1s, 1.8s down left column, 2.1s, 2.7s, 2.9s down right column) for the simulation using the SPHysics code.

By examining the results of the three SPH codes it can be seen from a qualitative perspective at least that the codes agree with each other very strongly especially for the early stages of the simulation. All provide an identical timeframe for the water surge to reach the right wall; all

suggest the same height of splash up the wall and all show a similar shape and free surface location. The timeshot at t = 2.1s shows that the thin sheet of water up the right wall has fallen back down in Hydra but is still present in the other two codes but otherwise this is identical as well. By t = 2.7s the wave generated has rebounded and come back to impact the left wall. By this stage it seems as though the particles in the SPHysics simulation are starting to lose energy and momentum faster than the other two codes and the water has not travelled as far up the wall. This is also apparent at t = 2.9s. Visually the fluid appears more like treacle than water whereas Hydra still flows and churns as real water would. The code from EDF and Hydra appear more similar to each other than SPHysics. It is worthy of noting again that the results from Violeau and Issa (2007) accounted for some of the effects of turbulence via the k-ε method. This however did not appear to have any tangible benefits in this scenario. Indeed it was noted, in private communication with Damien Violeau, that in a dambreak such as the one modelled, turbulence does not have a large effect on the results (of the order of 5%). This does not imply that the k-ε method as coded by Violeau and Issa (2007) is incorrect or of little benefit, just of limited use in the current simulation.

While it is a good indicator of solution accuracy that Hydra has performed well against other SPH codes it is important to judge the performance against other computational methods, specifically market leading mesh based ones. The same simulation was repeated therefore using the latest commercially available version of CFX (version 11). The following parameters were used in the creation of the CFX model. 40000 cells, all equally sized ($4cm^2$) hexahedral elements, utilising a 2^{nd} order backward Euler transient scheme with no-slip boundary conditions. The residual target was $1x10^{-4}$ with a 0.001s timestep and a coefficient loop target of ten. The results (water volume fraction) are in figure 4.6.

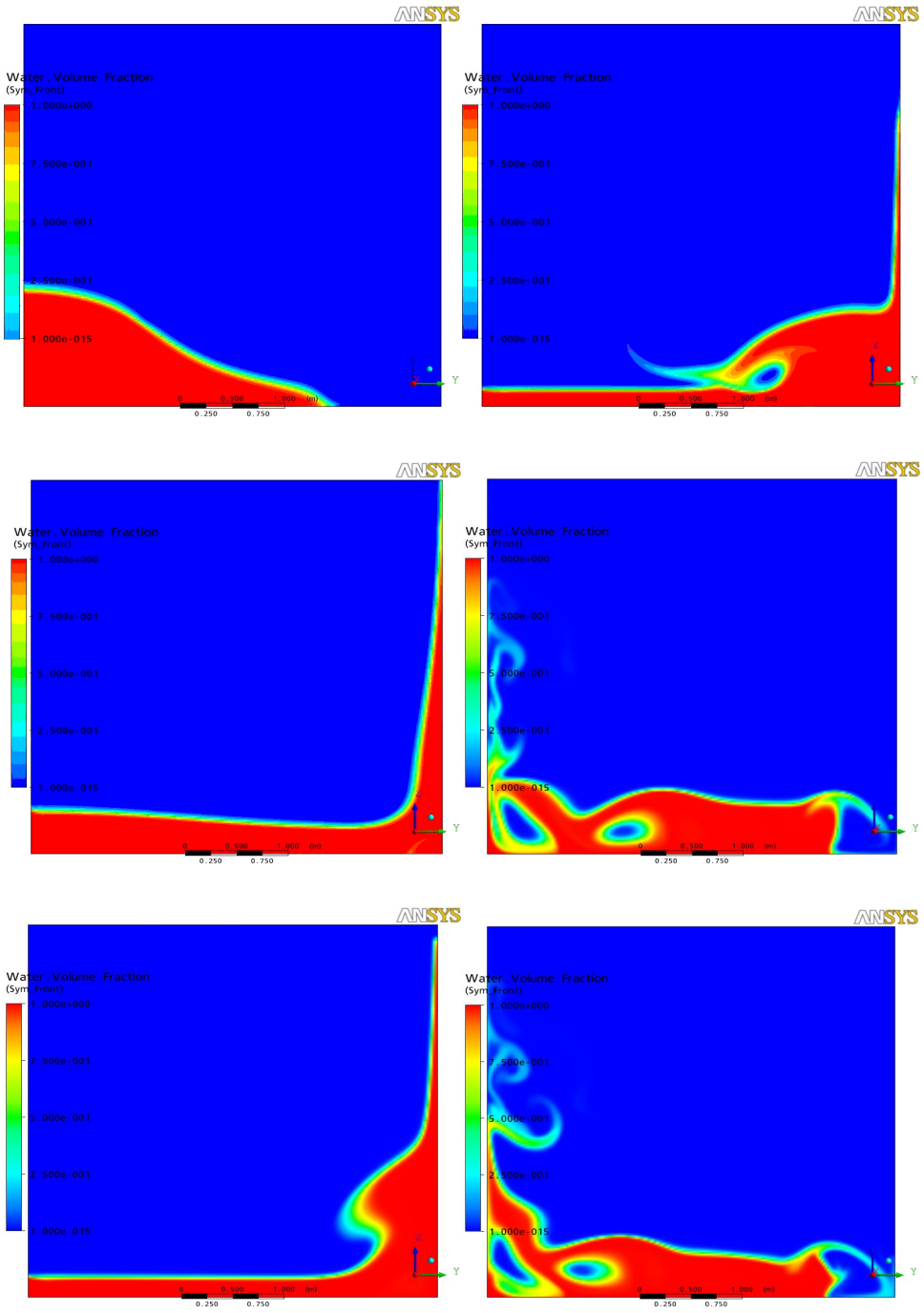


Figure 4.6. Snapshots of the simulation in progress at identical times (0.5s, 1.1s, 1.8s down left column, 2.1s, 2.7s, 2.9s down right column) for the simulation using CFX.

Comparing the CFX snapshots to the three SPH codes efforts indicates that the first three images are again identical. The snapshot at t = 2.1s shows very good agreement as well. The thin film up the right wall missing in Hydra is present but there is no dip in surface height just before it (same as Hydra) which is present in Violeau and Issa (2007) and SPHysics. In the

final two timeframes the CFX simulation begins to suffer slightly. The channel bed in the right corner can be seen to go completely dry for a fraction of a second, something which is not indicated by any other code. While there is no experimental evidence to say that this is definitely wrong, a wet bed going dry has not been observed in any of the experiments of various types carried out (these are described later) and therefore is assumed to be erroneous. When the wave impacts the left hand wall the water in CFX seems to fragment too much as seen by the colour of the water up the left wall. CFX does gain a good height up the left wall however, similar to Hydra (Hydra and CFX seem to keep more energy and momentum than the other two codes). In the left hand side corner two clear holes can be seen forming in the main body of water. The three SPH codes only have one such air gap though they disagree with which one it should be.

To get a better idea of how the codes have performed relative to each other during the initial stages of the simulation and to see the individual particles during simulation the snapshot at t=0.5s has been reproduced for all four (Figure 4.7). This time the images are zoomed in to a box stretching from 2m to 3m in the x-direction and from 0m to 0.4m in the y-direction. It is also interesting to note the different methods of boundary formation in SPH. Hydra and the code used at EDF have both used a single layer of wall particles and then several layers of ghost particles whereas SPHysics makes use of a double layer of staggered wall particles without any ghosts.

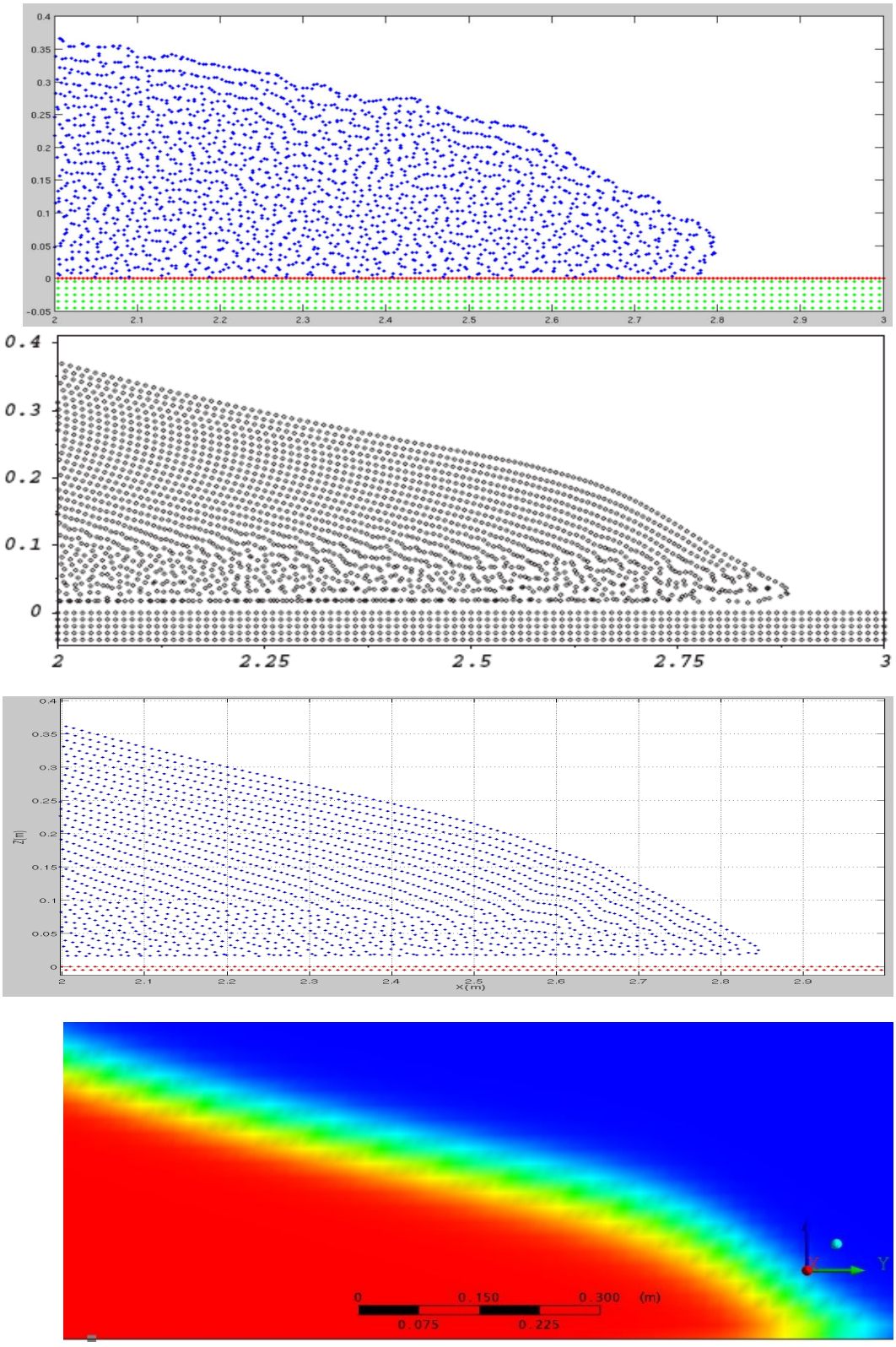


Figure 4.7. Comparison of Hydra, Violeau and Issa (2007), SPHysics and CFX at time t = 0.5 seconds. Note that the images do not represent the entire fluid body but are zoomed in to the edge of the water surge.

Figure 4.7 shows that the predictions of water surge location from all four numerical methods are extremely close. Hydra appears slightly slower down the channel than the others with it only reaching 2.8m compared to 2.85m for SPHysics, 2.875m for Violeau and Issa and CFX the quickest at 2.9m. Hydra also shows much reduction of particles remaining in chains – they are almost all completely free to move independently. This is not the case in the other two SPH codes where clear chains of particles can be seen. This is partly due to the glass start given to Hydra while the other two were started on grid but may also be an indicator of more realistic flow in Hydra. It is stated by Monaghan (2005) that increase in particle disorder as the simulation progresses causes errors to creep in to the numerical solution (though he admits by less than Monte Carlo methods predict). It is also implied in Violeau and Issa (2007) that the particle disorder in the lower levels of the simulation is undesirable. Note that the term disorder when used in this research is to imply an amorphous state (i.e. a glass) as opposed to a completely randomised state that may cause improper density estimates. It is an ordered disorder. The way Hydra is allowed to settle purposefully creates this disordered state in the initial conditions on the grounds that it is a lower energy, more natural state from which to begin. Nature tends to seek out the lowest energy state that it can across a whole range of scenarios. The hypothesis put forward in this research is that this is a better way of generating initial conditions than leaving the regular lattice of fluid particles used by most SPH researchers. The glass state allows a better approximation of liquid behaviour in the opinion of this author. It should also be pointed out that this particle disorder makes traditional error estimation as would be done in finite elements or finite differences impossible in SPH (Monaghan, 2005). Comparison with analytical or experimental data is a better indicator of solution accuracy. Hydra shows a slightly more rounded appearance to the others at the toe of the surge. Such a comparison cannot be made for CFX of course but it is interesting to point out how the density at the free surface fades over a short distance – a characteristic of grid-based solvers making exact free surface resolution in grid methods difficult (see figure 4.8). Typically the moment when the water volume fraction is reduced to 50% is taken as the mark (green colour). Techniques can be

used to extrapolate a better free surface position but the original data has been left in here to demonstrate the ease at which SPH codes can track a free surface. The free surface is easy to resolve in SPH, it is where the particles stop. Another point to be made is that there appears to be a gap between the floor boundary and the fluid particles in both Violeau and Issa (2007) and SPHysics. This does not appear in Hydra where the fluid particles contact and interact directly with the wall particles (just as real water would do). This may offer an explanation as to why Hydra appears slower, its particles pick up more friction from the channel bed.

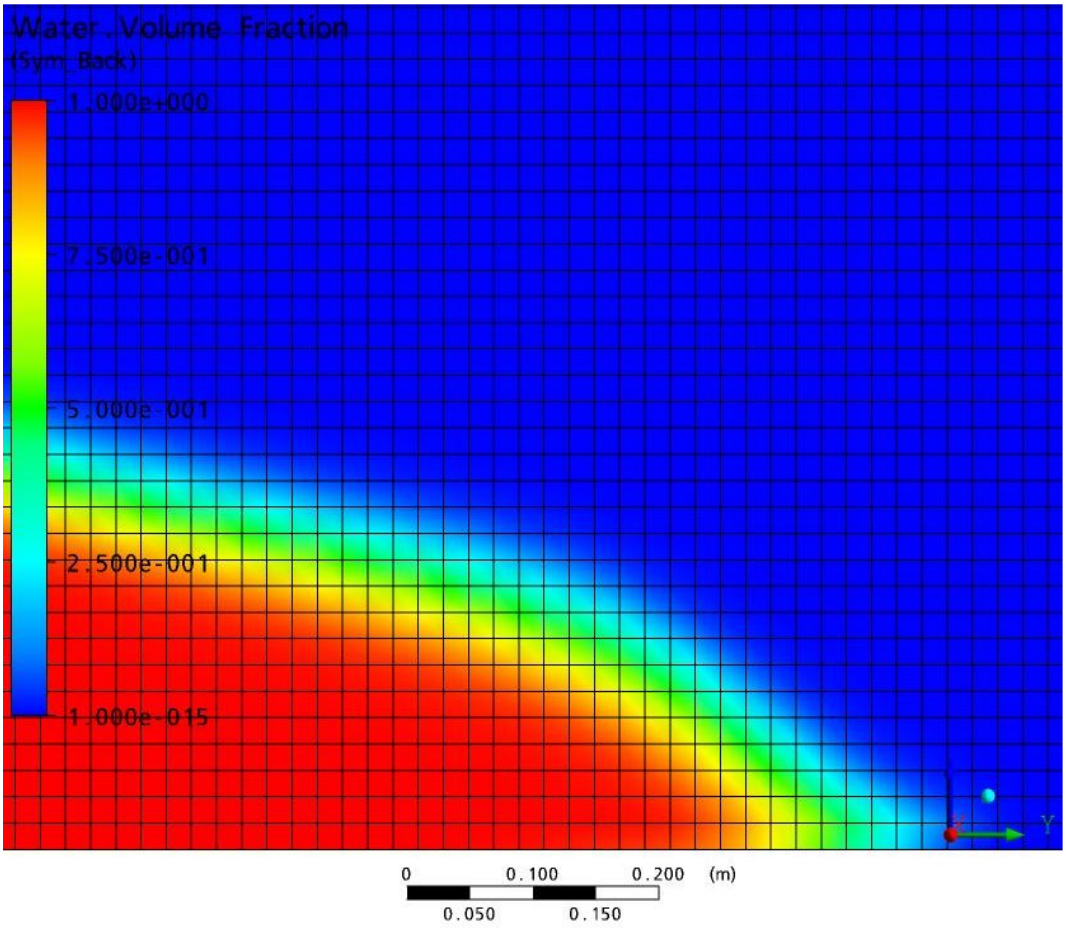


Figure 4.8. Zoomed in image of the free surface in SPH using 40000 cells. The free surface is stretched over approximately 3-4 cells widths allowing some error to creep in when determining the exact location.

The next step was to consider how the position of the surge of the wave is modelled as it travels down the channel. It could be important to know how long a surge of water would

take to travel down a river bed for example after a dam break in order to give evacuation crews a timescale to work from. This also provides a quantitative comparison for different methods. The x-coordinate of the toe of the water surge is considered against time, essentially making this a measure of the speed of the wave down the channel. Both of these quantities are plotted non-dimensionally so they could apply to any dam failure of equivalent relative dimensions. Here $X^* = x/a$ and $t^* = t / (a/2g)^{1/2}$. Hydra is plotted along with CFX. Values have also been taken from Violeau and Issa (2007) and from experimental work of this scenario done by Koshizuka and Oka (1996) in Figure 4.9. To produce this graph and the subsequent graphs of dambreak results in this thesis the analysis was done through the analysis package Matlab® with scripts written specifically for the task. Using Matlab® allowed detailed information to be extracted from the raw data and plotted so that it could simply be read off. Matlab® allows for zooming into plotted information and the scales to update themselves so a water depth, for example, can be displayed very accurately by observing the positions of the particles.

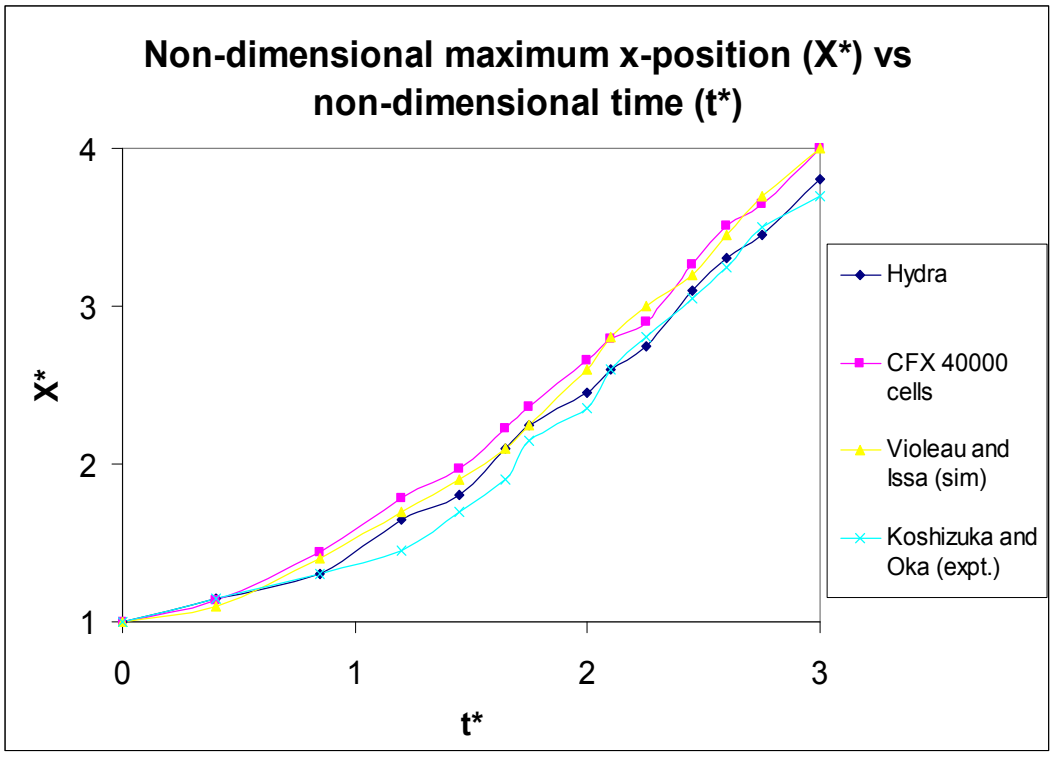


Figure 4.9. Non-dimensional x coordinate of water surge vs. non-dimensional time.

Good agreement between all the results. CFX seems to proceed slightly quicker than the others in the early stages but is soon caught by Violeau and Issa (2007) simulation. Hydra is a tiny bit slower with the experimental values being the slowest by a very small margin.

Now the water draining down the left hand wall will be looked at in detail in Figure 4.10. Again the results are plotted non-dimensionally with the water depth at the left wall H* = H/2a.

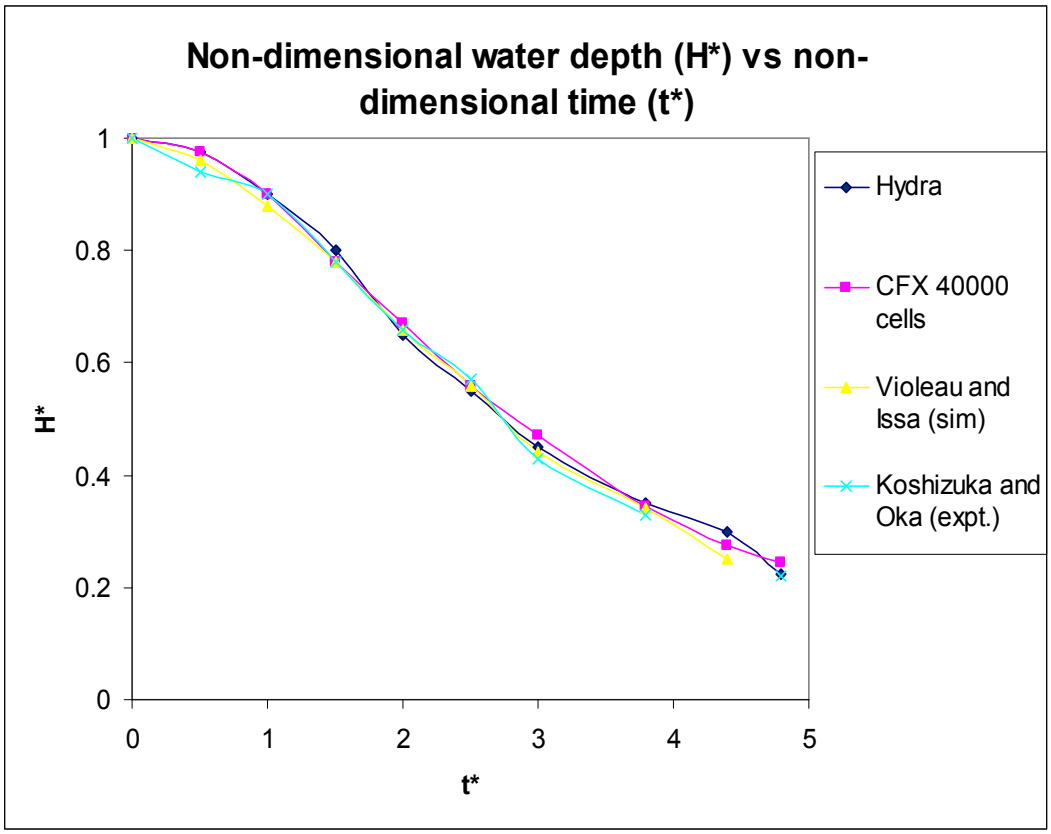


Figure 4.10. Non-dimensional water depth at left hand wall vs. non-dimensional time.

All three simulations show very satisfactory agreement with experimental values. The CFX curve is especially smooth.

The reason for the choice of 20000 particles in the SPH simulations was in order to make a fair comparison of Hydra with a leading SPH code, however in order to confirm the CFX

results were an accurate representation of the scenario a convergence check must be done. The idea of this is to repeat the exact same simulation using the exact same parameters and checking that the results agree with each other to make sure that sufficient cells were included to resolve the solution properly. Note that a convergence check will be done in Hydra later on.

The simulation was done in CFX using 5000, 10000 and 40000 cells, the grids of which can be seen in Figure 4.11;

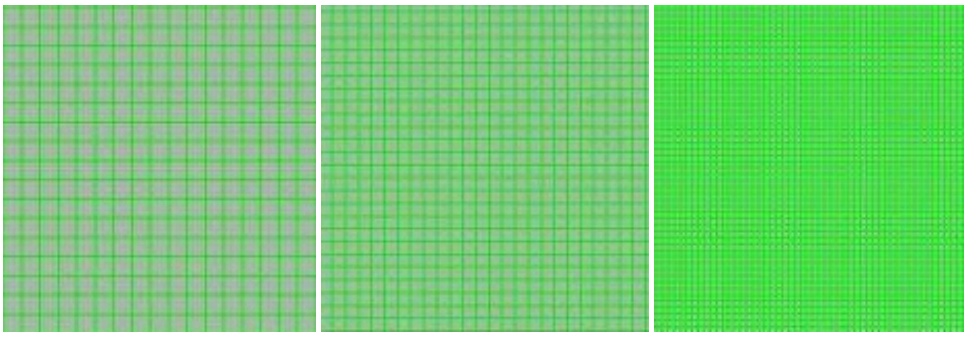


Figure 4.11. The 3 grids of 5000, 10000 and 40000 cells respectively. 1m by 1m square. Grids are uniform throughout.

In SPH a general rule is “the more particles the better the simulation” but in mesh based methods the same basic principle applies but with a caveat “the more cells in the region of interest the better the simulation”. SPH has the advantage that particles will naturally be in the region of interest as a fundamental consequence of the method. Of course increasing particle number and/or cell number increases the time required to perform the run. So a compromise between accuracy and cost is reached. The three runs at t=0.5s can be seen in figure 4.12.

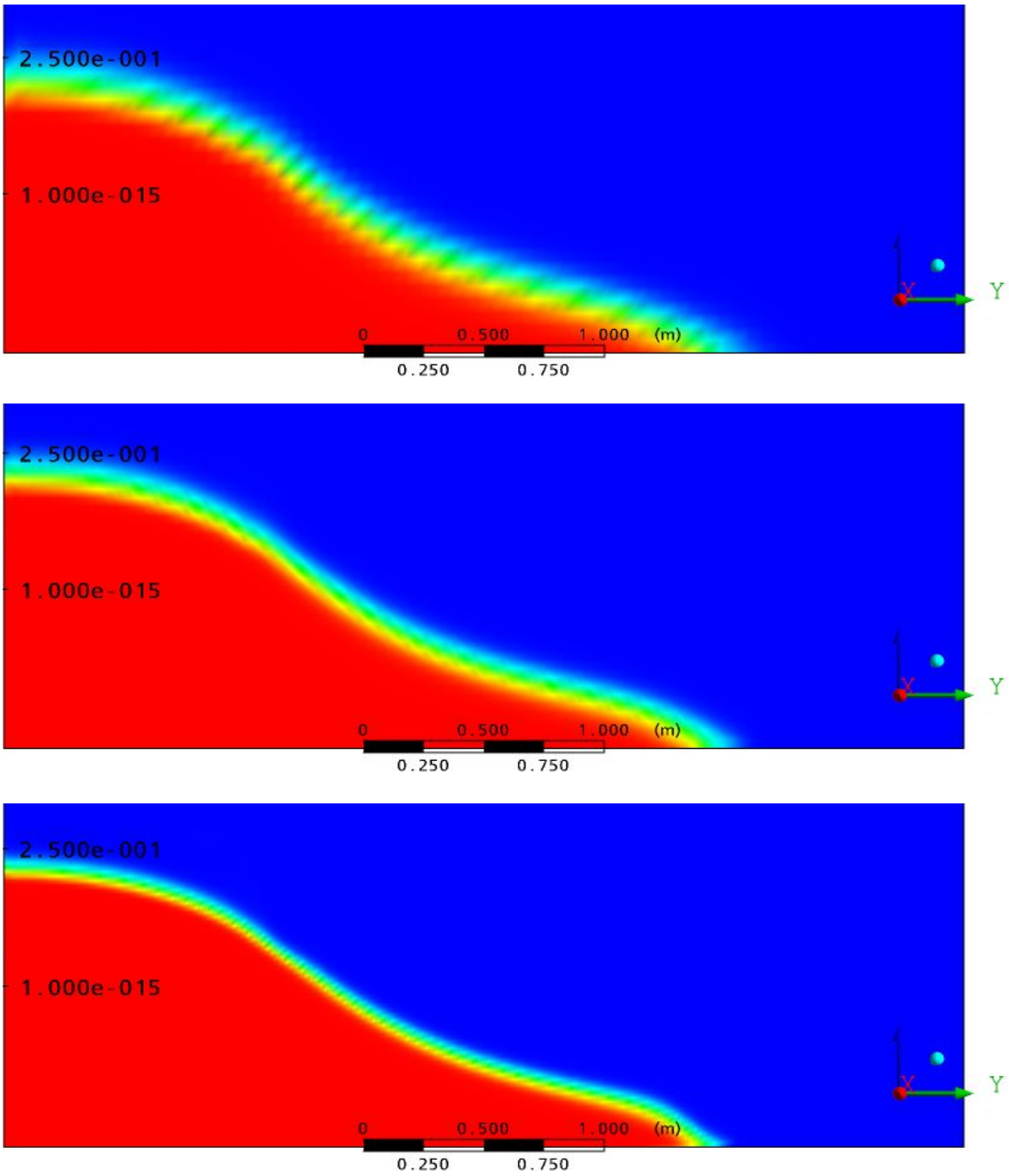


Figure 4.12. Snapshots at t=0.5s using 5000, 10000 and 40000 cells respectively.

As can clearly be seen from the images in figure there is excellent convergence of CFX even at low cell numbers with the images being close to identical for all the key points. However it is clear that there is some inaccuracy a low cell number can impart on the exact location of the free surface. The high cell number cases show a clear and reasonably well defined location of the free surface relative to the low cell number case where it is blurry and spread over a large distance. The surface being a slow fading away of density instead of a sharp

boundary can create inaccuracy in providing the surface location. Significant improvement can be seen between 5000 and 10000 cells, while some improvement is seen when upgrading to 40000 but it is not as big a jump in accuracy. Increasing further the number of grid cells should increase the accuracy of the free surface resolution further but at increasing computational cost for continuously reducing gains. It is difficult to make a resolution comparison between grid cell number and SPH particle number due to the differences in the way the domain is modelled especially as static equally sized grid cells are used. This is also noted in Agertz et al (2009) and addressed in Tasker et al (2008). It is stated in the latter of these sources that when areas of high density are the regions of interest a ratio of 1 cell to 1 particle is reasonable. When dealing with cases where this is not so, the particle number may need to be increased. It should be noted however that in Tasker et al (2008) all of the grid based codes used AMR techniques. This can substantially reduce the number of cells required compared to a non-adaptive mesh. For this simulation, 40000 cells are more than sufficient and roughly consistent with previous statements. As stated earlier the free surface takes up approximately 3 cell widths in the 40000 cell case. It is also convenient to note that the simulations of 20000 SPH particles in both Hydra and SPHysics and 40000 grid cells in CFX were completed in a very close time (approximately 18 hours on a modern PC).

In order to confirm the convergence of the CFX solution however, a quantitative analysis is required. The results from the graphs comparing CFX's performance against Hydra, Violeau and Issa (2007) and Koshizuka and Oka (1996) in tracking the x-co-ordinate of the water surge and the water depth at the left wall have been reproduced with the analysis done for all three cell numbers (Figures 4.13 and 4.14).

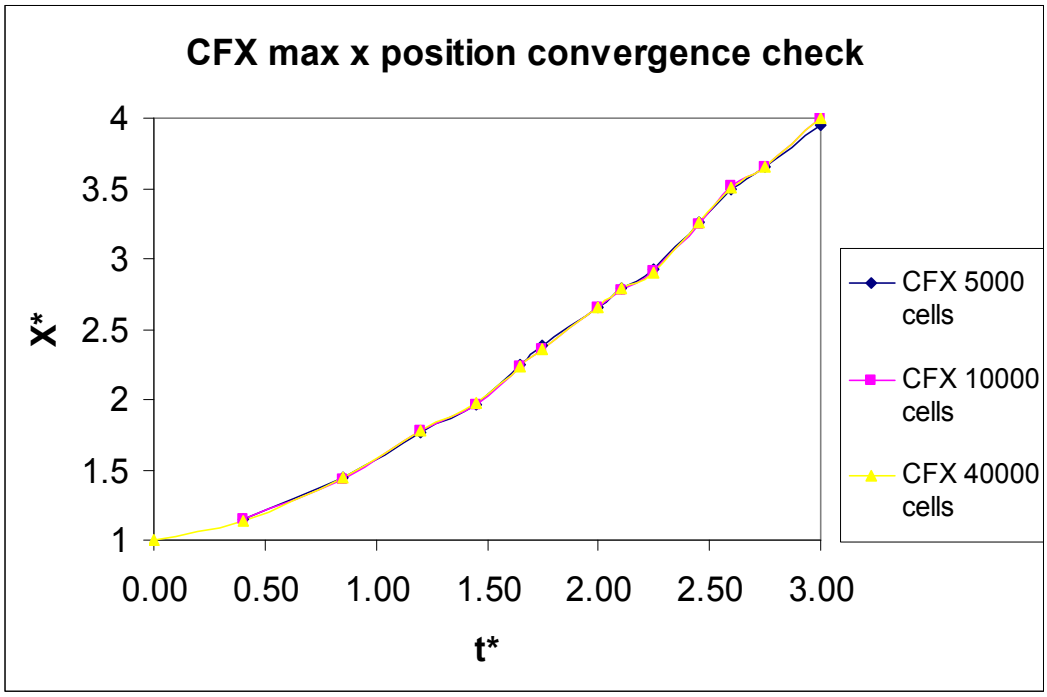


Figure 4.13. Non dimensional x coordinate of water surge vs. non dimensional time for all three CFX runs.

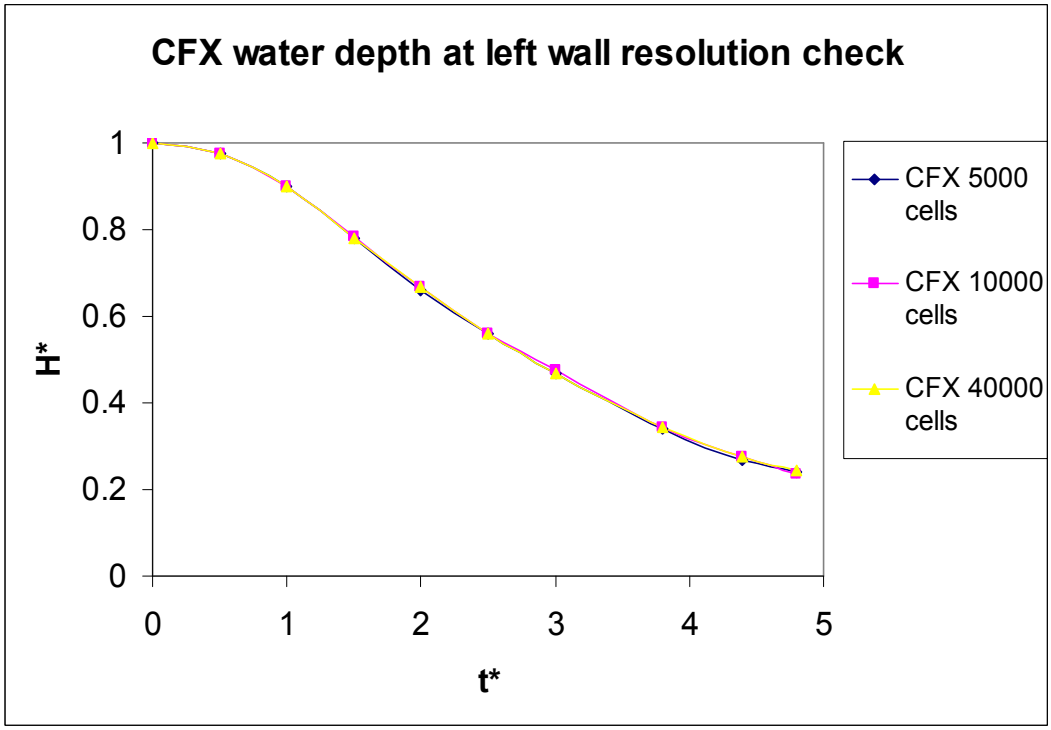


Figure 4.14. Non dimensional water depth at left hand wall vs. non dimensional time for all three CFX runs.

As can clearly be seen the convergence level is excellent. They are impossible to tell apart using these methods of analysis. Indeed the only tangible benefit to using the higher cell number seems to be the increased resolution of the free surface.

Looking at the above results as a whole there is broad agreement between the different simulation methods and with the experimental values of Koshizuka and Oka (1996). Hydra has shown that it can compete with the best SPH water codes currently available on this level of simulation. All of the codes display an ability to accurately model a water surge down a channel and water moving down a wall. All the simulations indicate a very similar free surface profile up until the wave has rebounded to the left wall. However as experimental results were not available to indicate the location of the free surface this cannot be validated. The level of agreement between the different methods (especially as CFX is commercially released) is a good sign though. CFX has shown a remarkable level of resolution independence but has highlighted a disadvantage to SPH when it comes to plotting the free surface of a moving body of water.

4.2 Wide Dambreak onto a Dry Bed

Dambreaks onto a dry bed have been considered by several researchers in varying degrees of detail. Following on from the previous simulation in Section 4.1 a similar set up has been attempted based upon the experiments of Martin and Moyce (1952). A variety of researchers using different methods (e.g. Colagrossi and Landrini (2003), HR Wallingford (2007), Veen and Gourlay (2008)) have used this as a measure of a computational code. Hydra will be compared against these published results. The volume of water is the same as previously but instead of a 1m by 2m rectangle a 2m by 1m area of water is used (Figure 4.15). This gives the dambreak a shallower height to fall from and a shorter relative channel length.

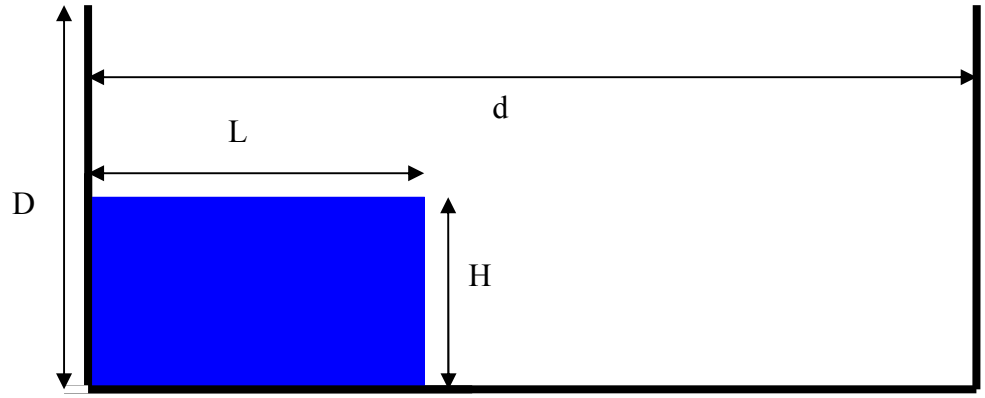


Figure 4.15. Set up of dam break onto dry bed. L = 2m, H = 1m, d = 5.366m, D = 2m.

Using a height of 1m to set up the sound speed; the initial conditions were generated in Hydra, left to settle into a glass and released. A visual comparison of the simulation in progress can be seen in figures 4.16 and 4.17 compared with the results from Colagrossi and Landrini (2003) and HR Wallingford (2007). Colagrossi and Landrini (2003) have used a different SPH code to perform their simulations whilst the HR Wallingford report (2007) shows CFX results. The snapshot times are measured in non-dimensional form according to equation 33;

$$t = \frac{t^*}{\sqrt{\frac{g}{H}}} \tag{41}$$

where t = time, t* = non-dimensional time, g = gravity, H = initial water height.

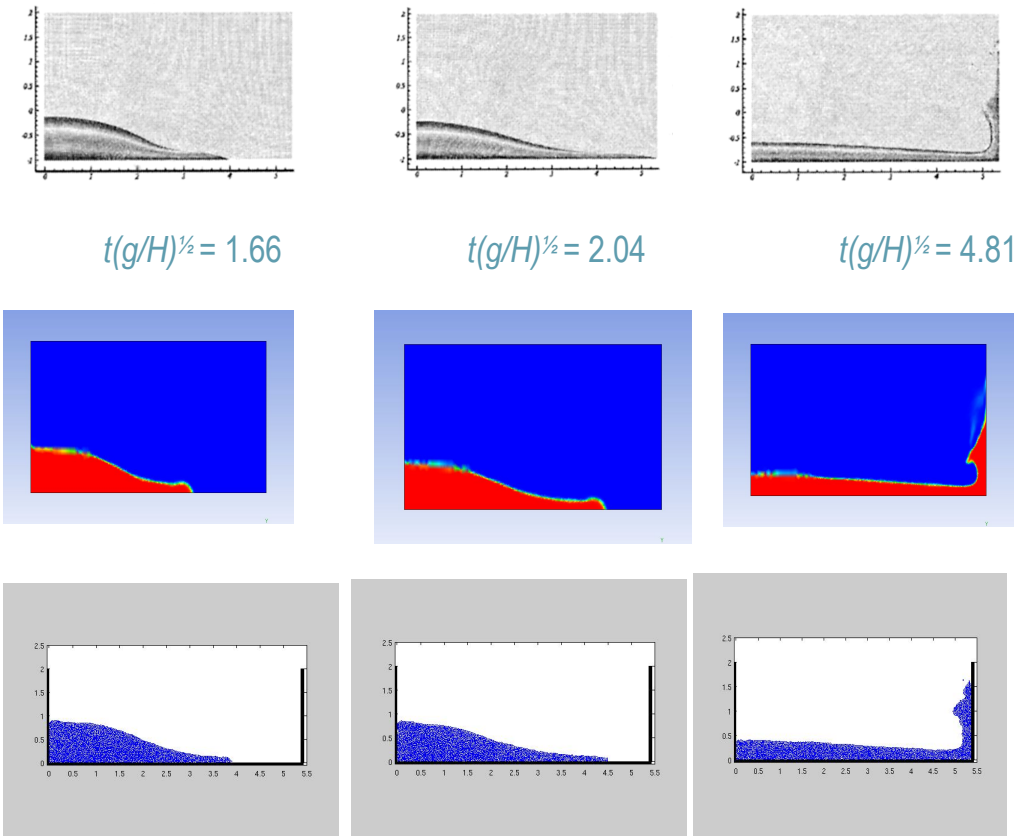


Figure 4.16. Snapshots of dam break onto dry bed at non dimensional times indicated in-between first and second row pictures. 1st row from SPH simulations by Colagrossi and Landrini (2003). 2nd row from CFX simulations by HR Wallingford (2007 presentation). 3rd row are Hydra results.

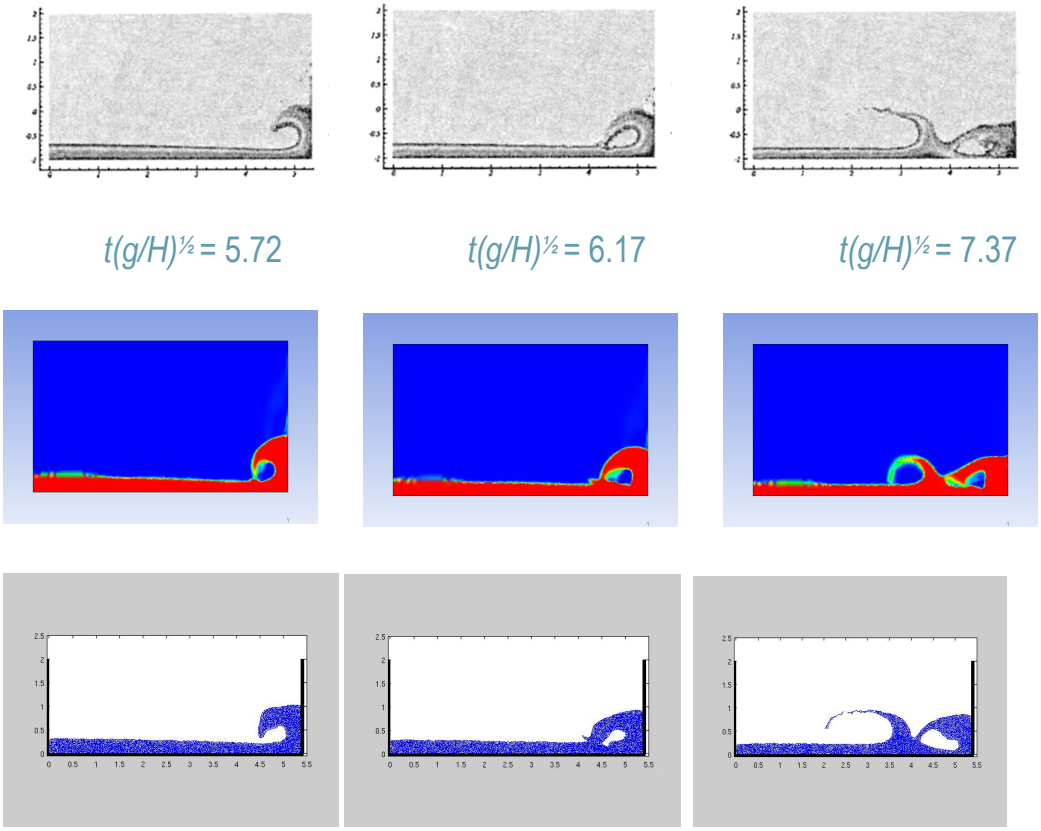


Figure 4.17. Continuation from previous figure.

There is good agreement between all three methods right up to the end of the simulation at 2.35 seconds (non-dimensional time of 7.37). This is enough time to show the water surge reaching the wall, climbing up it and crashing back down into the flow. There is no indication as to how the methods would compare after this point and again there is no experimental validation of the free surface. The splash height is consistent for all three models though there is some evidence that CFX is moving slightly quicker than the SPH codes.

A separate SPH code has been used to consider this problem by Veen and Gourlay (2008). A brief qualitative comparison is included in Figure 4.18.

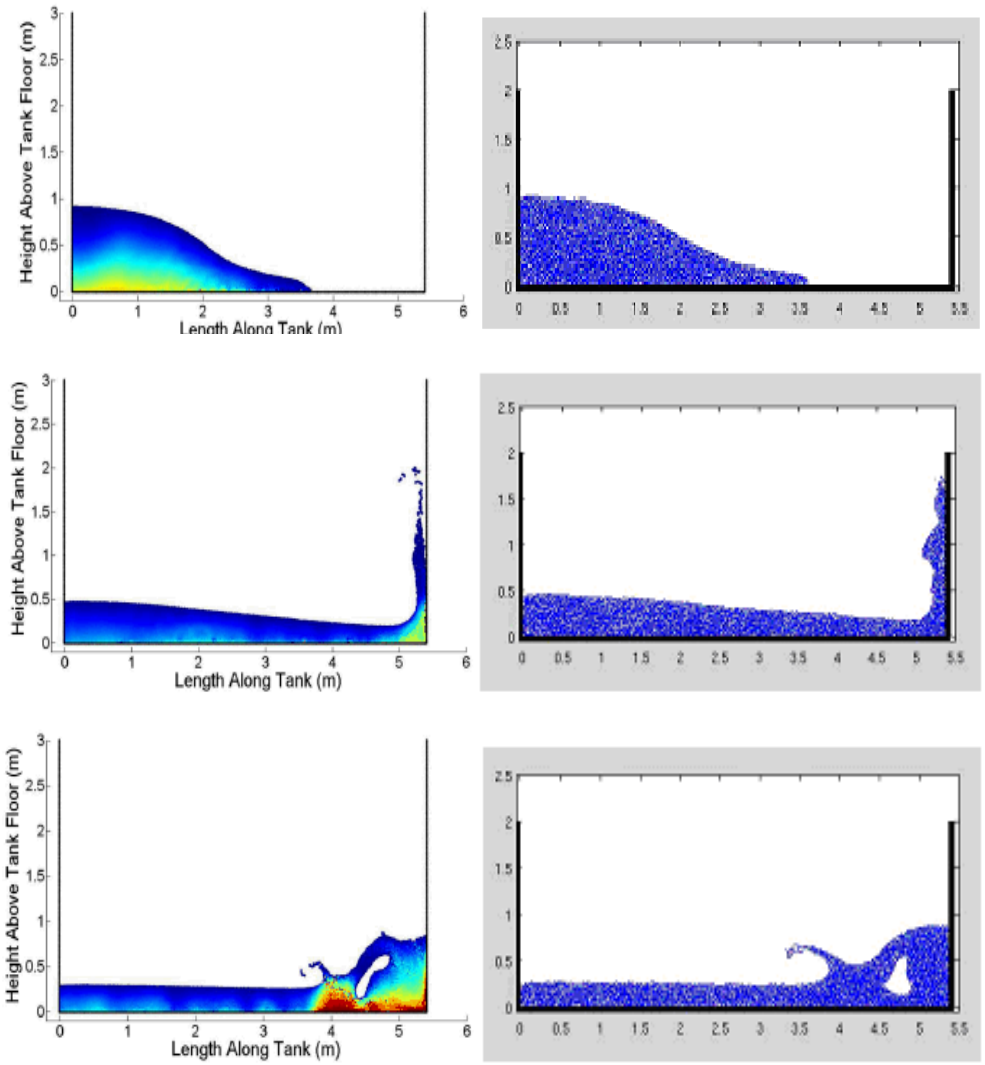
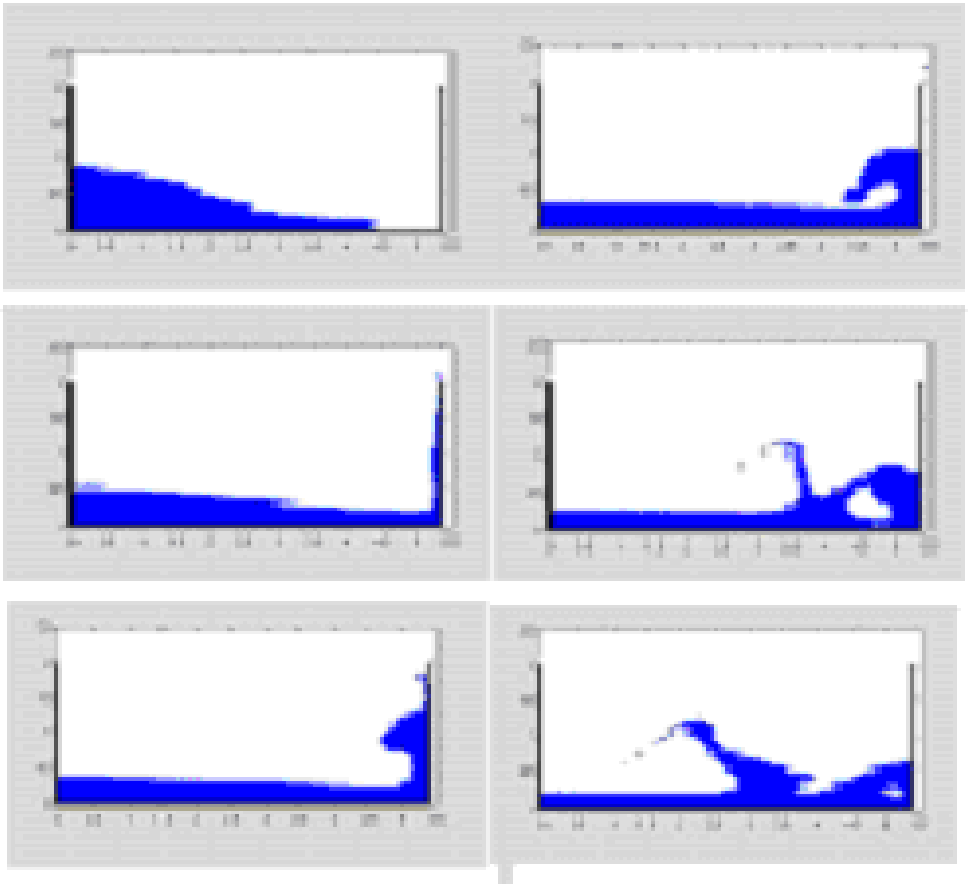


Figure 4.18. Snapshots of dam break onto dry bed at times t = 0.48s, 1.4s and 2.1s. Left hand column shows results from Veen and Gourlay (2008). Right hand column shows Hydra results.

Once again the results show excellent similarity. The water can be seen to be the same height against the left wall throughout and climbs to the same height on the right wall. Analysis done by Veen and Gourlay (2008) confirms these findings by comparison with Colagrossi and Landrini (2003) where there is no notable difference and with the experiments of Martin and Moyce (1952). The only difference is a slower surge front velocity from the experimental values which was also noted in the previous simulation. This is consistent with findings from Hydra. According to Streeter, Wylie and Bedford (1998) the maximum run up

height on the right wall for this problem should be approximately 2m. Both Hydra and Veen and Gourlay (2008) come close to this target but fall slightly short.

Abdolmaleki, Thiagarajan and Morris-Thomas (2004) use Fluent to simulate a dimensionally identical dam break to the one investigated here. Fluent is a grid based fluid dynamics package based on an Eulerian finite volume method. The initial conditions of the problem they have solved are L = 1.2m, H = 0.6m, d = 3.22m. If these numbers are multiplied by one and two thirds they become identical to the original problem. Both x and y co-ordinates of the water and the channel length are dimensionally the same and as the timing of the Fluent simulations has been given non-dimensionally; the results are directly comparable. It is not clear from the paper exactly how many grid cells are used in the published results but of the order of 10000 plane quadrilateral cells is implied. They were a mixture of sizes, with a finer grid in the expected region of the free surface. Hydra has been added in Figure 4.19.



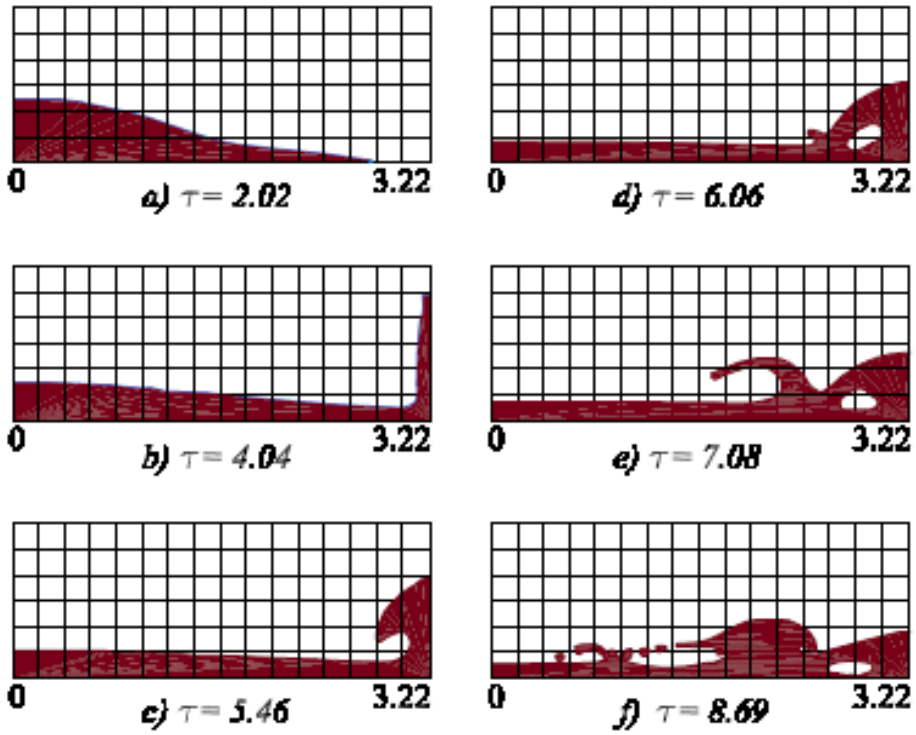
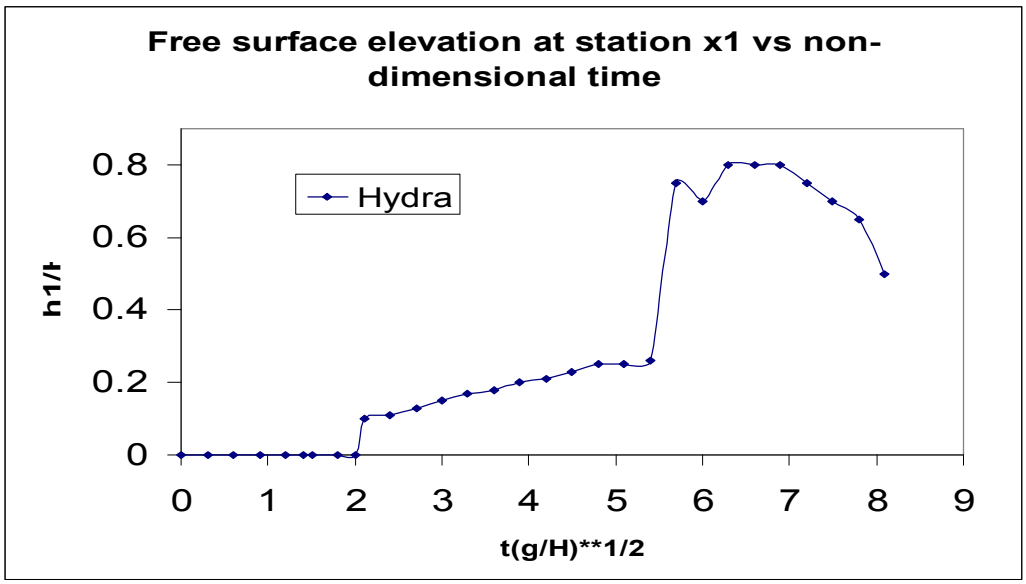


Figure 4.19. Snapshots of the simulation in progress with Hydra (first set) and Fluent (second set). The non-dimensional times of the snapshots are given in the Fluent images.

Good agreement between the two methods until the final snapshot where there is some discrepancy; a higher, more forceful splash is observed in Hydra with Fluent travelling further back towards the left wall. Similar to CFX, Fluent appears marginally faster than Hydra throughout. Note that the times of these snapshots are different to the other methods already detailed so cannot be compared.

The tracking of the free surface of water down a channel or in a flood inundation event is an important aspect of this area of fluid dynamics that has so far been overlooked for this problem. Abdolmaleki, Thiagarajan and Morris-Thomas (2004) have used a Volume of Fluid (VOF) scheme to compare their Fluent results with experimental values gained from Zhou, Kat and Buchner (1999). The measurements take the form of tracking the position of the free surface as it evolves over time at two separate places down the channel. There is no experimental validation for measuring the whole channel free surface but studying how the water depth changes at particular points down the channel should give us an idea of accuracy.

The two measuring stations are listed as x1 – 2.725m down the channel and x2 – 2.228m down the channel. Remembering that the Hydra model is 5.3m / 3.22m = 1.6 times larger than the Fluent model the same modification to the measuring positions must be made. So, x1 and x2 in Hydra become 4.485m and 3.667m downstream respectively shown in Figure 4.20 and 4.21.



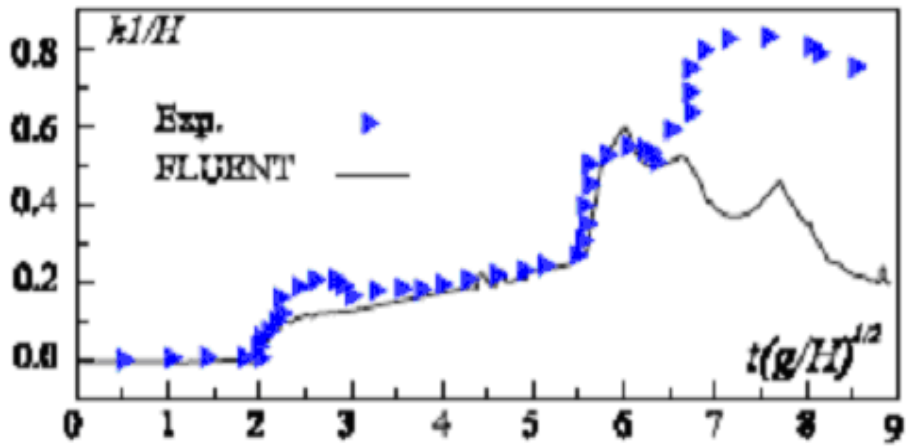


Figure 4.20. Variation of free surface level over time at measuring station x1.

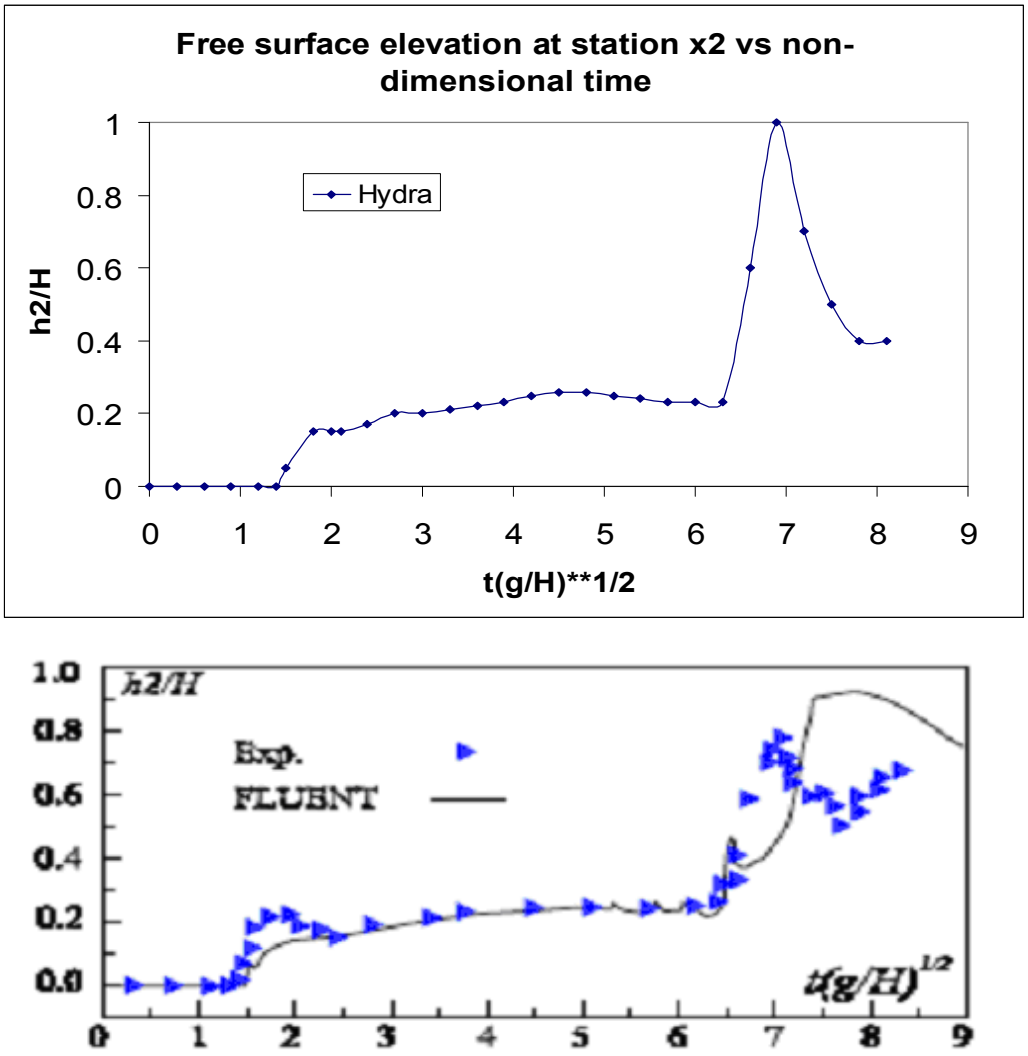


Figure 4.21. Variation of free surface level over time at measuring station x2.

At both measuring stations the timing of the surge first striking the station is the same in all three but in both experiments there is a bigger initial jump, going up to 0.2 instead of 0.1 in both simulations. After this results closely agree for all three until τ~6 for measuring station x1 and τ~7 for measuring station x2. It is after this point that the wave of water has rebounded off the wall and has re-entered the main flow travelling in the opposite direction. This creates a disordered flow and splashing. After this there is disagreement between the results. The free surface height in Hydra is characteristically higher than Fluent and marginally higher than the experiment but overall shape of the Hydra plot follows that of the experiment closer than Fluent.

4.2.1 Initial Water Toe Surge

Abdolmaleki, Thiagarajan and Morris-Thomas (2004) have also gone on to look at the initial evolution of the water front (x-coordinate) in the moments after the release of a total dam failure in detail. This simulation takes the form of a 5.7cm by 5.7cm square of water that is released and simulated for only 0.2s and is designed to test how codes deal with the very early stages of a Dambreak or water column collapse. A comparison of results from Fluent (FVM), experiment, SPH, Boundary Element method, Leve Set method and an analytical solver solution (Ritter) is shown in figure 4.22. The graph has been taken from Abdolmaleki, Thiagarajan and Morris-Thomas (2004) who have themselves taken data from Martin and Moyce (1952) and Colagrossi and Landrini (2003). Hydra has been added on the right.

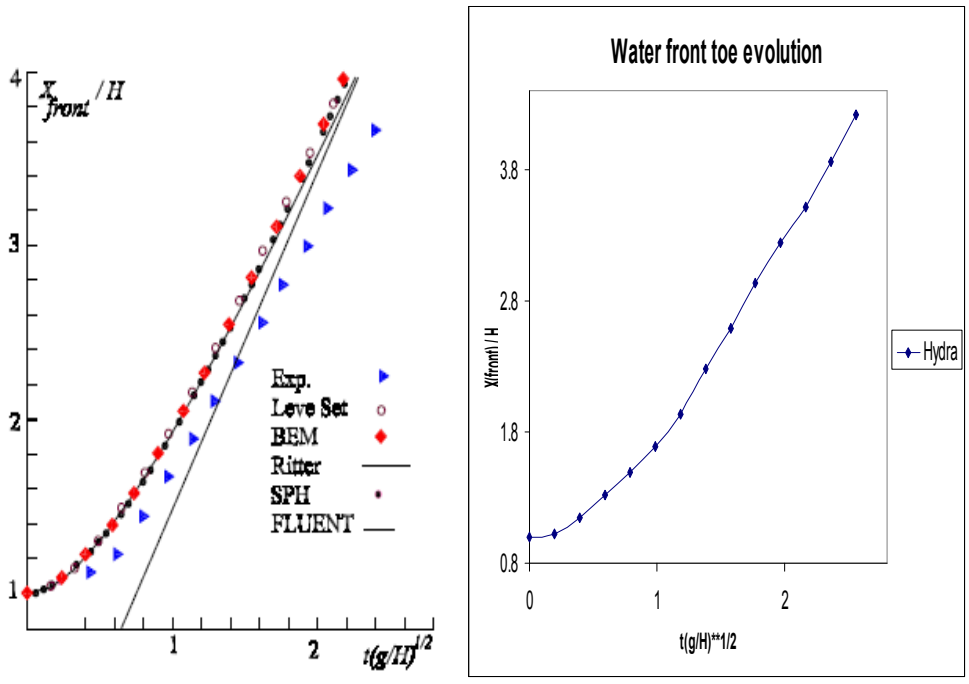


Figure 4.22. Comparison of Hydra (right) with other methods in the initial water surge model.

Hydra shows slightly reduced speed compared to the other simulation methods putting it between experiment and other simulation methods. According to Abdolmaleki, Thiagarajan

and Morris-Thomas (2004) the other numerical results asymptotically approach the shallow water solution as time increases. It is not clear if Hydra has evaluated the physics better than the others putting it closer to experimental values or if the experiments themselves had imperfect initial conditions or physics not contained with the other models.

Overall the results in this section provide a similar conclusion to the previous one, with good agreement between simulation methods and experiment. Hydra and SPH have demonstrated a clear ability to simulate the effects of a simple dambreak as well as a variety of codes and methods. The free surface of a moving water surge was tracked and compared to a grid based method. Once again numerical methods seemed to overestimate the water surge velocity but not by much. Flow down the channel in Hydra simulations comes out marginally slower than other numerical methods, with a possible explanation being that it picks up more friction from boundary interactions along the channel bed.

4.3 Dambreaks with Experimental Validation

In sections 3.3, 3.4, 4.1 and 4.2 Hydra has been successfully converted to water simulation and solutions to dambreak problems have been compared to other simulation codes and methods. While some comparisons against experimental results have been shown there has so far been a lack of validation of the free surface across the entire channel. In order to ensure the simulation results were accurately representing the flow of real water a laboratory experiment was devised to measure a variety of properties. This has the advantage that both experiment and simulation were performed by the same researcher ensuring an exact match of parameters. With the full experimental data available the free surface shape can also be validated at every stage of the simulation. This is of clear interest to anyone studying open channel flow and flood inundation. The experimental design consisted of an open-top tank made of clear Perspex with a gate near one end. Water could be contained at this end until the gate was pulled out (breaking the dam). A removable barrier was also placed inside the

tank to increase the complexity in some of the runs as can be seen in figures 4.23, 4.24, 4.25 and 4.26. The resulting flow was recorded on a very high speed digital camera (600Hz frame rate) allowing for frame by frame measuring and movie creation. Visually the shape of the flow could be compared and the time taken for the water to reach the end of the tank and the height of splash measured. This experiment was inspired by a similar set up in Maxwell (1977) but we use a dry bed.

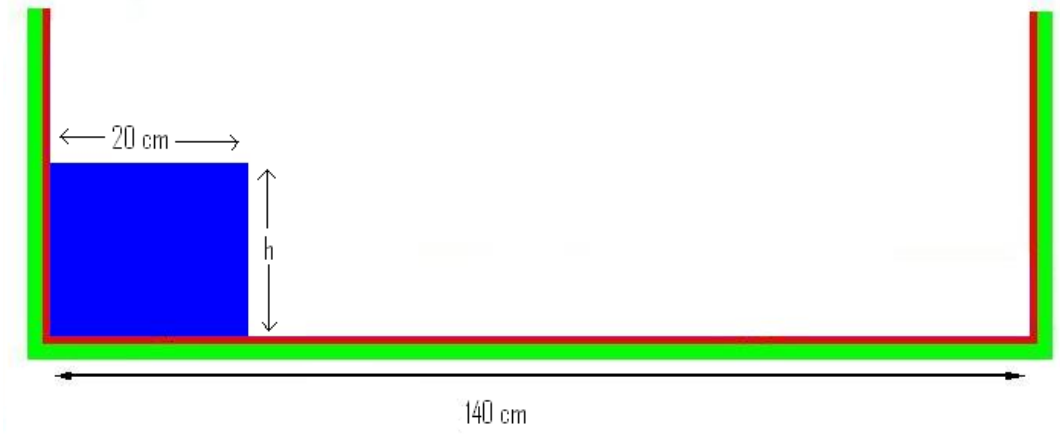


Figure 4.23. Set up of complete dam break onto dry bed without obstacle. *h* is the height of the dam. The colours are coded to represent the different particle types.

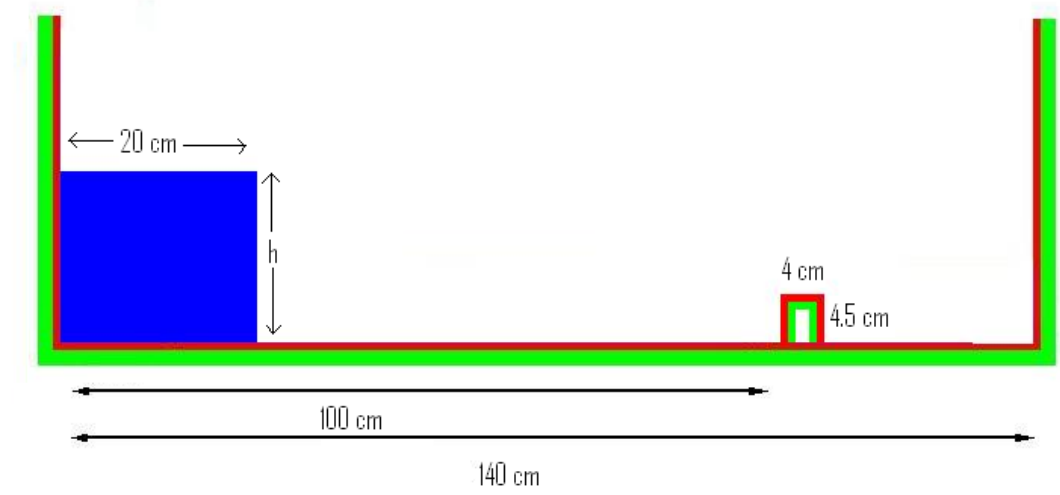


Figure 4.24. Set up of complete dam break onto dry bed with obstacle. *h* is the height of the dam.

Figure 4.25. Picture of the experimental apparatus with water drained out. The gate is partially open near the left hand side.

Figure 4.26. Picture of the experimental apparatus with water drained out taken from behind the high speed camera.

The camera used was a Nanosense Mk 111 made by Dantec Dynamics and controlled by a laptop computer with the program MotionPro Xstudios. For these experiments the camera was used with a 150μs exposure, no gain (signal amplification) and a 600 frames per second recording rate. To improve contrast against the background a few drops of potassium permanganate solution was added to the water and the apparatus was illuminated from multiple angles by spotlights.

Whilst obvious comparisons can be drawn between this experiment and ones covered in the previous sections there are a couple of important differences. The scale of the set up has changed down to 20cm of water rather than 2m and the relative channel length has been

increased. Several researchers have spent time studying dambreaks using the SPH method where the channel length is approximately 2-3 times either the length or height of the initial fluid. However there are very few examples in the literature where longer channels have been studied. Presented here is a study of dambreaks and wave propagation down a channel up to 7 times the width of the water column and up to 9 times the initial height. This presents a step up in the difficulty of the problem to be solved by the code.

4.3.1 Dambreak 1

Beginning with a simple complete dam failure onto a dry bed with no obstacle down the channel (as seen in figure 4.23), the position of the surge, depth at left and right walls are tracked. The sound speed needs to be adjusted down accordingly. For this experiment/simulation h was chosen to be 15cm. It was decided to run this simulation in both Hydra and SPHysics multiple times and compare against experimental values. The Experiment can be seen in figure 4.27.

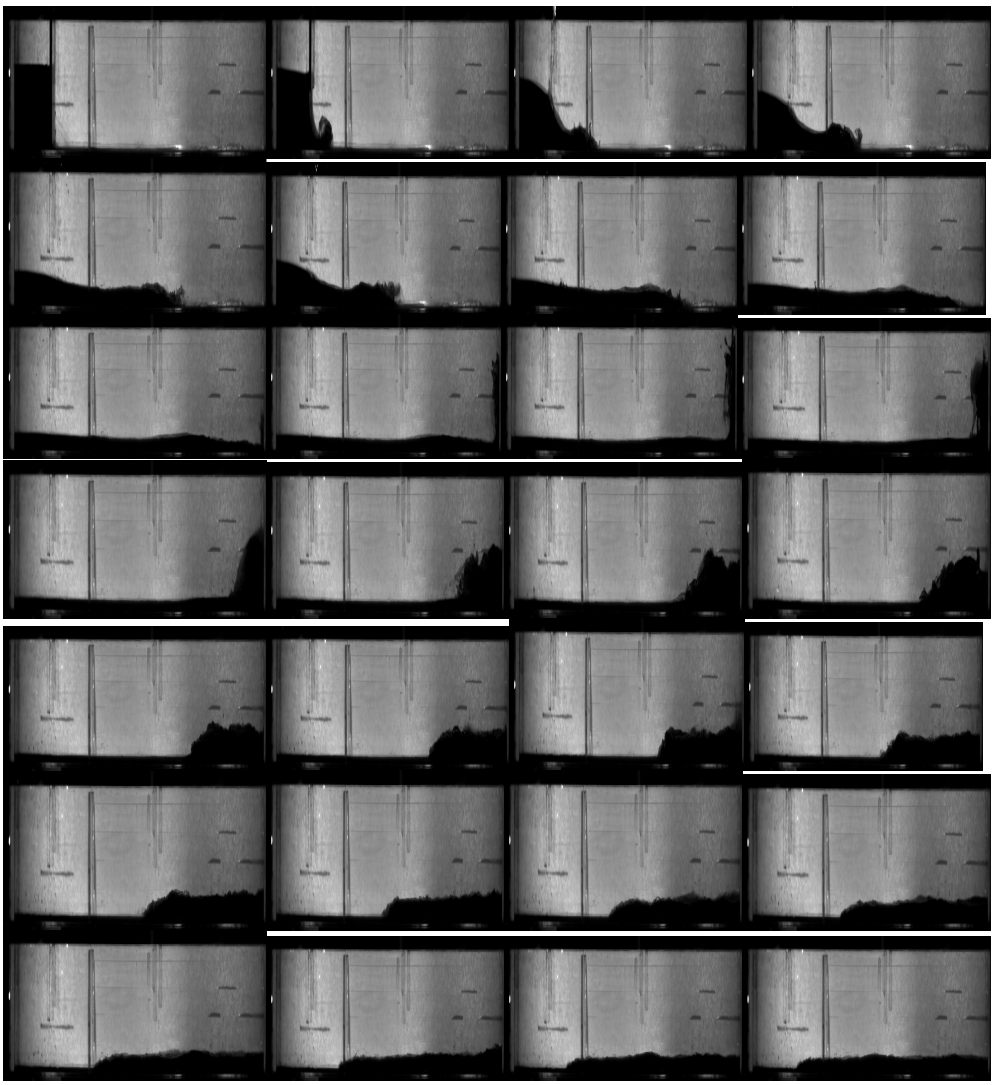


Figure 4.27. Snapshots of the experiment from beginning to end. Images every 0.1 seconds.

Markers at known locations were placed on the side of the tank and the size of the tank itself was precisely known. Each frame contained just the entire length of the tank. This made it possible to calibrate on-screen measuring software to accurately determine the location and depth of the water at any point and within any frame of the experiment. This was helped by the contrast generated by the lighting setup and the solution added to the water.

Hydra is used to create the domain and lay down particles exactly as in previous sections. Beginning with the tracking of the surge as it travels down the channel in Figure 4.28.

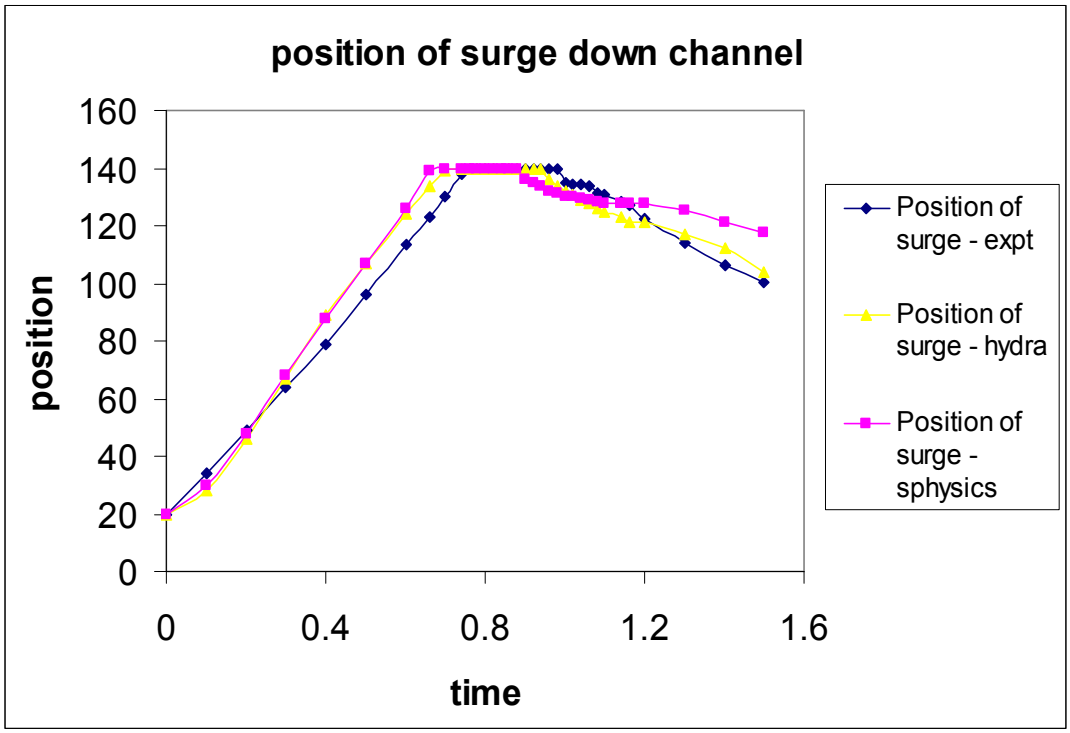


Figure 4.28. Tracking of the position of the wave surge down the channel in Hydra, SPHysics and Experiment.

In agreement with previous findings and the observations of previous researchers (e.g. Jones and Belton, 2006; Colagrossi and Landrini, 2003) the simulated water progresses down the channel slightly faster than the experiment hitting the right wall 0.05 seconds before it. This is the case for both SPHysics and Hydra. Both simulation codes also show the water rising up the wall and then collapsing back into the main flow before the experiment but this is expected as they struck it earlier. The total amount of time spent climbing the wall is equal in all three lines. With regards to the rest of the graph and with the free surface shape in general agreement is good between SPH methods and experiment though SPHysics shows some signs of slowing after it has rebounded off the wall.

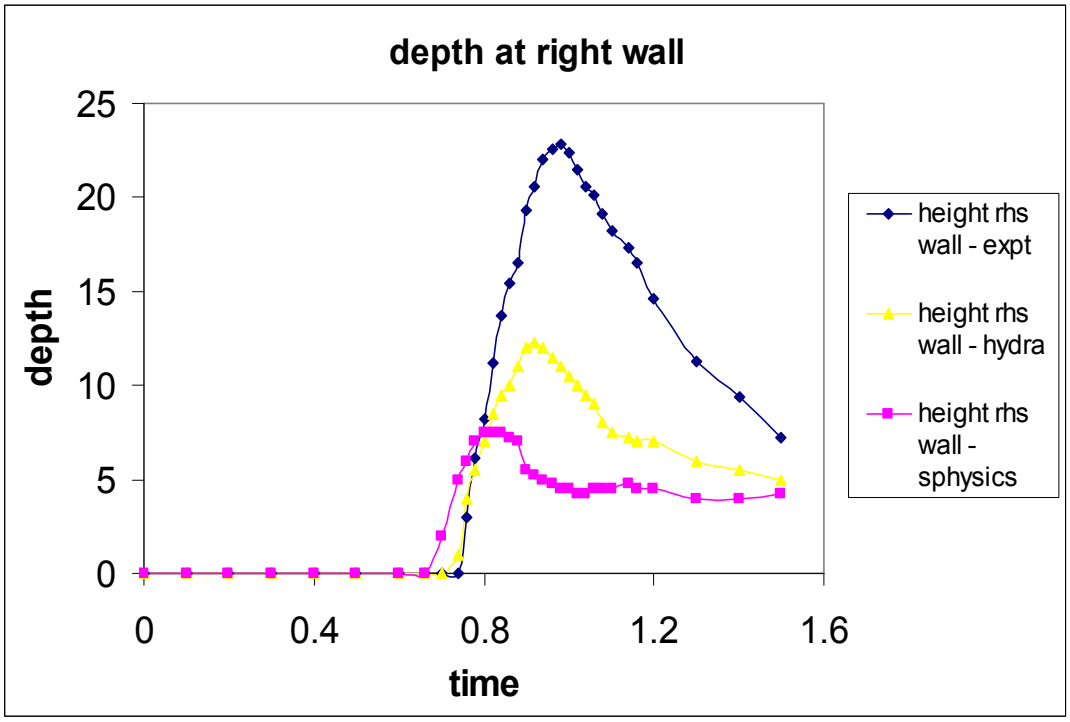


Figure 4.29. The depth at the right hand wall in Hydra, SPHysics and Experiment.

In figures 4.29 and 4.30 we look at the height to which the thin sheet of water reaches up the far wall. As can clearly be seen by the graphs both numerical methods fall far short of the target. After the long channel length there is simply not enough energy left in the particles to climb high enough up the wall. The overall shape and the manner in which the sheet is formed and then curls back down into the flow is very good in both codes but it is simply too low.

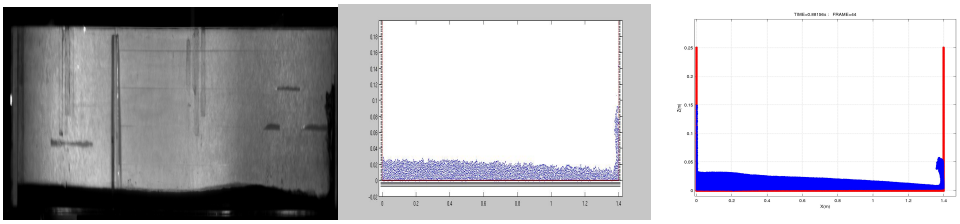


Figure 4.30. Snapshots of the results at t=0.88s for experiment, Hydra and SPHysics.

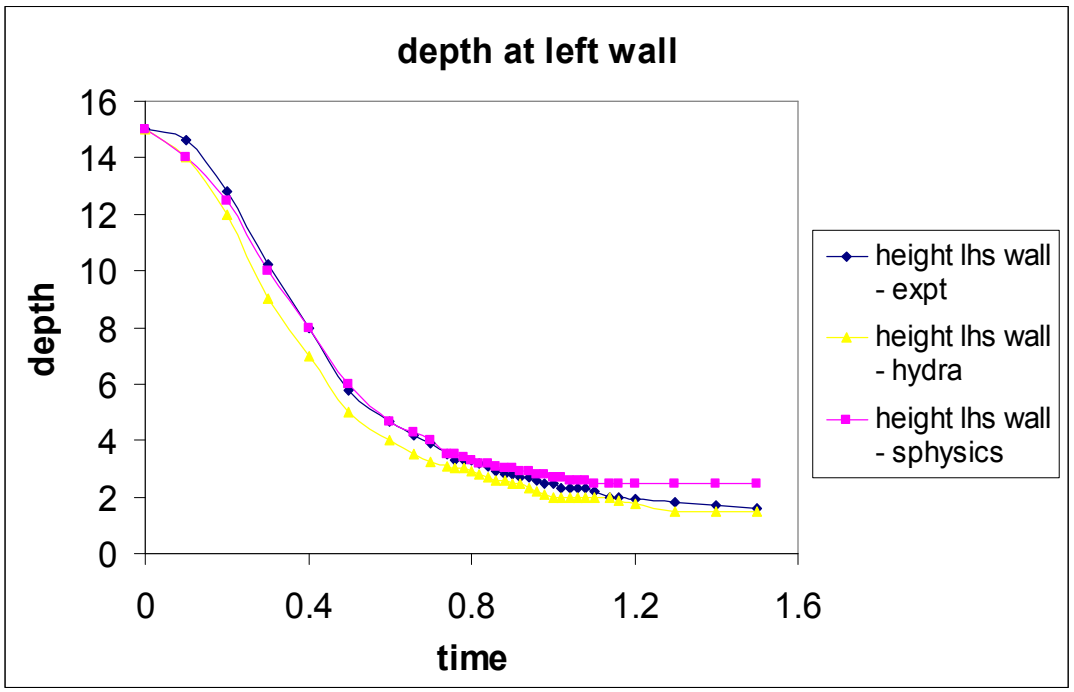


Figure 4.31. The depth at the left hand wall in Hydra, SPHysics and Experiment.

Much better agreement is seen between the three results on the left hand wall in Figure 4.31. There is a slight overestimation of the initial drop speed in Hydra but after 0.8 seconds there is very little to distinguish the lines. The depth to which SPHysics settles to is overestimated though this may in part be due to measuring errors as the visualisation routines which are contained within SPHysics produce large particles.

In order to try and explain the poor performance of the codes in predicting the height of the run up of the right wall the experiment was repeated using different particle numbers. Hydra was used with every property the same except for particle number; 7500, 15000 and 24000 were used. The results of the resolution comparison can be viewed in figure 4.32 and 4.33.

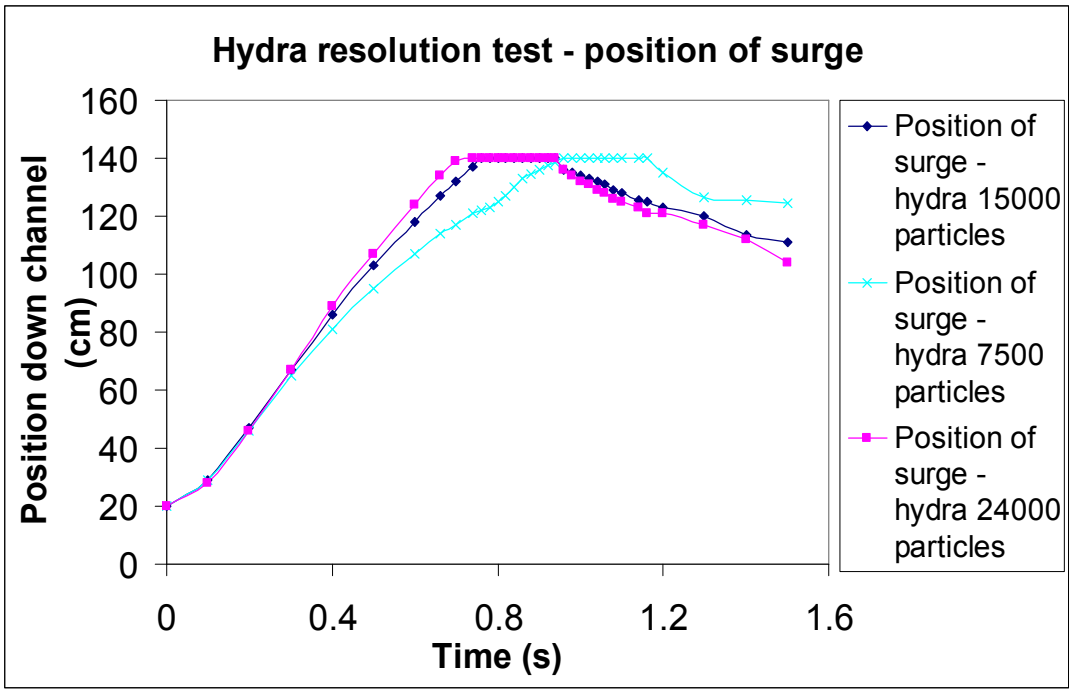


Figure 4.32. Comparison of surge position for various particle numbers in Hydra.

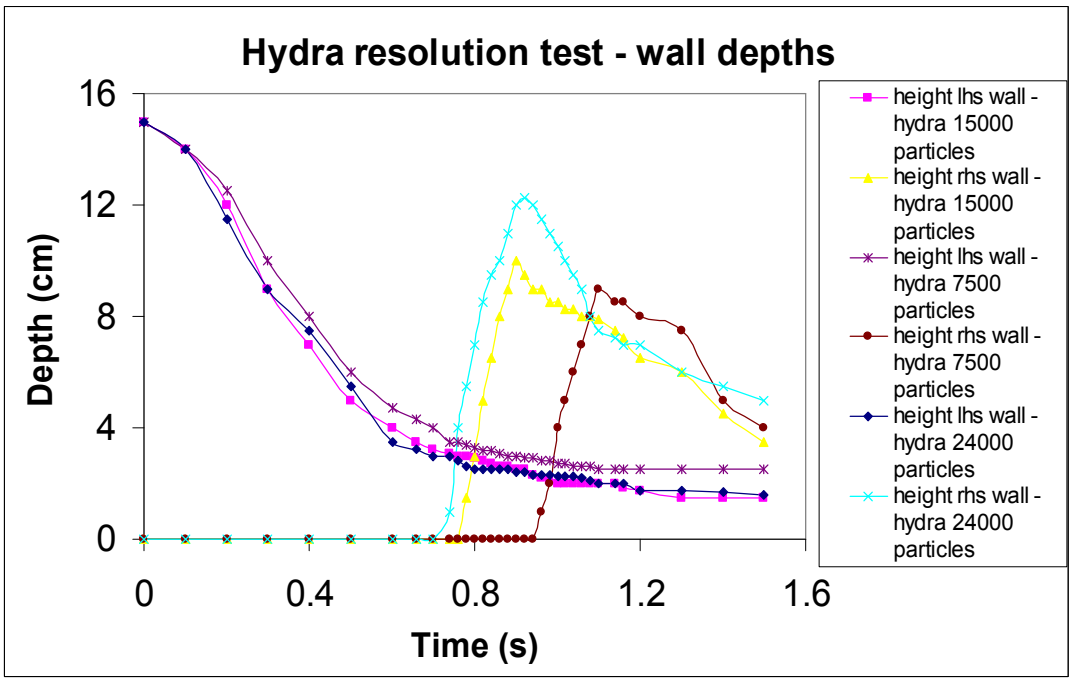


Figure 4.33. Comparison of left and right wall depths for various particle numbers in Hydra.

Left hand wall depth is affected little by particle number. However the same cannot be said for right wall and position of surge. There is a clear improvement in accuracy with increasing

particle number. Whilst some increase in accuracy may be expected there is too large a difference to be easily explained away. This is especially true between the 7500 and 15000 particle cases. Observing the actual simulation shows clearly that as the water surge approaches the end of the tank (after ~1.2m) it slows dramatically. This is borne out in the graphs. With momentum being reduced by friction along the bottom layer of boundary particles it is only natural that the height to which the wave reaches up the right wall is correspondingly reduced as well. The real water seems to slip across the bottom of the tank more easily than the SPH particles. This is entirely possible due to the nature in which the boundary particles apply friction by effectively generating a retarding force on any particle within a set range. The magnitude of this force also increases inversely with distance toward the boundary particle. When the water is deep the majority of fluid particles are not in contact with the boundary particles but when the water goes shallow, e.g. at the tip of a surge, a large proportion of the particles are within range of the bottom boundary particles so the amount of friction is much greater. As long as water can be kept deep enough throughout the domain there will be no problem. This is made easier by using additional particles, but of course thereby increasing computational cost. In this type of simulation there will always be a shallowing of the average water depth until the end of the tank is struck. However if water is too shallow, errors will be introduced into the simulation because there will simply not be enough particles. If a particle is contained within water so shallow that its smoothing length is of the same order as the water depth then a proper density estimate will not be made. As was stated in Chapter 2, good density estimates are crucial to the accuracy of an SPH based solution. Using this as a base it is possible to estimate a maximum channel length that can be accurately simulated using an SPH based code based on particle number, smoothing length and initial conditions. This may be useful because it will enable a researcher to predict how many particles are required for a simulation of given length before it is carried out. The author does not believe that this is a problem solely for Hydra but is in fact a theoretical limit of the SPH method itself in the modelling of long open channel flow.

Based on the observations of the simulation above and attempts to model it with a variety of particle numbers and SPH theory the following hypothesis has been devised. It was assumed that throughout the channel the shallowest point (i.e. the toe of the surge) must be at least equal to twice the smoothing length (one above and one below the particle) in order for a proper estimate of density. As neighbours are chosen from all directions the depth required in water Hydra to get a good density estimate should be equal to $2 \times \sqrt{\frac{32}{\pi}} = 6.38$. The initial conditions of a dambreak rely on particles being laid down horizontally and vertically based on a certain number of particles per unit length. In the above simulation this length is equal to the initial width of the water, i.e. 0.2m. Taking the number of particles in this length and multiplying by the number in the height and then dividing by the minimum number allowed (6.38) gives the maximum number possible in the length. Divide this number by the average of the number of particles in the initial height and length and the result is the maximum number of unit lengths that the channel can be with that particle number. However experimentally it was discovered that this result is twice the length that can actually be simulated accurately. The minimum depth of the water toe must be twice that expected so that there are particles in the flow which do not see the surface or channel bed. Looking at the cases above with 15000 and 7500 particles the maximum channel length can be predicted.

7500 particles	15000 particles
100 x 75 = 12.76 x 587.8	141 x 106 = 12.76 x 1171.3
587.8 / 87.5 = 6.72	1171.3 / 123.5 = 9.48
6.72 x 0.2 = 1.34	9.48 x 0.2 = 1.9

With 15000 particles the maximum channel length is 1.9 metres which is fine for the 1.4 metre channel presented. However the predicted maximum length allowed for 7500 particles

is only 1.34 metres. Interestingly this is near the point at which the speed of the surge takes a real dip and the simulation fails (though admittedly it is poor for sometime prior to this). A formula can therefore be proposed to predict the maximum channel length for any such simulation.

$$\text{Initial } P_w \text{ x Initial } P_h = 4 \times \sqrt{\frac{P_s}{\pi}} \text{ x Max } Pw$$

$$MaxP_w \div \frac{InitialP_w + InitiaP_h}{2} = \text{Channel multiplication factor}$$

Max channel length = Channel multiplication factor x Initial W

Where P_w is number of fluid particles in width, P_h is number of fluid particles in height, P_s is number of particles within smoothing length and W is the initial width of the water column.

A simulation should not be attempted unless the number of particles and the initial conditions can cope with the channel length. By this formula doubling particle number alone only increases the maximum channel length by the square root of 2. This is a maximum length and for peace of mind a greater number of particles would ideally be chosen than the minimum predicted by this formula. A theoretical limit such as this cannot be found calculated in any of the literature but no simulations presented by other authors have utilised channel lengths long enough to breach the limit.

4.3.2 Dambreak 2

After the poor performance of the numerical methods in predicting the wall height in the previous simulation a test was set up to show that the channel length/particle number is the

important factor not the change in size of the dam. The same set up without any obstacle was created but with h = 12cm. The right hand side wall was moved closer to the left shortening the channel to 80cm. The surge position and the water height at the right hand wall were measured in a simulation using 24000 fluid particles. The surge position is plotted in figure 4.34.

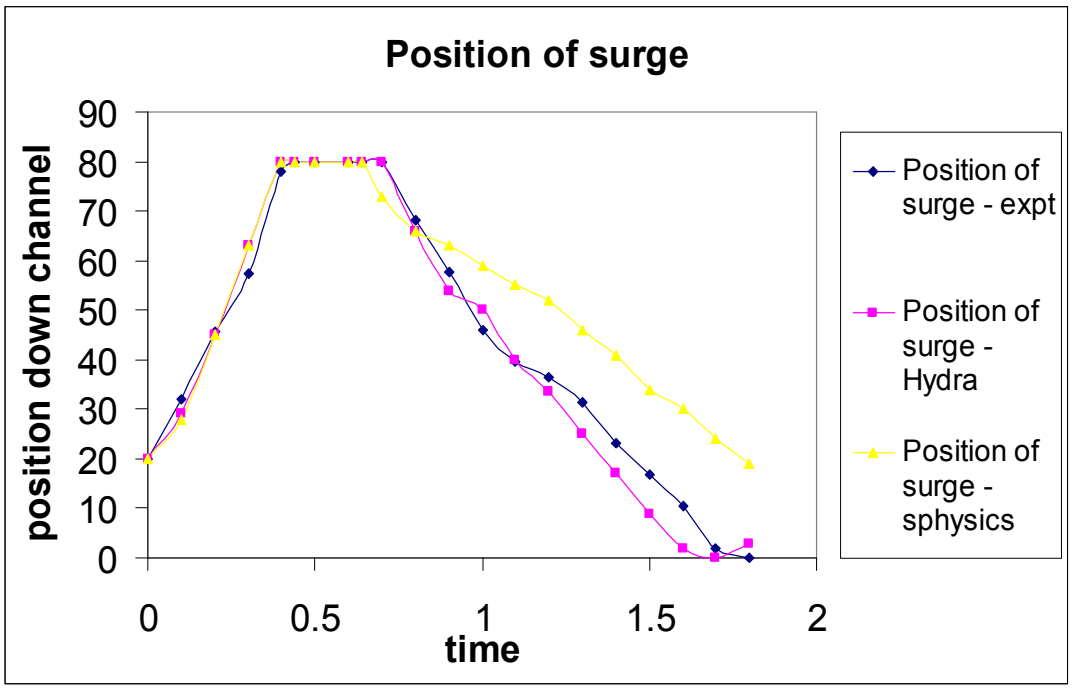


Figure 4.34. Tracking of the position of the wave surge down the channel in Hydra, SPHysics and experiment.

The timing of the surge is closer to before with only the slightest difference between all three. SPHysics loses momentum after the wave falls back into the main flow and it slows too much but Hydra performs well. Moving onto measuring the height of water up the right hand wall in figure 4.35.

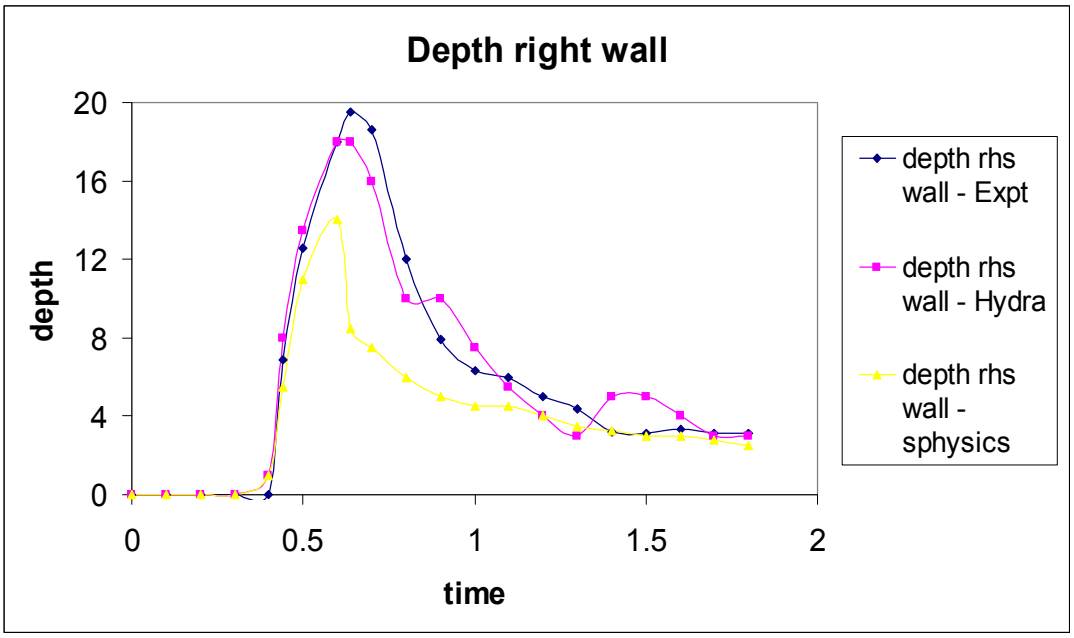


Figure 4.35. The depth at the right hand wall in Hydra, SPHysics and experiment.

There is still some under prediction present in the numerical methods but the results are much improved over before. With a shorter downstream channel the channel length limit is not approached here and the results are much improved.

4.3.3 Dambreak 3

Introducing an unmovable obstacle into the channel at 1m down stream provides a similar test to before whilst progressively increasing the complexity of the problem further. As well as investigating properties of the flow as before this provides an opportunity to investigate how a code handles striking and flowing over an object rather than a wall which it cannot go over. In this experiment h = 15cm and 16000 particles were used in both Hydra and SPHysics. Parameters were kept the same as before. CFX was also used to simulate this experiment using the same parameters and schemes as in the previous section; 16700 cells were used. Several properties of the resulting flow were measured and compared against the experimental results. Beginning with the position of the surge down the channel Figure 4.36.

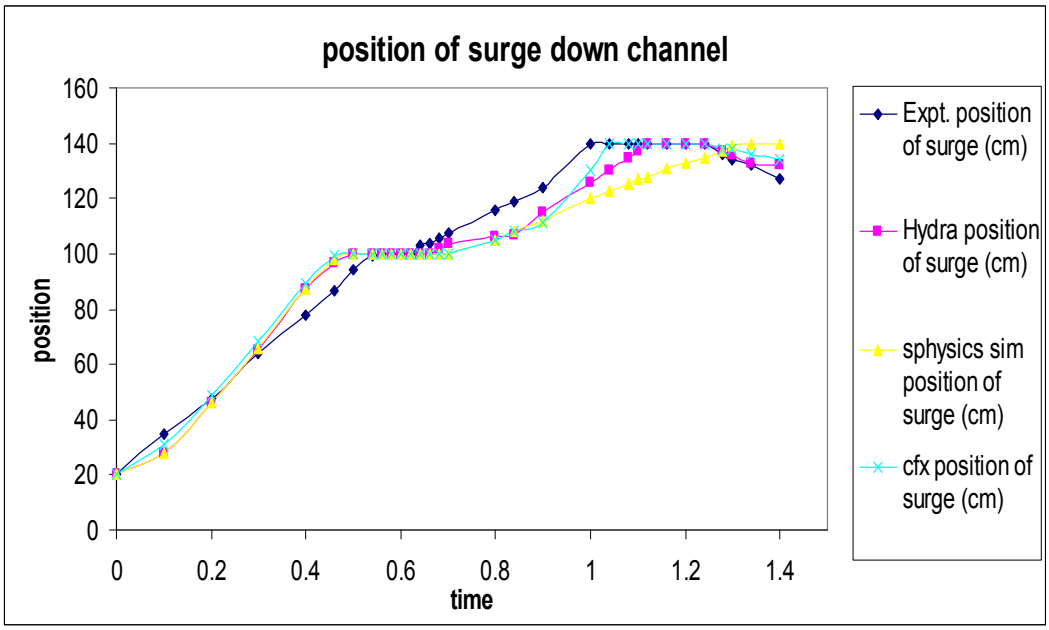


Figure 4.36. The position of the surge down the channel in Hydra, SPHysics, CFX and experiment.

All three numerical methods can be seen to slightly over predict the surge velocity up until the barrier strike at ~0.5s. After striking the barrier however the opposite is true, with the real water moving on to the right wall faster than any of the numerical methods (at 1s compared 1.1-1.2s). In order to better understand these findings the following graph (Figure 4.37), measuring the height of the surge, should be considered together with Figure 4.36.

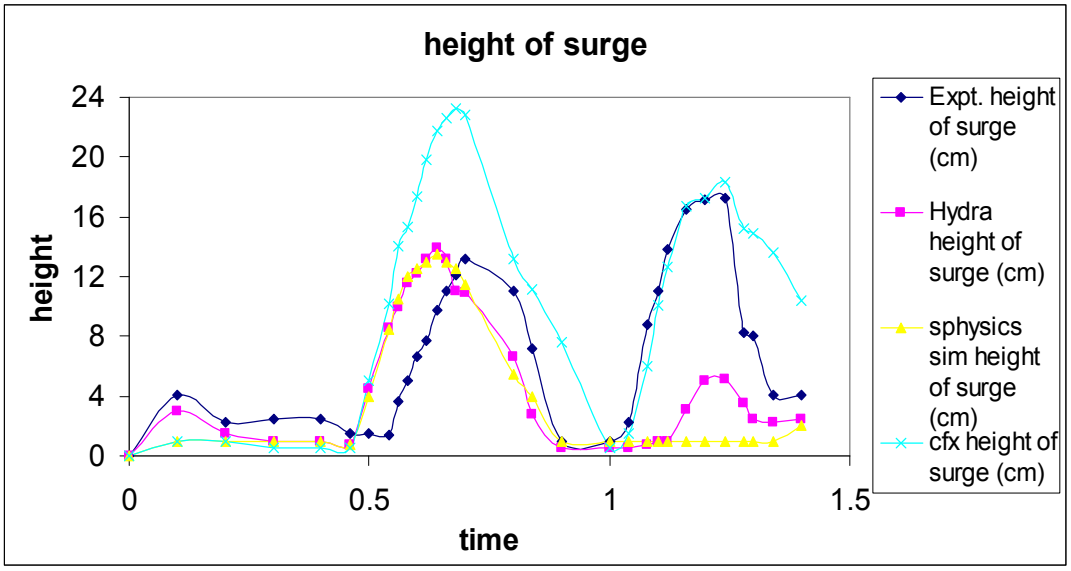


Figure 4.37. The height of the surge in Hydra, SPHysics, CFX and experiment.

The barrier is struck in the simulations just before the experiment. The water is thrown up over the barrier and crashes down over the other side of it before flattening out in both directions before the bulk of the flow hits the right wall and rises up it in a thin sheet. Both SPH codes predict the height and duration of the splash over the barrier very well. CFX can be seen to over predict the height which the water attains by a considerable margin. Despite this neither SPH methods can accurately predict the run up the wall for the second splash. CFX recovers from it's over prediction around the barrier to perform well at the right wall. Close analysis of the flow around the barrier may give an explanation of why this might be.

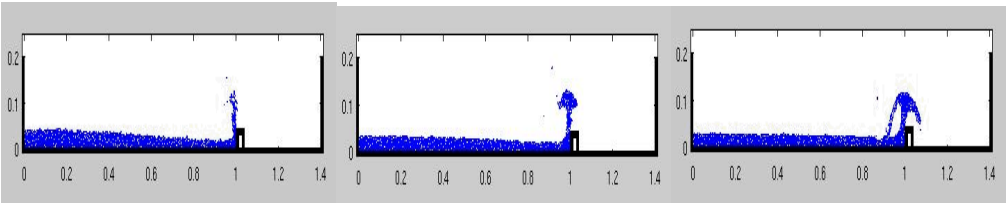


Figure 4.38. Snapshots of the simulation showing the flow of the water as it strikes the obstacle.

The SPH particles travel almost vertically upwards (Figure 4.38.) before forming a mushroom cloud shape. They do not retain much horizontal momentum and many begin to fall back

down in the same line that they rose up in. The particles on the left hand side of the "mushroom" also fall back into the main flow; this does not happen with real water where sideways motion carries the splash over the barrier. The impact of the particles falling back into the main flow causes an interruption in the motion of the water and therefore the particles do not have sufficient momentum to make it over the barrier in sufficient quantities. This also causes the slow down in the simulation seen in the graphs in figures 4.36 and 4.37. There is a possible reason for this in the form of the physics which the SPH code uses to simulate. Each boundary particle exerts a force normal to its orientation on each particle within its range. When the water is sliding along the bottom this means that the force is acting vertically upwards and when the water is climbing the barrier the force is back towards the left of the tank. A good analogy of how the fluid and boundary particles interact is to consider snooker balls striking each other. In agreement with the findings of Crespo, Gomez-Gesteira and Dalrymple (2007) there is a near elastic collision with little energy loss between the particles. This causes the SPH particles to be thrown vertically and lose their sideways motion as they strike the barrier. This happens constantly throughout their contact with the boundary particles and then the particles climb until gravity slows them and they interact with the particles behind them in a similar fashion to a fountain. Looking at some zoomed in images of the experiment in figure 4.39 a different story is told.

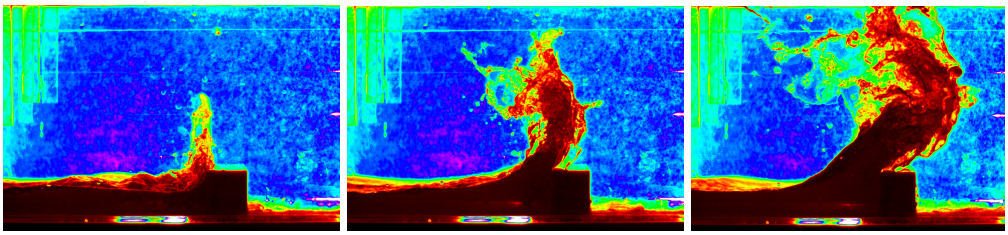


Figure 4.39. Snapshots of the experiment showing the flow of the water as it strikes the obstacle.

As the water strikes the obstacle it rises vertically up in a similar way to the simulation initially but does not experience forces pushing it to the left. Instead there is energy dissipation in the corner of the obstacle/floor. Water is a continuous fluid and strikes in a

softer way than solid balls more akin to a snooker ball striking a soft cushion. There is impact between water and obstacle but the water does not bounce off the solid surface instead it spreads out over the surface. A ramp is formed by water that allows the water behind to be launched over the barrier and continue on its way to the right wall. This is in agreement with observations of the 2004 Boxing Day tsunami where the wave was seen to enter buildings above the ground floor after striking a seawall and being projected diagonally upwards (Dalrymple and Kriebel, 2005). It is possible that a good turbulence model could account for the energy dissipation and ramp forming via eddies and improve the numerical method. CFX does not have particles rather a continuous, albeit discretised, fluid and turbulence is built in to it and it creates a better model of flow over the barrier with more horizontal momentum maintained.

As can be seen in the free surface profile at t = 0.4s at around 70cm down the channel (Figure 4.40) and in the pictures of the different methods at the same time (Figure 4.41), in the experiment there is a jump in surge height just at the toe of the water. In contrast, the simulations all show a neat tapering off of the water surge to a point. The reason for the irregularity in the experiment was due to the gate opening mechanism. This was done manually and as such did not create a clean break, rather a break where the bottom of the dam is opened first. There is also a shear force created in the upwards direction by friction between the moving gate and the water. It is this which causes the water to jump slightly at the beginning of the experiment and this can still be observed some way down the channel. The consequences of this upwards force have been named gate effects.

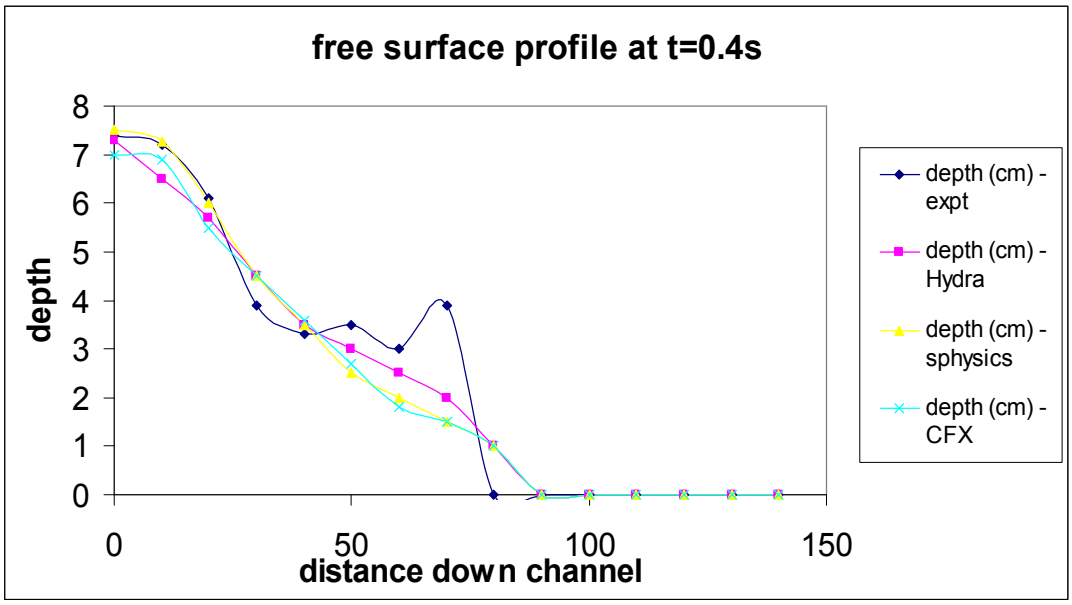


Figure 4.40. Free surface profile at t=0.4s.

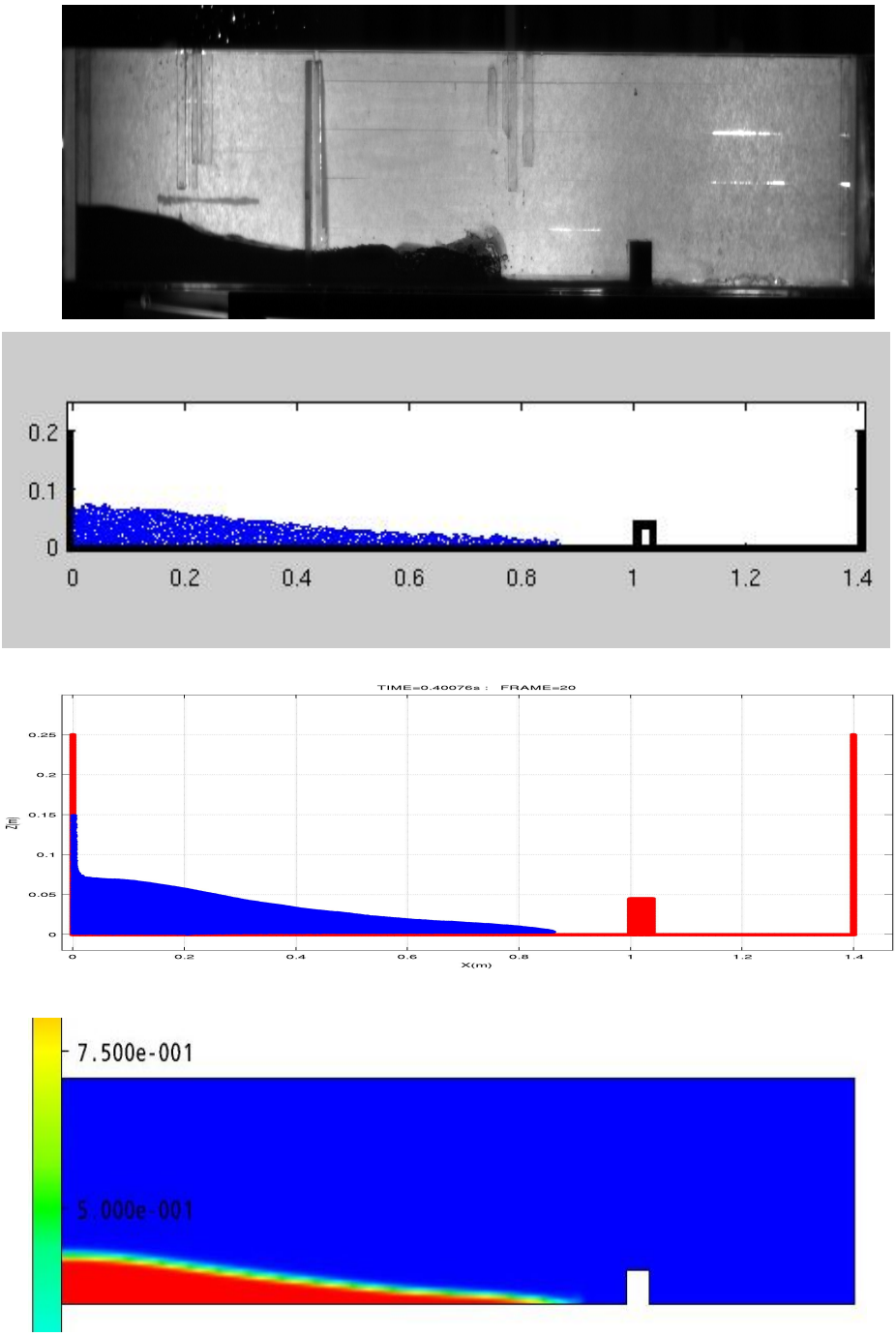


Figure 4.41. Snapshots of all 4 results at t=0.4s.

It was initially believed that after the water had travelled some way down the channel that the gate effects would be dispersed amongst the flow but this may not be the case. Despite great care to ensure that the gate was removed in the same way and at the same speed for each of the experiments (over 20 were carried out) there were small variations in the way that the water travelled over the barrier. The shape of the free surface could be seen to vary

depending upon the gate removal. These findings are in agreement with work by other researchers where the initial flow conditions in a dambreak can substantially influence the wave generated (Liem and Kongeter, 1999). Very similar experiments on fluid flow in Perspex tanks using a gate as a release mechanism carried out by the UK government's Health and Safety Laboratory have found that the gate can have a large effect on the evolution of the system for much longer than expected (private communication with Dr Mat Ivings at HSL). Even small variations could be responsible for changing the shape and height of the wave as it hits the barrier and perhaps even further along. This has consequences for the analysis of the flow after it has struck the barrier. As described earlier one of the problems that the SPH codes had was that the water went almost vertically up when it hit the barrier and lost a lot of its horizontal momentum. This did not happen in experiment. It is possible that this is partly due to gate effects and that if a perfect break could be achieved the water would behave more like the SPH simulations. A problem with this hypothesis is that CFX, even though it over predicted water height over the barrier, maintained a much closer shape to the real water and recovered very well to accurately predict the behaviour after the barrier. If these experiments were to be repeated, improving the gate release mechanism would be highly recommended.

Looking at the free surface profile late on in the experiment (figures 4.42 and 4.43) and comparing it to the simulations gives us insight into how the different codes cope with a longer simulation.

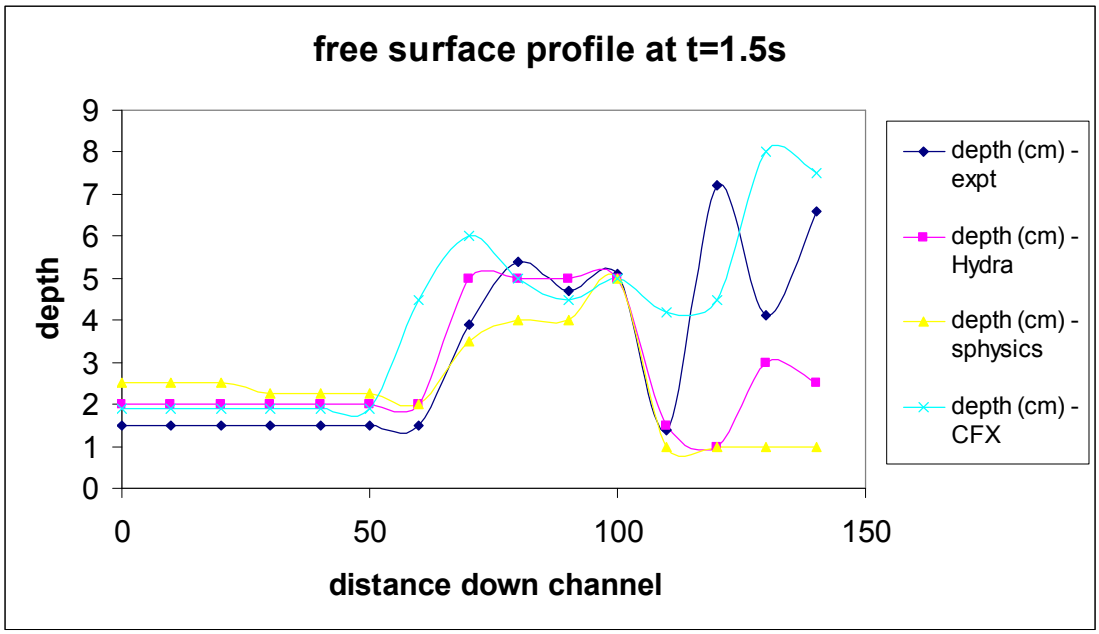


Figure 4.42. Free surface profile at t=1.5s.

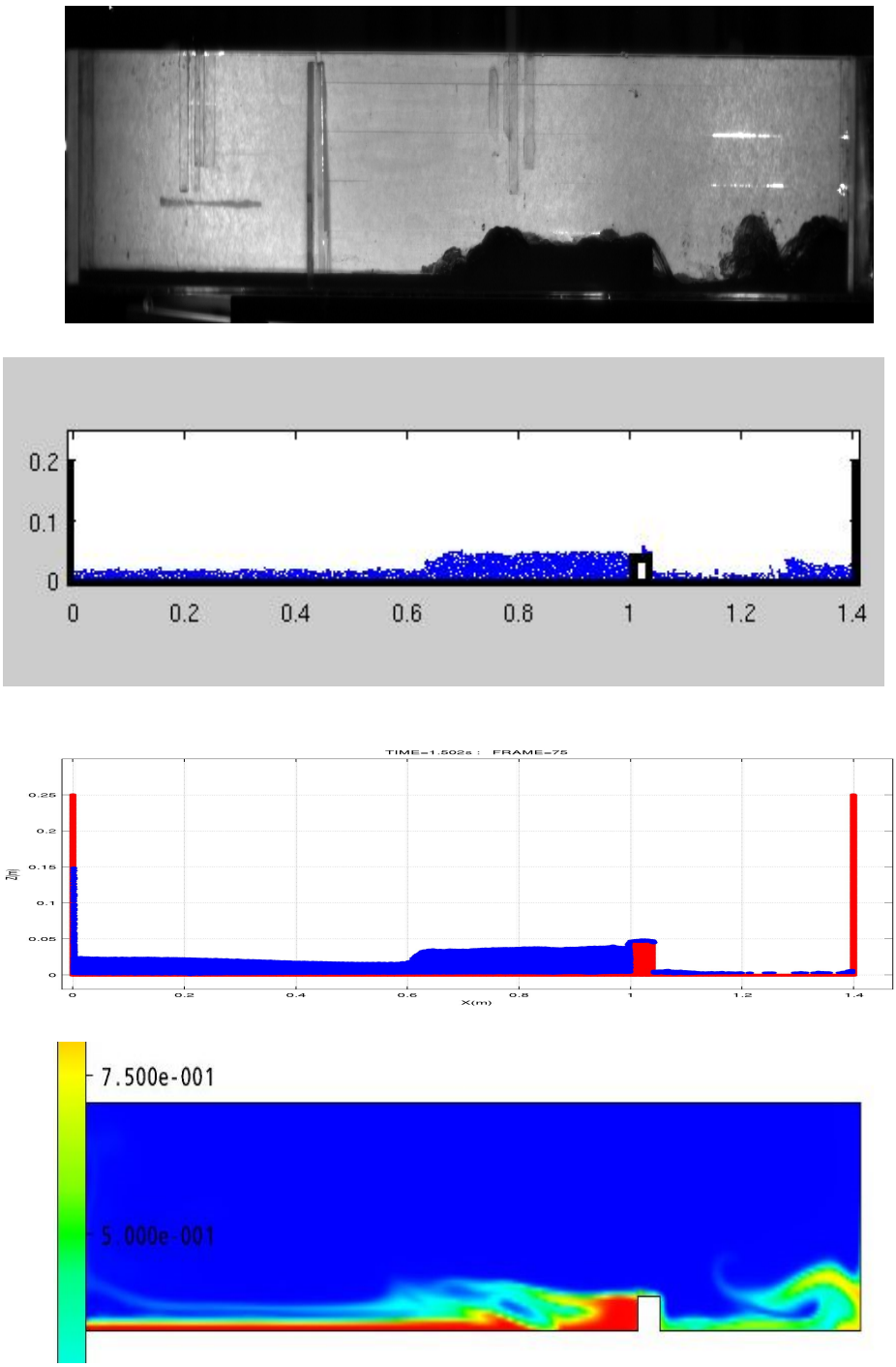


Figure 4.43. Snapshots of all 4 results at t=1.5s.

When compared to the experimental results both SPH codes have modelled up to the 1m barrier well. Due to the problems in maintaining forward momentum over the barrier not enough water has crossed over to the far right hand corner so this is shallower than it should be especially in SPHysics. CFX has for the most part performed well in this test and has a

good amount of water transferred over. However the problems with the resolution of the free surface are present and this makes an exact comparison difficult. This is most noticeable around 0.8-0.9m down the channel with a hole in the water volume fraction noticed in a region which should be filled with water.

4.3.4 Dambreak 4

The same tank was used to perform another experiment. This time it was set up with obstacle at 1m and h = 15cm however the downstream section of the channel has been filled with water to a depth of 3cm turning the experiment into a complete dam break onto a wet bed. A wet bed seemed a natural progression from the previous simulations as it provides a different challenge to a dry bed and more accurately models a dam break into an already existing river channel. The Hydra code was made to place water to a depth of 3cm all down the channel so that the barrier pokes out of the surface by 1.5cm. The gate was shut and the entire system allowed to settle into a glass and then released; 21000 fluid particles were used.

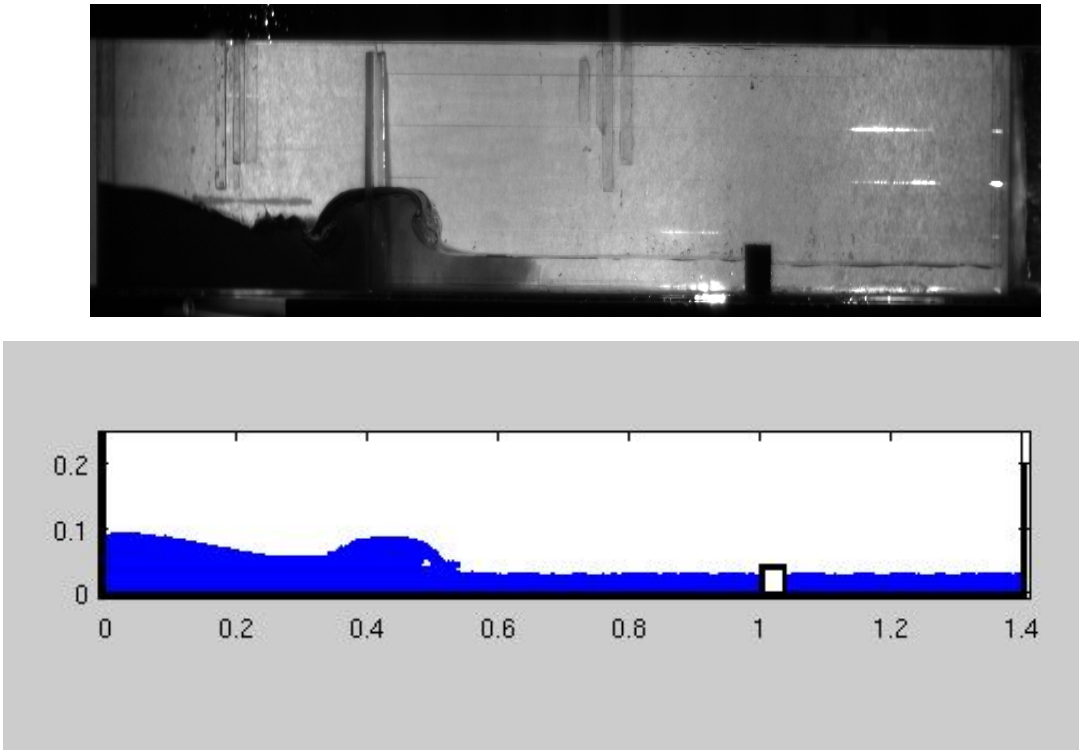


Figure 4.44. Snapshots of the experiment and Hydra at t=0.3s.

Figure 4.44 shows Hydra capturing a breaking wave. These types of phenomena are of great interest to coastal engineers studying wave impacting shorelines and offshore structures.

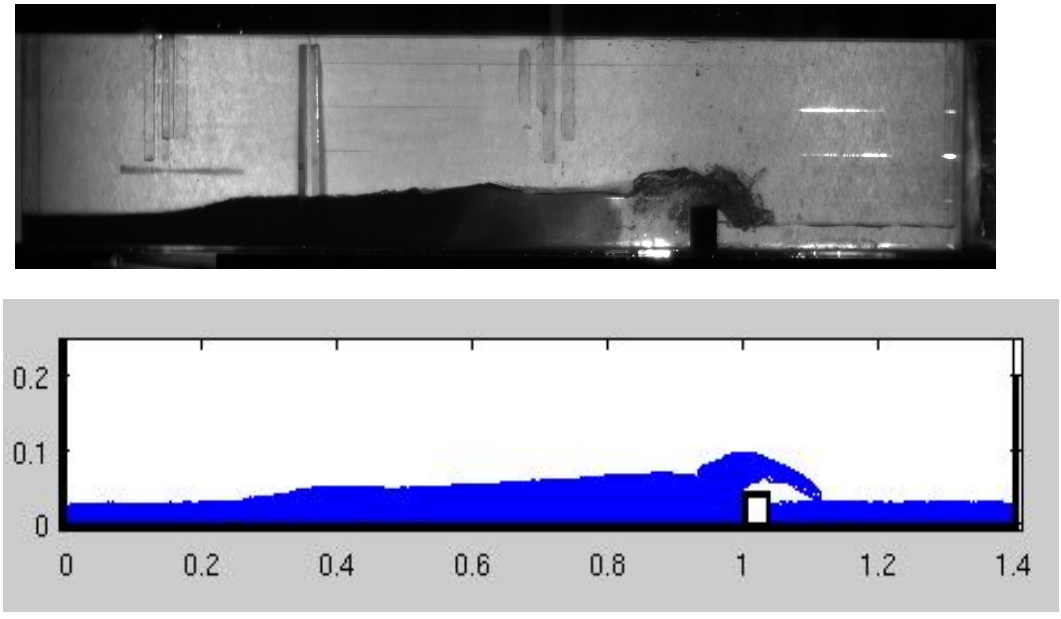


Figure 4.45. Snapshots of the experiment and Hydra at t=0.78s.

In figure 4.45 the wave is just in the process of crashing over the obstacle. The SPH model has lifted over the barrier and travelled slightly further down the channel but has otherwise captured the motion of the water very well.

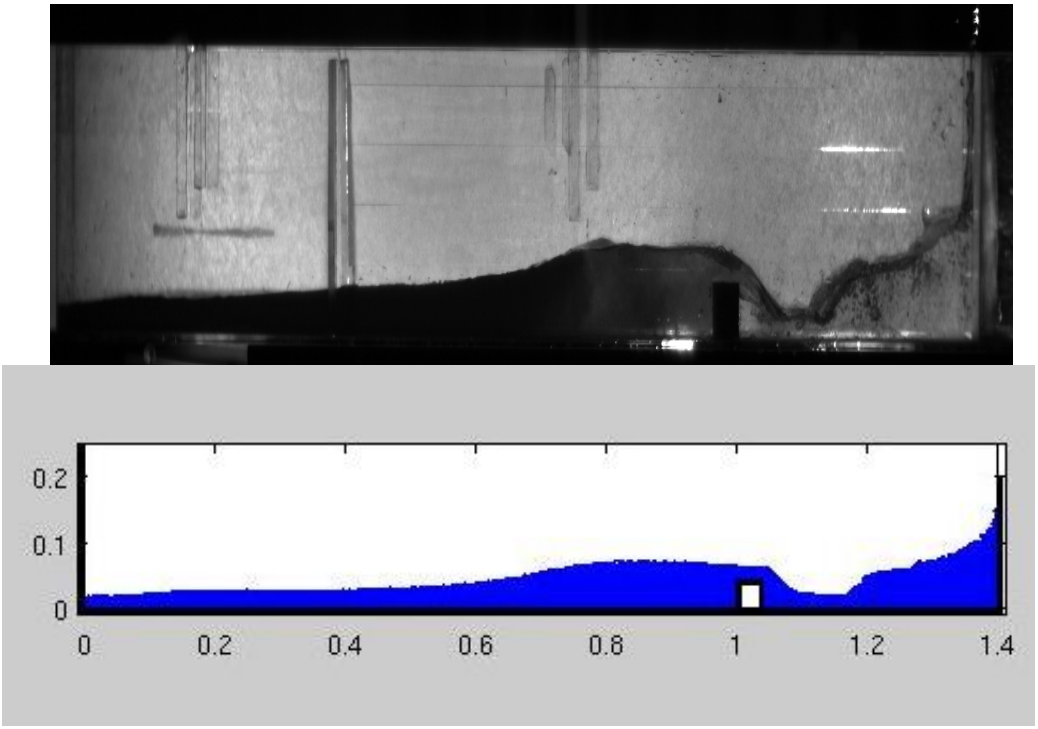


Figure 4.46. Snapshots of the experiment and Hydra at t=1.3s.

Figure 4.46 shows the water has reached the right wall and formed a thin sheet up it. Hydra has maintained good free surface shape throughout. Water continues to pour over the barrier but with a substantially reduced rate. It is interesting to point out a characteristic of the wave generation in the experiment. The water contained behind the gate was infused with the same dye as before but the water in the channel was left clear. After the experiment was finished the water left over the barrier is still remarkably clear with most of the dyed water still in the left half of the tank. The wave does not travel over the surface, but is instead created by the displacement of the water before it. Gravity causes the initial displacement and then the whole channel is pushed to the right so the dyed water never reaches the barrier.

An analysis was carried out looking at the depth profiles of the water at four points down the channel, studying how the depth of water changed over time, in Figure 4.47.

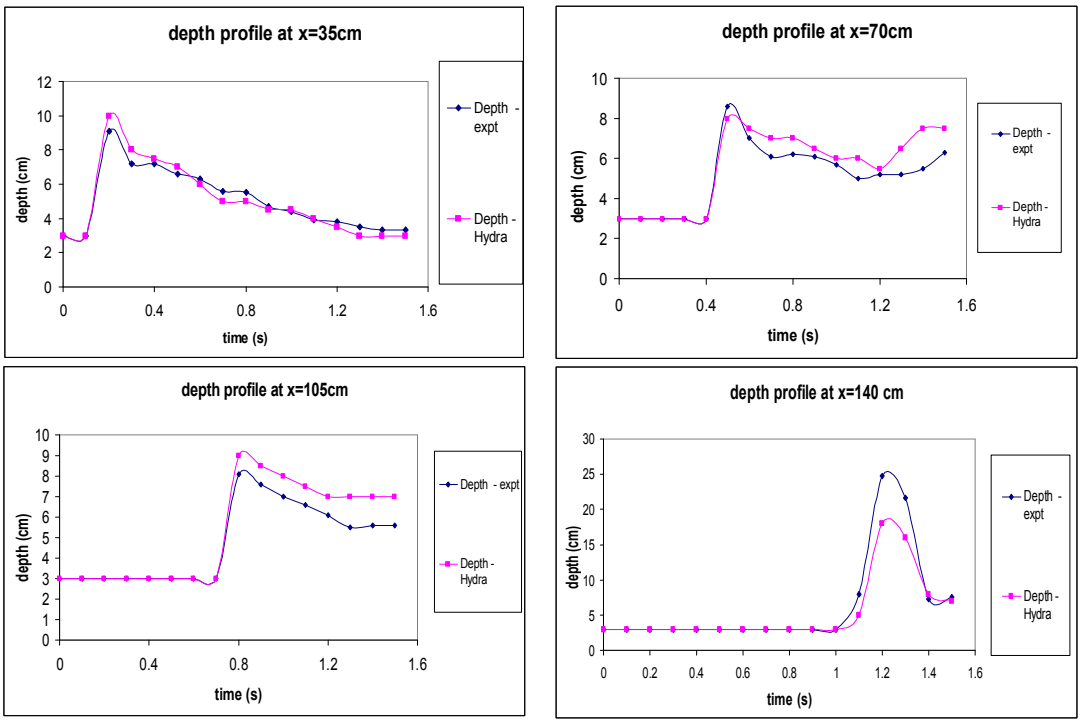


Figure 4.47. Comparison of depth profiles in experiment and Hydra at 35cm, 70cm, 105cm and 140cm down the channel.

The depth of the water as the wave passes is modelled well though with some overestimation especially at 105cm. This is as the wave passes over the obstacle. The opposite is true however in the final graph at 140cm (i.e. the right wall) where the height is underestimated. This may be due to energy loss in the SPH model as the barrier is interacted with. It should also be noted that when a wet bed is present, the characteristic overestimation of wave speed that is seen in numerical methods is not present. This is explored in more detail in the next section.

4.4 Partial Dambreaks onto Wet Bed

It is now time to move on to the investigation of partial dambreaks onto a wet bed. This keeps much of the dam wall intact while removing the lower part of it. This provides a different challenge in the form of the wave produced and the deeper initial water level of the channel. It also provides a simple way of generating a breaking wave travelling down the channel. This was done in considerable detail experimentally by Maxwell (1977) and the comparison is from that same source. This can be used to provide detailed experimental validation of Hydra against independent experiment. In the experiments detailed by Maxwell (1977) the experimental method is quite unusual in the manner in which gate effects are neutralised. Water is filled into the Perspex tank till the required volume is reached. A vacuum chamber is placed on top of the water column and pressure reduced in order to draw water up into the column. This continues until the required water depth is reached in the channel. In the first case detailed this is 7.6cm water depth allowing just the top of the obstacle to be visible above the water surface. The gate holding the dam back is permanently fixed in place which negates any of the spurious gate effects that proved problematic in the previous section (4.3). The water in the channel is never allowed to fall below the bottom of the gate so a constant seal on the chamber is preserved. To generate the wave the pressure in the vacuum chamber is immediately reduced to normal, allowing gravity to create a sudden drop in the column. Maxwell ran several experiments of this nature and four of them are published in his PhD dissertation. The size and speed of the wave generated was controlled by the height of water in the column and depth in the tank. In order to ensure the 2-dimensionality of the waves, the tank was 0.3048m wide. The Reynolds number of the flow was approximately 100 and the experiment was captured by a Bolex 16H 16mm cine camera running at 54 frames per second.

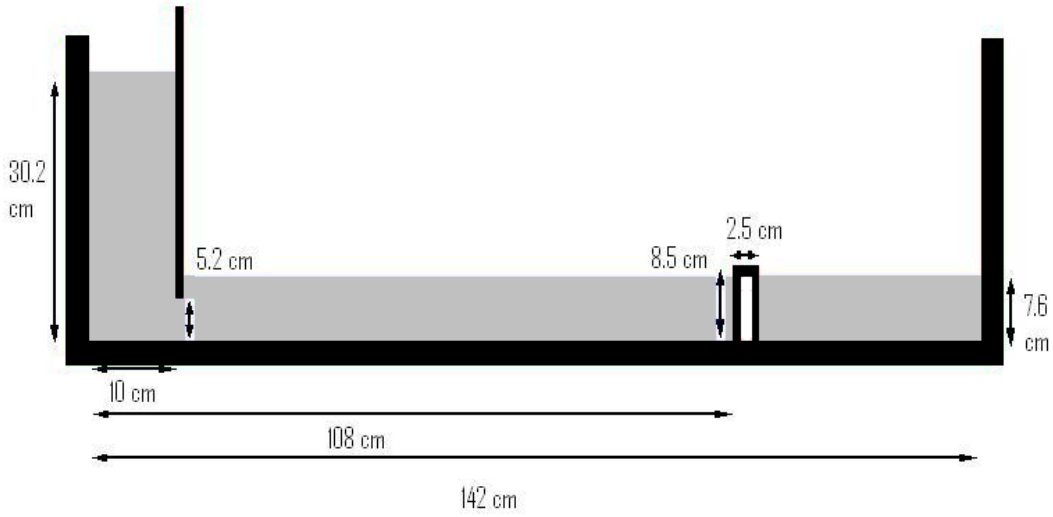


Figure 4.48. Schematic diagram showing the set up and dimensions of the experiment. The obstacle at 108cm is removable (Maxwell, 1977).

Maxwell (1977) measured three properties in detail and therefore these are extracted from Hydra for comparison. These are the height of water in the column, the amplitude of the wave as it propagates down the channel and the position of the crest of the wave as it propagates down the channel. These are illustrated in Figure 4.49.

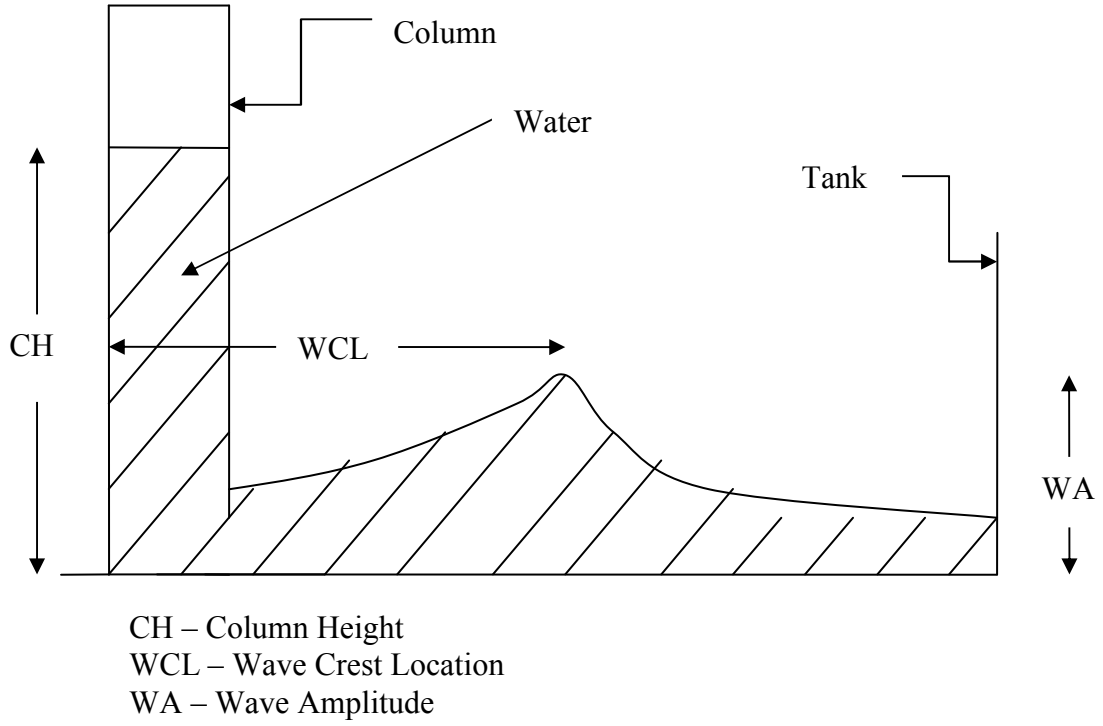


Figure 4.49. Diagram demonstrating the properties to be measured (Maxwell, 1977)

4.4.1 Partial Dambreak 1

The simulation was set up in Hydra in a similar fashion to previous sections and allowed to settle with the gate fully down (to avoid premature wave generation) into a glass state. The gate was made using boundary particles and ghost particles for the majority of it but for the lowermost portion the ghost particles are removed as they may interfere with the water if they come into contact. There were 3 models of this set up made, one with 5000 fluid particles, one with 20000 fluid particles and one with 80000 fluid particles. Based upon the experiences of particle numbers in the previous sections this should allow for a good resolution test and accurate results. In figure 4.50 are some snapshots of the simulation in progress.

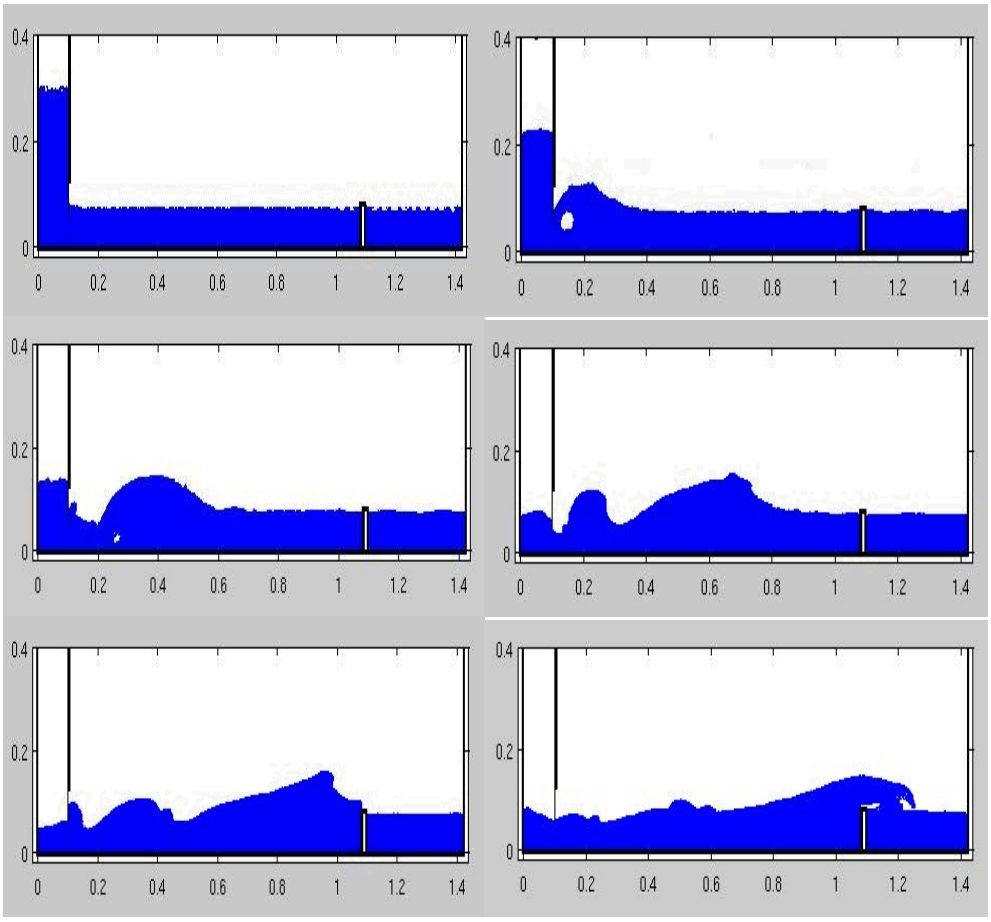


Figure 4.50. Snapshots of the Hydra simulation in progress at times t=0s, 0.2s, 0.4s, 0.6s, 0.8s and 1.0s.

The results of the 3 Hydra simulations are in Figure 4.51 showing how the variation in particle number affects the results.

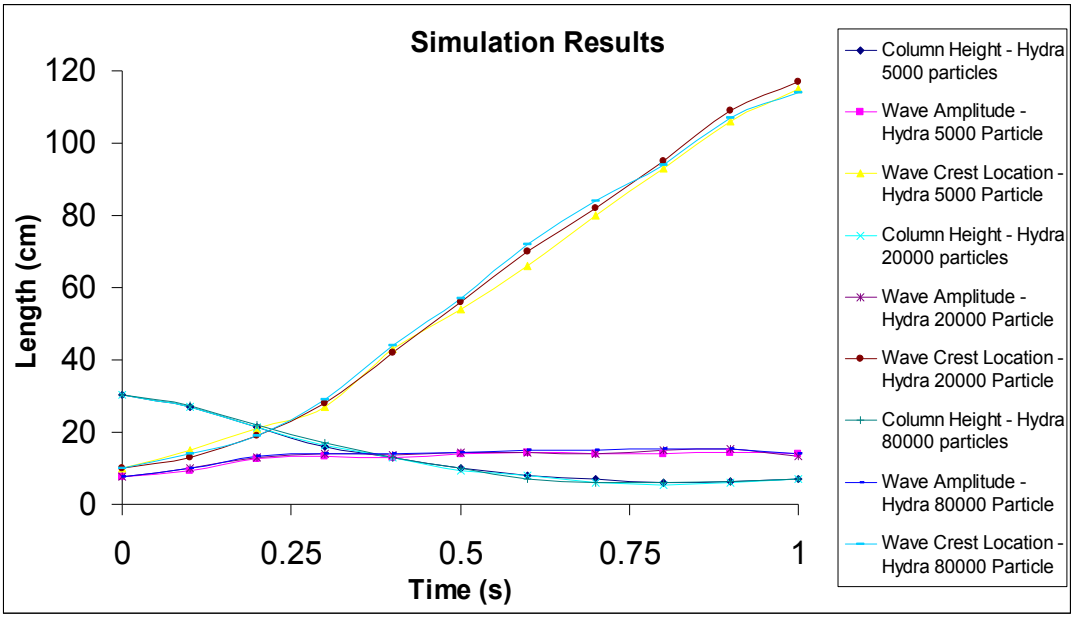


Figure 4.51. Graph showing the simulation results for column height, wave amplitude and wave crest location. All three simulations (5,000, 20,000 and 80,000 particles) are plotted.

The results all agree favourably with each other, indicating that in general the solution is resolution independent. This was mildly surprising for the 5000 particle case which it was expected would struggle with a channel of this length based upon the limit calculations in section 4.3. It has been observed and commented on (private communication with Reza Issa of EDF energy) that 10000 particles is a good benchmark for a minimum resolution level for this type of simulation). The presence of a substantial depth of water on the channel seems to allow for fewer particle numbers to be used than would be required onto a dry bed. The simulation results will now be investigated separately with respect to the three properties being measured and validated against the experimental results of Maxwell (1977), beginning with the column height in Figure 4.52.

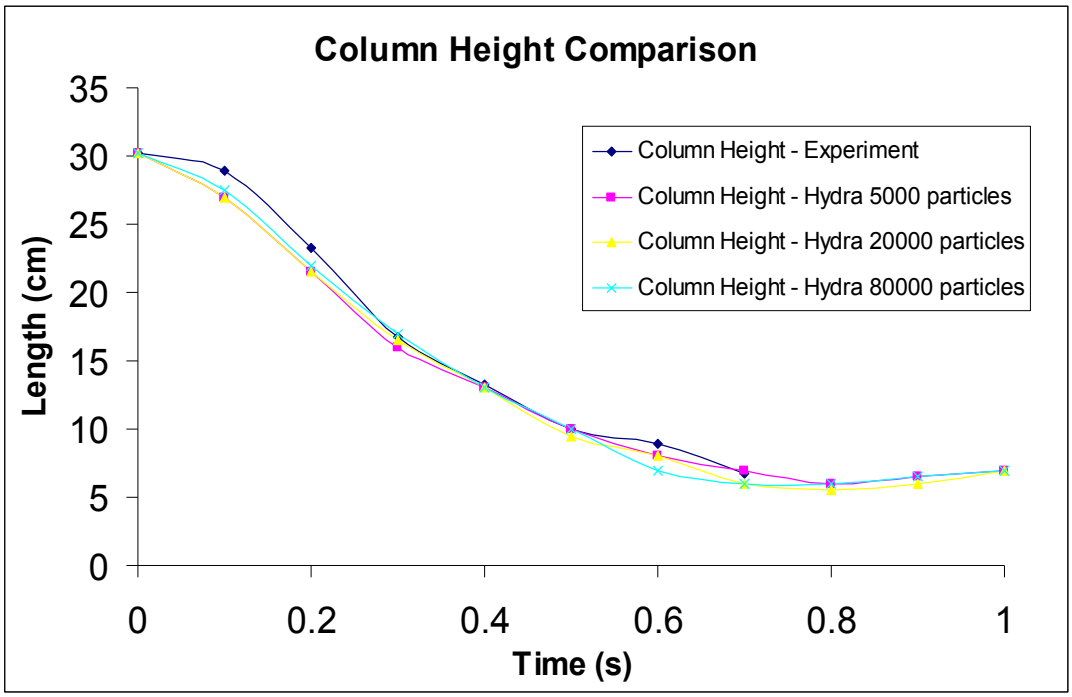


Figure 4.52. Graph comparing the column height between all three simulations and experiment.

A good match overall which is in agreement with the results of the previous sections where water draining down the left hand wall was measured. Simulation seems to drop quicker in the beginning than real experiment but this effect is minor. There is no experimental data available after 0.7s.

Continuing with a comparison of the wave amplitude in Figure 4.53.

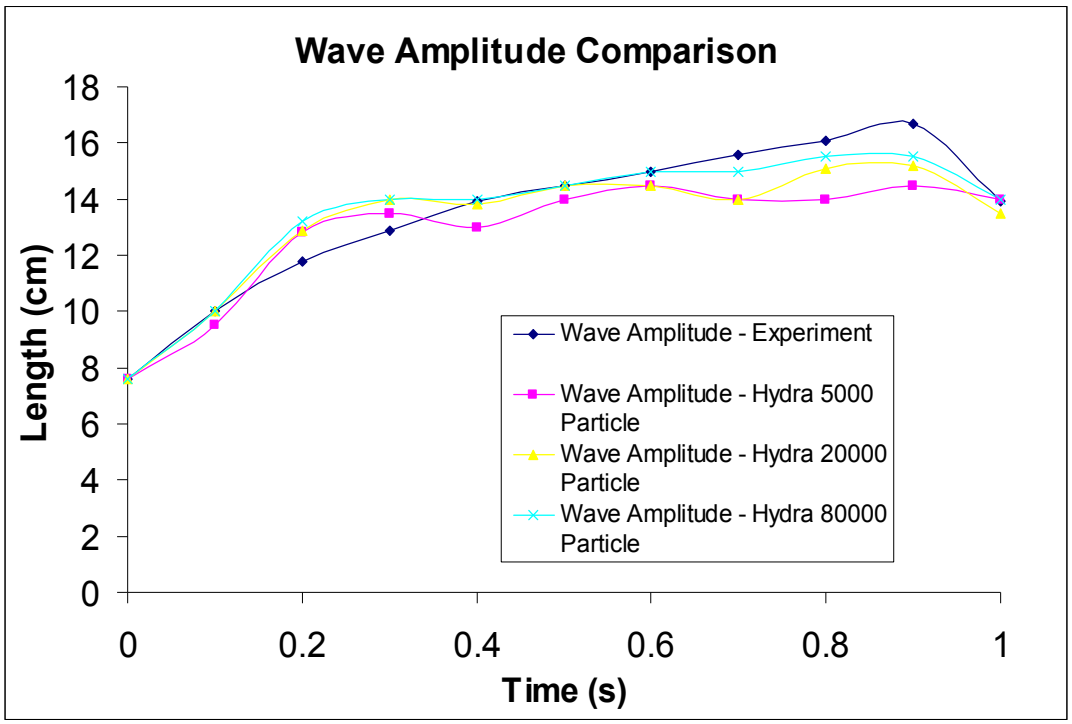


Figure 4.53. Graph comparing the wave amplitude between all three simulations and experiment.

The general trend for the numerical results here is observed to be consistent with experiment but there are discrepancies. The simulations show an overestimate of wave amplitude in the beginning which becomes an underestimation as the system evolves. The experiment curve follows a much smoother path over time until t~0.9s. It is at this point that the wave passes over the baffle which is sticking about a centimetre out of the water. This baffle does not act as one placed on a dry bed which forces the water to be thrown into the air. Instead the wave continues straight over it but its momentum is disrupted and it breaks and splashes down into the pool to the right. The three simulations do follow the same general trend especially in the beginning but it is the highest particle numbered model which is closer to experimental values over the entire simulation. Of the three properties it is this graph which shows the least amount of correlation.

The wave crest location is then identified in detail in Figure 4.54.

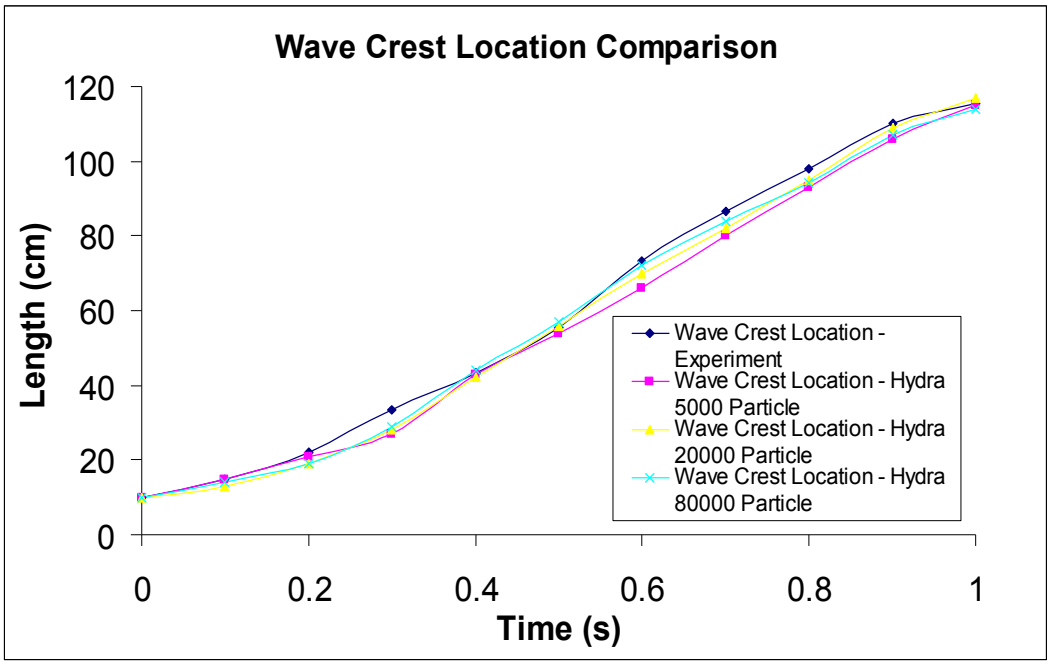


Figure 4.54. Graph comparing the wave crest location between all three simulations and experiment.

Figure 4.54 is essentially a tracker of how fast the wave moves down the channel and produces good results compared to experiment. The experimental results end at t=1s at which time the wave has not yet reached the wall. It is presumed that this is because it proved impossible to identify the wave crest beyond this point as the wave has been broken by the baffle. There are two things to note in this graph; firstly that once again the higher the particle number the closer to experiment the simulation results are. Secondly, that the experimental results indicate that the real water is travelling slightly faster than the simulation. This is in stark contrast to all the previous simulations and observations from previous authors (e.g. Crespo et al, 2007) with the exception of the final simulation of Section 4.3. This was the only previous simulation where a wet bed was used. The most reasonable conclusion to explain this is that the channel length limit explained previously does not apply when dealing with a wet bed – or at least does not apply in the same way. This is probably due to the fact that the moving particles never come across the situation where water depth approaches the smoothing length and as such always have sufficient neighbours. The wave pushes the water in front of it long the channel rather than flowing along the surface.

It is interesting to consider how a grid based code handles a scenario such as this as none could be found in the literature. The same version of CFX as used in section 4.1 was used to allow a three-way comparison between SPH, CFX and independent experiment. There were, however, some differences to the way that CFX discretised the domain to before. 10000 hexahedral elements were used however they were not equally sized or distributed. The elements where no water was expected were increased in size allowing a much finer mesh to be formed around the expected locations of the free surface, major boundary interaction and impact points. The use of a mesh divided into fine and coarse sections allows a higher resolution simulation to be performed while keeping costs down with a smaller overall cell count. Several different sizes of cell were used creating an increasingly finer mesh along the expected line of the free surface of the propagating wave. This could be considered somewhere in between a standard mesh technique and fully adaptive meshing where the mesh alters each timestep. The mesh used and snapshots of the simulation in progress can be seen in Figures 4.55 and 4.56 respectively. The simulation used a 2^{nd} order backward Euler transient scheme, with a 0.005s timestep, a $3.5x10^{-5}$ residual target and a 20 coefficient loop target that was never reached.

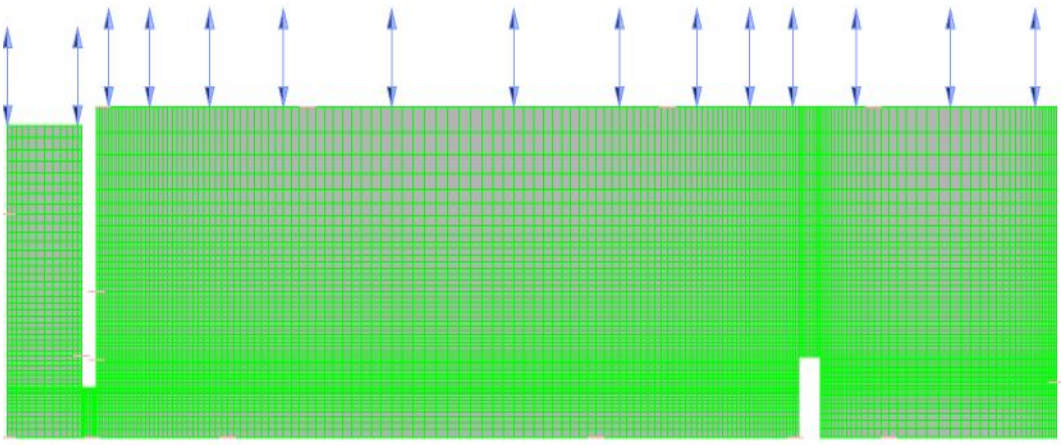


Figure 4.55. Mesh used for the simulation showing the variation of grid size.

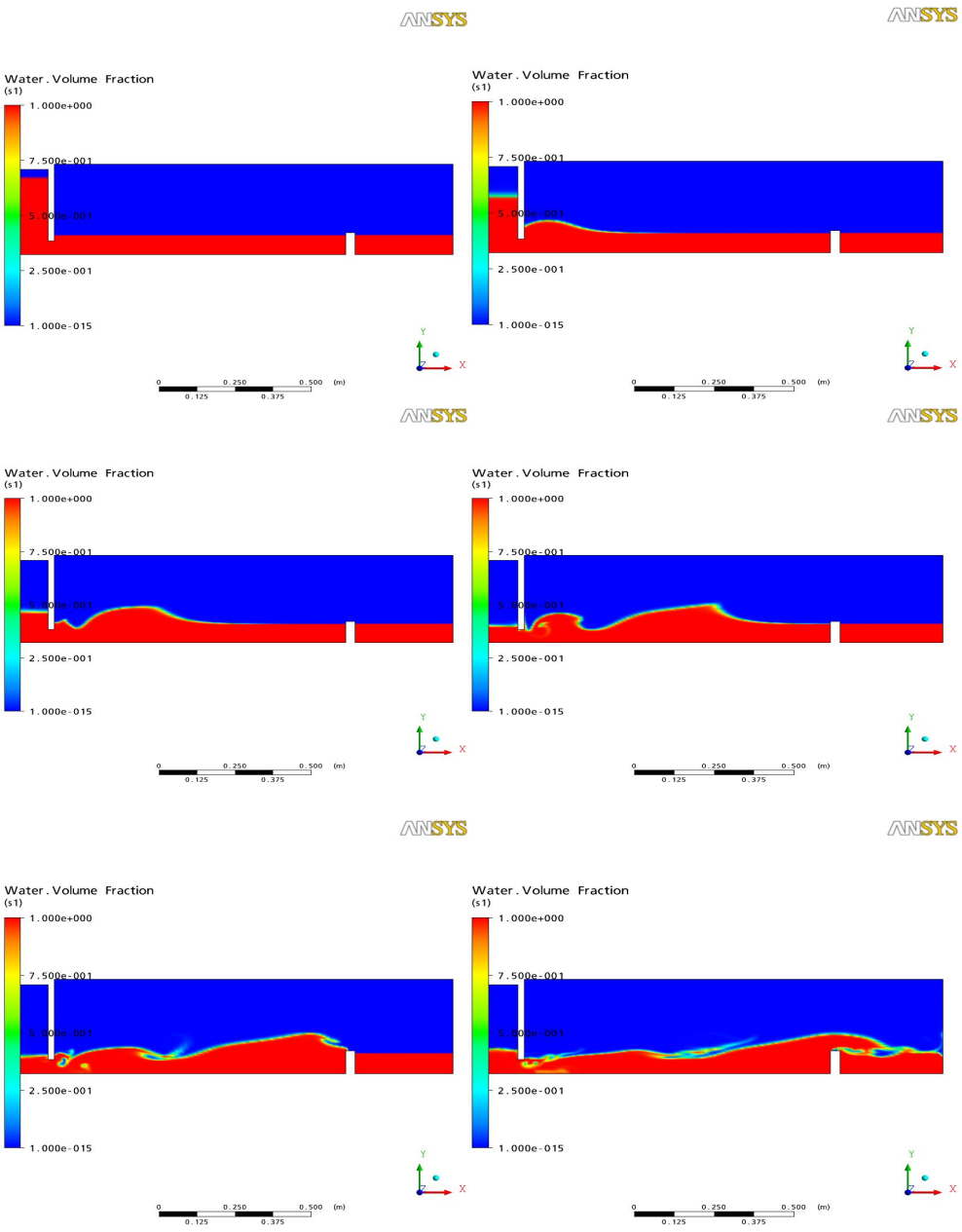


Figure 4.56. Snapshots of the CFX simulation in progress at times t=0s, 0.2s, 0.4s, 0.6s, 0.8s and 1.0s.

The same three properties as before were measured and compared against the experimental results and the SPH simulation. A direct comparison between particle number and grid cell count is extremely difficult given the multiple levels of grid size but it seemed reasonable to use the highest resolution Hydra results (80000 particles).

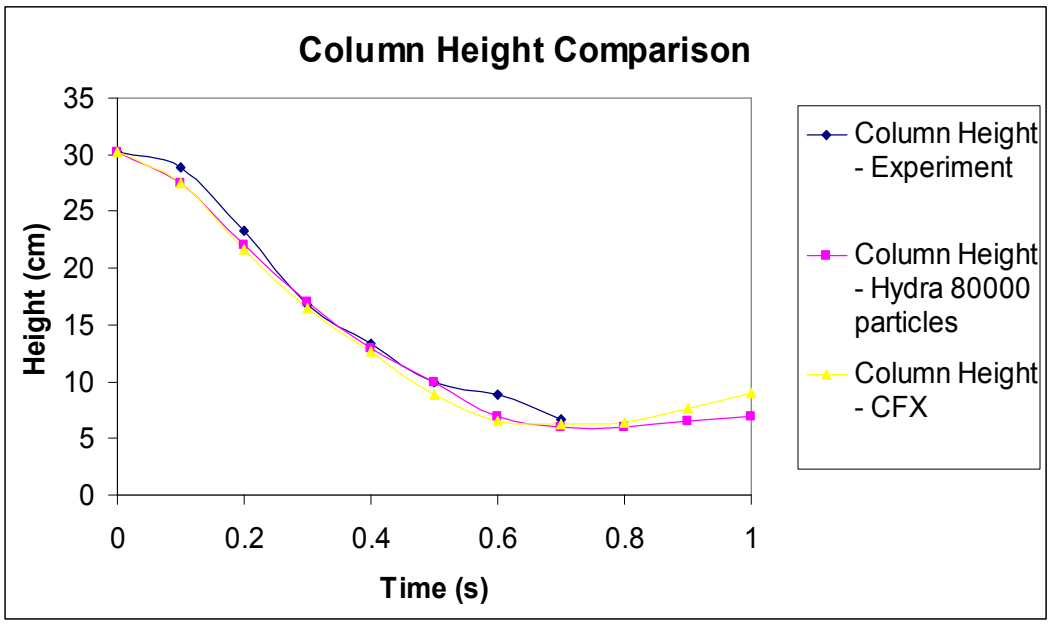


Figure 4.57. Graph comparing the column height between SPH, CFX and experiment.

The results in Figure 4.57 agree favourably with each other. In the initial stages both numerical methods show that the water column falls in height a little too quickly compared to experiment before the experiment catches up just after 0.2s.

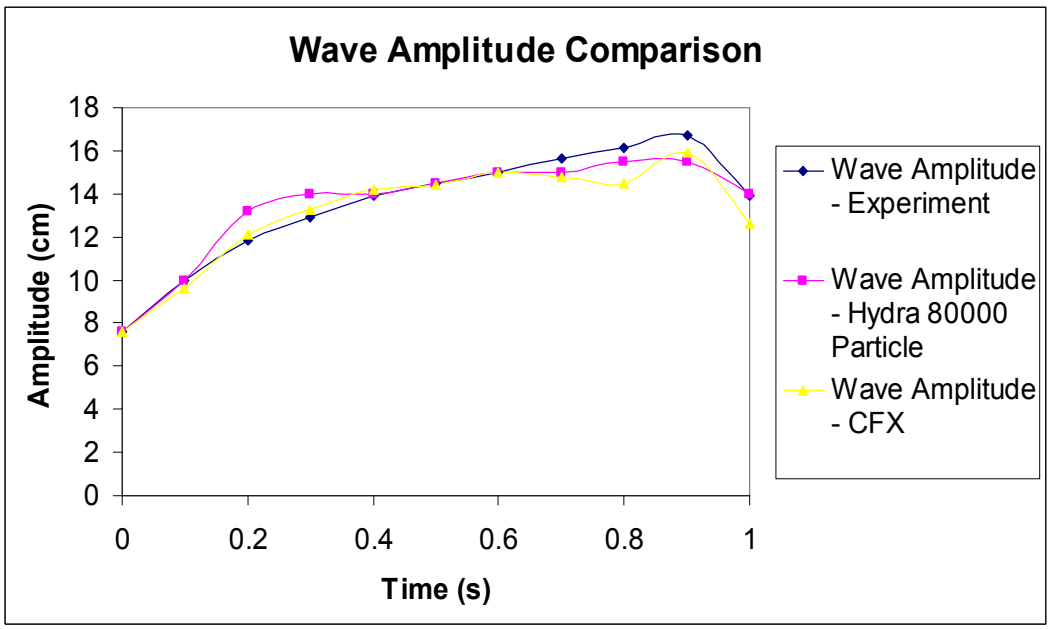


Figure 4.58. Comparison of the wave amplitude between Hydra, CFX and experiment.

The CFX shows a better match to the experimental values than Hydra does in the initial stages with a much smoother curve showing the height of the wave increasing down the channel. Both CFX and Hydra under predict the wave amplitude post the 0.7s mark. This is the region in which the wave breaks over the obstacle. Clearly some difficulty is experienced here by both methods.

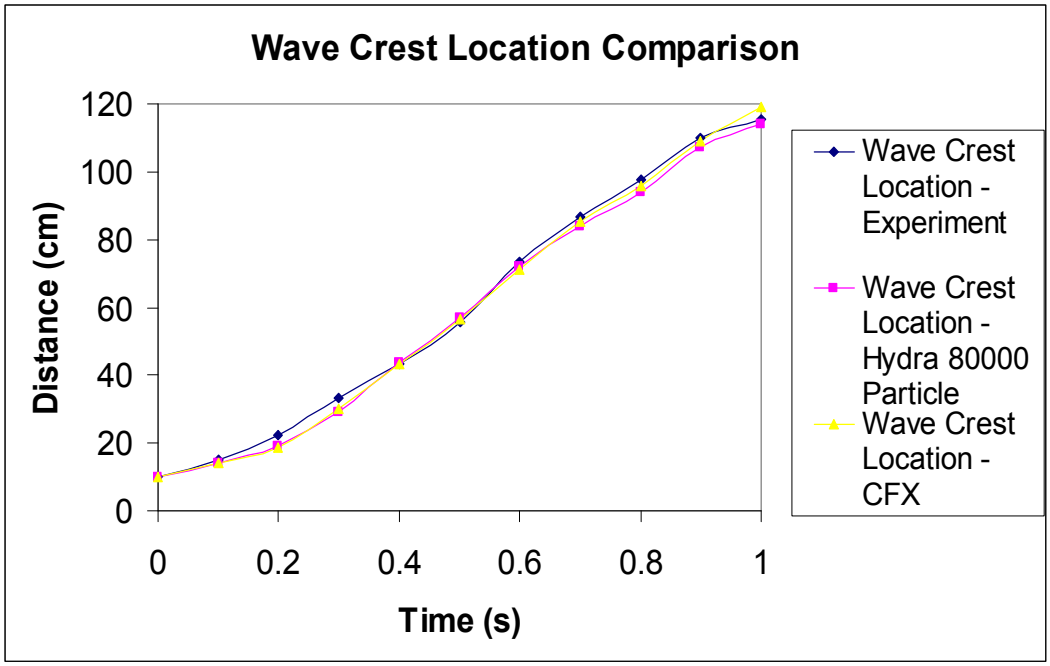


Figure 4.59. Comparing of the wave crest location between Hydra, CFX and experiment.

A very good agreement is seen in figure 4.59 between all three results. The wave crest location is tracked acceptably by both numerical methods and the lines lie on top of each other for most of the graph. The key point of interest to note here is that, as in the Hydra results previously, the experiment travels faster then the simulation which is the opposite to what has been experienced on a dry bed. It is obvious by simply looking at the pictures of the simulation in progress how much more defined the free surface of the water is compared to the CFX simulations in section 4.1. Allowing the grid to be very fine along the line of the wave has made a large difference to the codes ability to resolve the free surface accurately. This relies on the position of the free surface being predicted before starting the simulation but is a clearly superior method to equally sized cells when used.

4.4.2 Partial Dambreak 2

The next experiment is similar to the case in section 4.4.1 but the baffle 1.08m down the channel is removed and the initial column height is increased to 33.5cm. This will allow the wave to propagate down the full length of the cannel without the complication of the baffle as shown in Figures 4.60 and 4.61. In this simulation 33000 fluid particles are used.

Figure 4.60. Snapshot from Maxwell (1977) at t=0.33s.

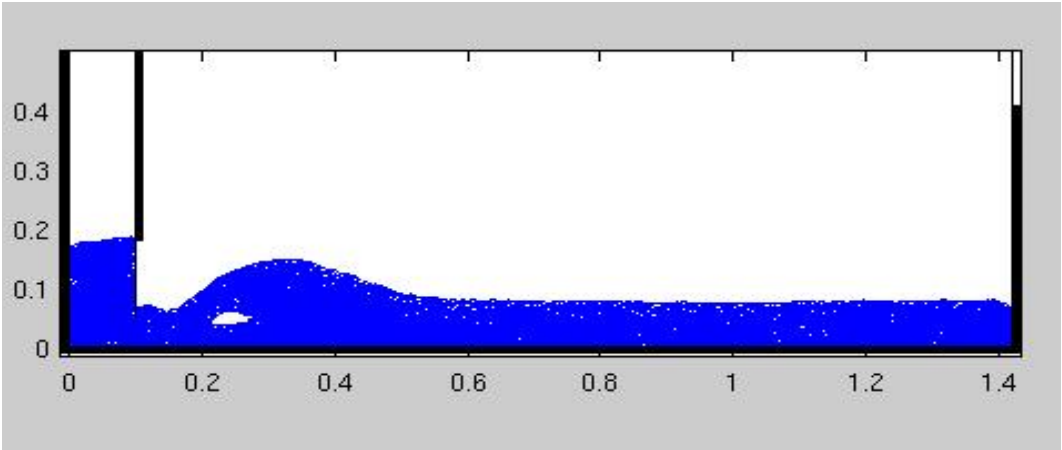


Figure 4.61. Snapshot of Hydra at t=0.33s.

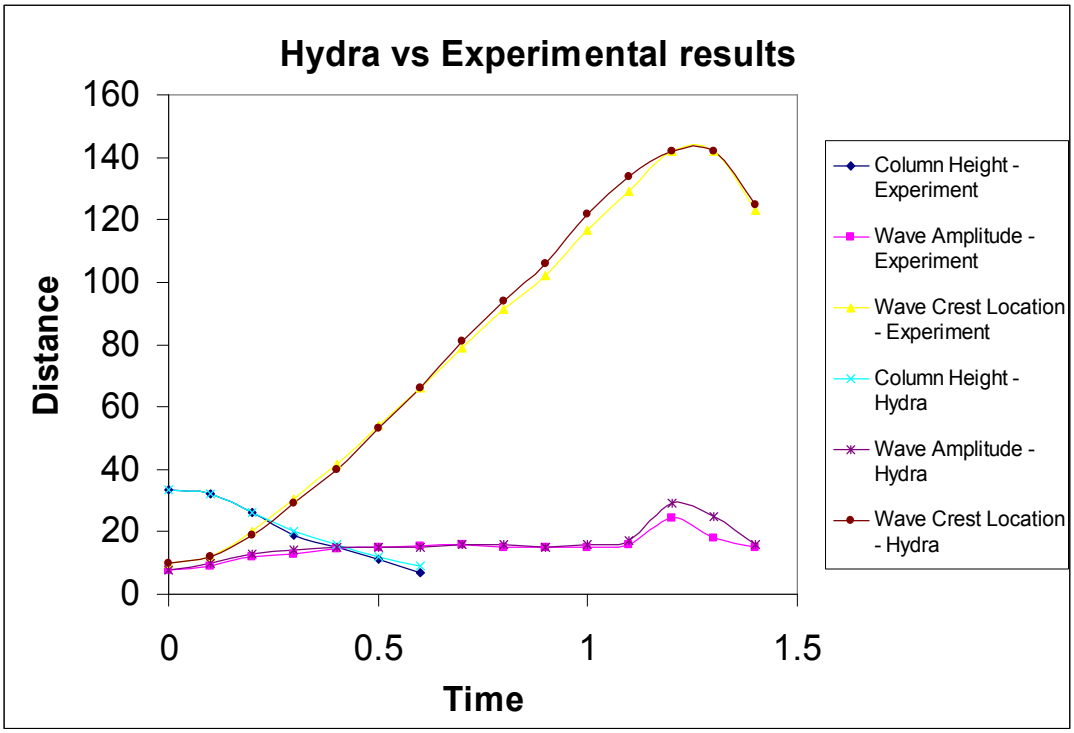


Figure 4.62. Experimental results vs. Hydra.

The simulation was successfully completed and the results compared to Maxwell (1977) in Figure 4.62. The simulation tracks the experimental results very well with only slight discrepancy post the 1 second mark where the wave has struck the right hand wall. Hydra slightly overestimates the wall run up height in figure 4.62 indicated by the jump in wave amplitude.

4.4.3 Partial Dambreak 3

The next experiment is similar to the case in section 4.4.2, i.e. no baffle 1.08m down the channel but the initial column height is increased to 55.3 cm. The water filling the downstream side of the channel is also deepened to 13.7 cm. As the gap at the bottom of the gate is kept at a constant 5.2cm this run is designed to simulate a relatively small dambreak into a deep river channel as can be seen in Figure 4.63. To account for the increase in water volume the particle number is increased to 37350.

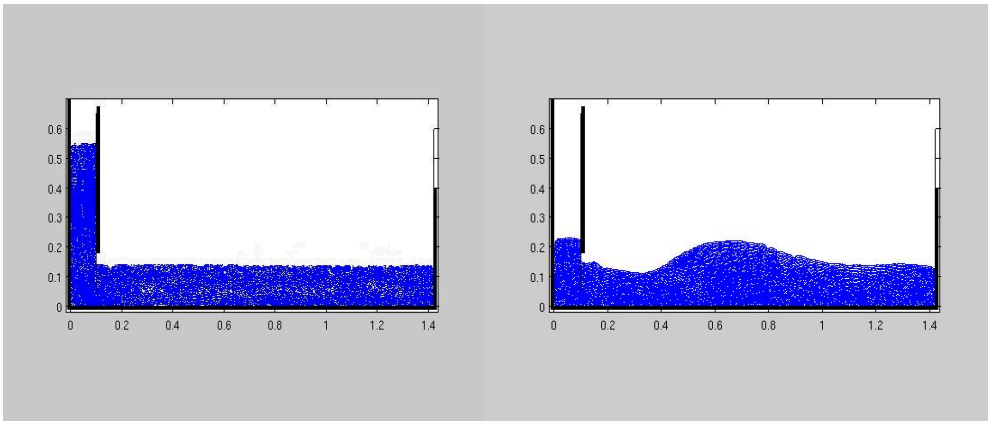


Figure 4.63. Snapshot of the simulation at t=0.04s and t=0.6s.

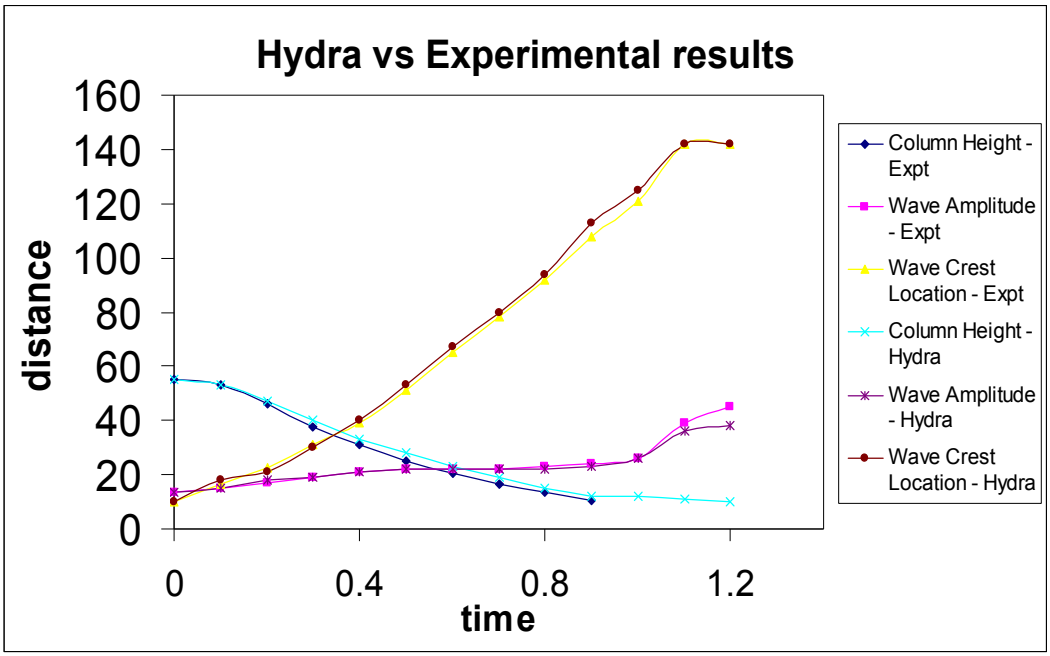


Figure 4.64. Graphs showing the experimental results vs. Hydra.

Analysis of this simulation in Figure 4.64 shows good agreement with experimental results exactly as in the previous case. This time Hydra slightly under predicts the wave run up height on the right wall instead of over prediction in the previous example. Slight discrepancies such as this could easily be the result of errors generated in the measuring process.

4.4.4 Partial Dambreak 4

The next experiment returns to the original format of placing a baffle of the same size (8.5cm by 2.5cm) 1.08m down the channel. However the water filling the channel is 13.5 cm deep. This means that the baffle is completely submerged providing a slightly different test to any that have been observed in the literature (figure 4.65). The initial column height is once again increased to 59.1cm. Everything else is kept the same as previous simulations and using 37200 particles.

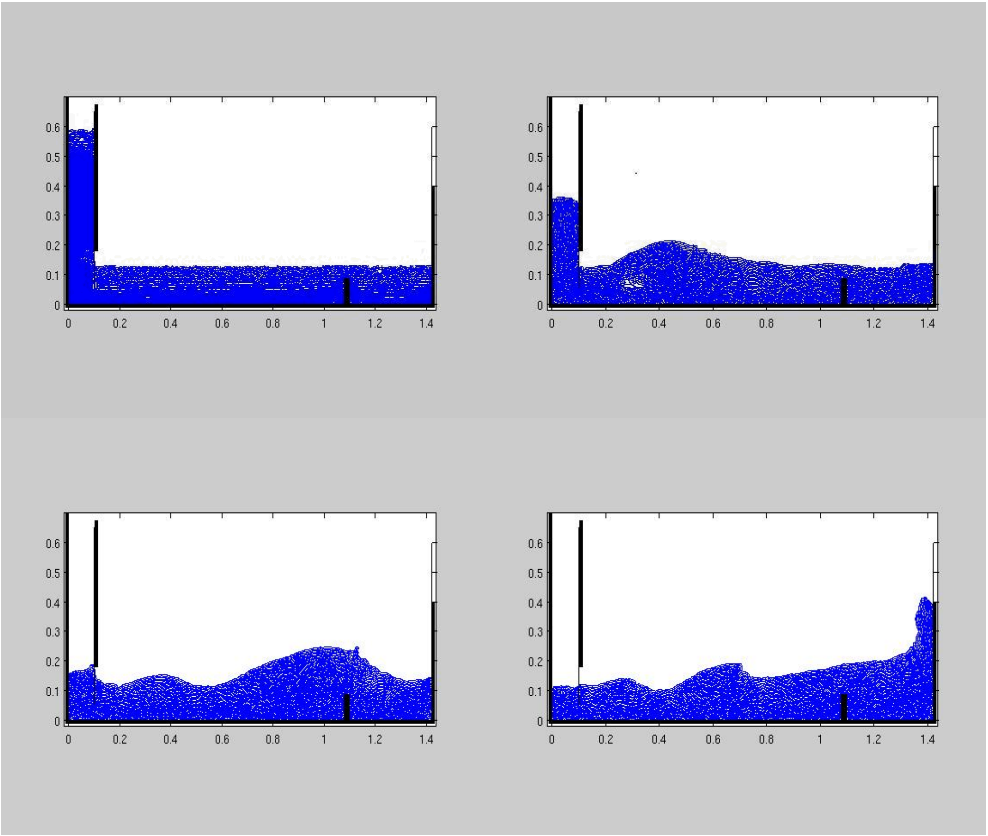


Figure 4.65. Snapshots of the simulation in progress at times t=0s, 0.4s, 0.8s and 1.2s.

The simulation progressed as usual with the largest wave yet generated. A higher initial column height with a deeper channel depth seems to generate a larger, broader wave. The same properties were measured as before Figure 4.66.

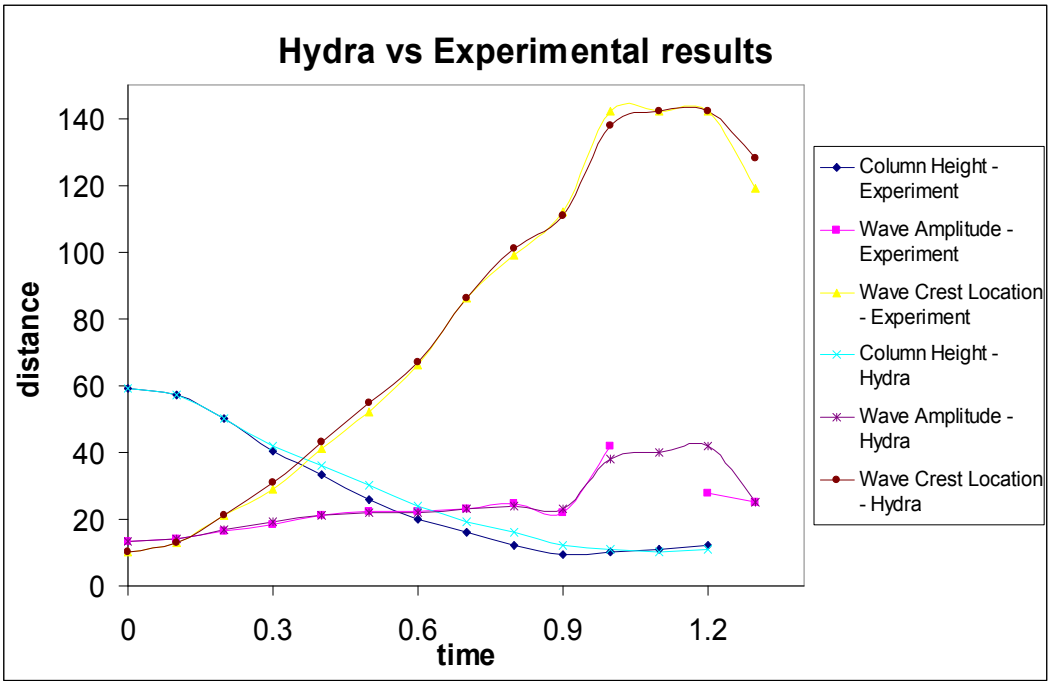


Figure 4.66. Experimental results vs. Hydra. No wave amplitude experimental data was available at 1.1s.

The Hydra solution matched very well with experimental values except for a slight underestimation of the rate at which the column drains. Perhaps surprisingly the submerged baffle did not have as much effect on the overall evolution of the system as had been expected. The wave passed over the baffle almost without alteration in its size; there was a small spike for a fraction of a second which can be viewed in the snapshot at t=0.8s. The baffle also caused a slowing of the wave before it struck the wall.

The general conclusion from the partial dambreak simulations is that the Hydra can handle partial dambreaks into a wet bed just as easily as it can do a dry bed. In fact experience has shown that a wet bed is easier to simulate than a dry bed because there is no sliding of a small number of particles across the floor boundary particles to consider. When Hydra was being developed several different parameters for calculating properties such as smoothing length, sound speed, viscosity, pressure coefficient, wall force intensity, wall force range were considered. In order to ensure the robustness and validity of the code these parameters must be kept the same for all different simulations. It was found that for a wet bed these properties

could be altered by a considerable margin and still produce a satisfactory solution, whereas when modelling a dry bed, a much finer range of values was acceptable to ensure code stability. CFX showed no problem in handling a scenario such as this either with results as good or better to Hydra in the simulation that it performed. The CFX simulation demonstrated that grid methods can be used to provide accurate modelling of free surface locations but a more complicated discretisation than a standard mesh is required. This made the CFX simulation more time consuming to set up than Hydra but was able to recover time lost in generating the initial conditions by reducing the overall number of cells without sacrificing accuracy. 10000 cells done in this fashion provided the same or better level of accuracy as 80000 particles and in half of the computational time. To create a mesh as fine as the smallest cells used with standard equal discretisation would require well over 100000 cells.

4.4.5 Kernel Analysis

As so much of the SPH method relies on properties being smoothed according to the weighting function, or kernel, many researchers have spent considerable time investigating different types and configurations of kernel. From the early Gaussian kernels of the 1970's and 1980's through to recently proposed quintic kernels that go to 5^{th} order equations there has been constant improvement and updating. Presented here is a comparison of four different kernels and their performance against each other in one of the partial dambreak onto wet bed simulations above. The main problem with early kernels was their lack of compact support. This means that they never actually quite went to zero so particles a long way away had an infantismly small impact on the properties of a particle. The issue was that they still took up computer resources even if they had no effect on properties. Modern kernels include a property called compact support where the weighting function falls to zero once the range of the smoothing length has been reached. It is important to avoid ringing at this range limit as errors can creep in. Most kernels used today in SPH fluid dynamics are based upon one

devised by Monaghan and Lattanzio (1985). This is a relatively simple cubic spline function but performs well as a weighting kernel for SPH; the kernel used as standard by Hydra (as shown in section 2.3) is based upon this one. The aim of this section is to determine whether or not the more complicated higher order kernels improve results in open channel flow simulations. Several researchers have in recent years attempted to introduce new kernels of varying complexity and order, as discussed in Liu and Liu (2003) but it has not been made conclusively clear whether there is any major benefit to these efforts. The three kernels which are compared here (in addition to Hydra's kernel outlined in section 2.3.1) are:

- Simple quadratic kernel (by far the simplest shown)

$$W = \frac{1}{8\pi}\begin{cases} 4 - x^2 & 0 \le x \le 2 \\ 0 & else \end{cases} \tag{42}$$

- Fourth order spline (Violeau and Issa, 2007)

$$W = \frac{96}{1199\pi}\begin{cases} \left(\frac{5}{2} - x\right)^4 - 5\left(\frac{3}{2} - x\right)^4 + 10\left(\frac{1}{2} - x\right)^4 & 0 \le x \le 0.5 \\ \left(\frac{5}{2} - x\right)^4 - 5\left(\frac{3}{2} - q\right)^4 & 0.5 \le x \le 1.5 \\ \left(\frac{5}{2} - x\right)^4 & 1.5 \le x \le 2.5 \\ 0 & else \end{cases} \tag{43}$$

- Fourth order polynomial (Liu and Liu, 2003)

$$W = \frac{15}{7\pi}\begin{cases} \frac{2}{3} - \frac{9}{8}x^2 + \frac{19}{24}x^3 - \frac{5}{32}x^4 & 0 \le x \le 2 \\ 0 & else \end{cases} \tag{44}$$

All of these kernels have been normalised in 2 dimensions so that

$$\int W(r-r',h)dr' = 1$$

The difference between these four kernels can best be seen graphically. In figure 4.67 the kernels have been plotted. The kernel from Violeau and Issa (2007) is very centrally peaked compared to the others whilst the quadratic is very broad. A high central peak indicates that much of a particles density estimate comes from itself. Note that for an easy comparison the kernel from Violeau and Issa (2007) has been scaled to fit within the same range as the others.

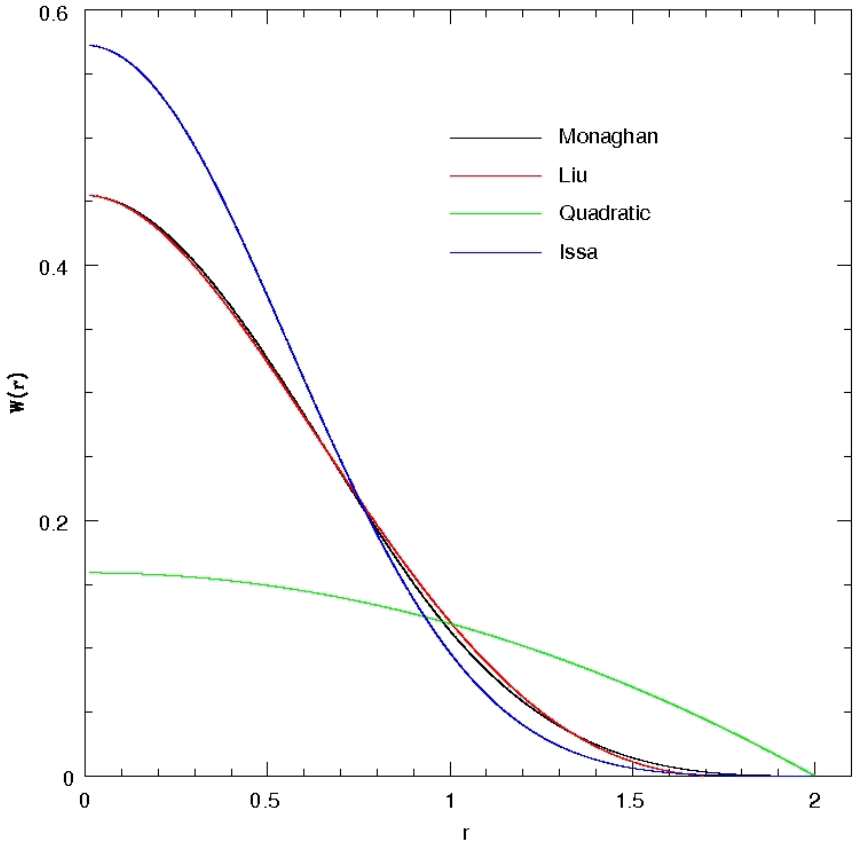


Figure 4.67. Graphical display of the 4 kernels tested.

The partial dambreak chosen to test the kernels was the one outlined in section 4.4.1 with a 30.2cm column height, 7.6cm channel depth and a baffle at 1.08m. Hydra was used to

perform the simulations with all of the kernels written into the kernel.f subroutine. All of the simulation parameters were kept identical for the 4 runs with the exception of the quadratic kernel where the sound speed (which controls the pressure coefficient B) had to be reduced by a factor of four to avoid the water blowing up. 20000 particles were used in each and the same properties as before were measured and can be seen in Figures 4.68 and 4.69.

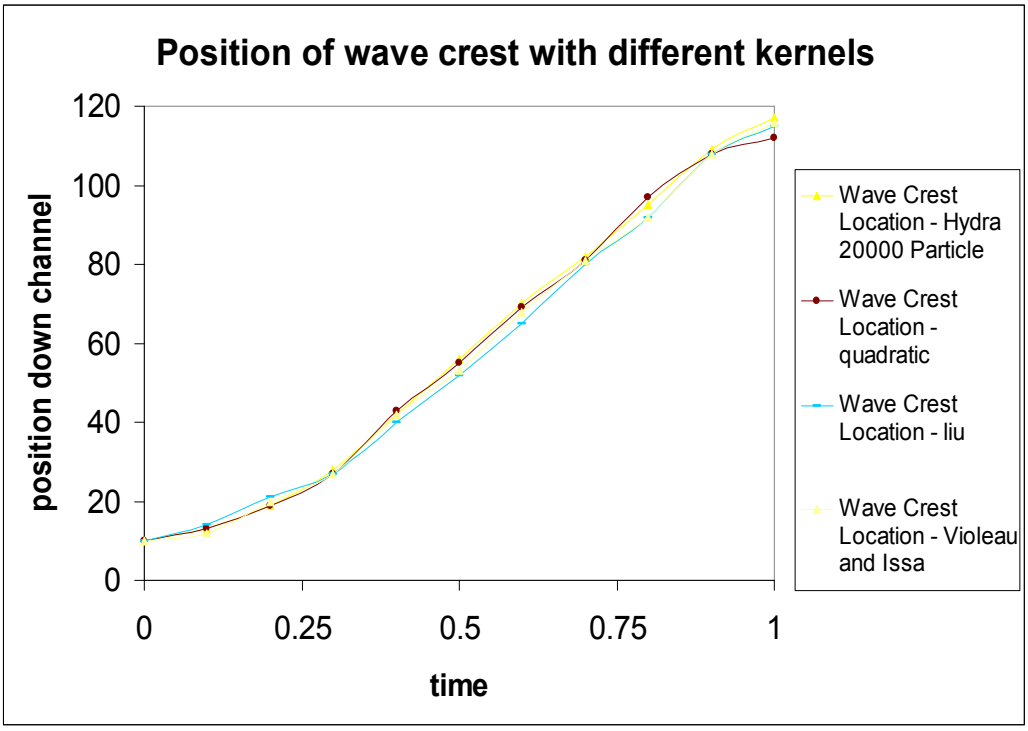


Figure 4.68. Graph comparing the different kernel's resolution of the position of the wave crest.

There are some differences between the various kernels as is to be expected due to their weighting curves being slightly different but there is not enough variation in the tracking of the wave location to say one works better than the others.

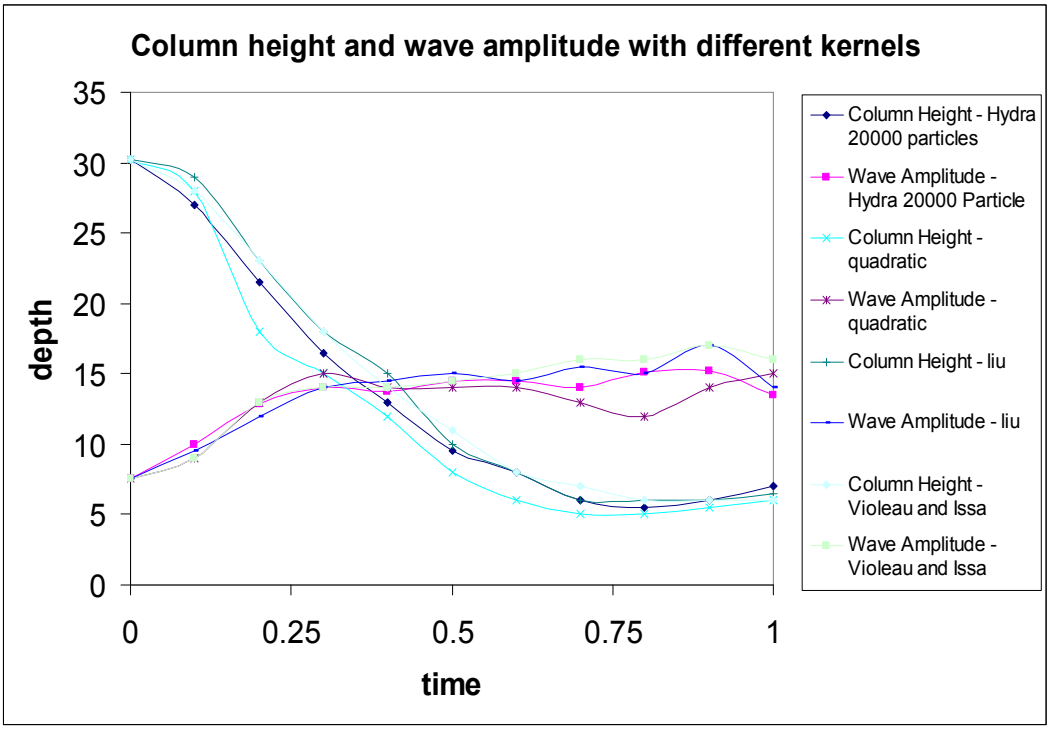


Figure 4.69. Graph comparing the different kernel's resolution of the column height and wave amplitude.

Both of these measurements come out showing discrepancy for the quadratic kernel solution especially in the early stages of the simulation with respect for column height and the late stages for the wave amplitude. Given that in order to make a sensible solution at all, the quadratic kernel required a reduction in the B coefficient, essentially reducing the energy or temperature of the water, it is reasonable to dismiss this kernel as insufficient. There is not much to distinguish the other three kernels' solutions to this problem. The implication of this is that a cubic spline kernel based on the one originally devised in the eighties (Monaghan and Lattanzio, 1985) provides as good results as more complicated modern ones. It is proposed that increasing the complexity of the kernel and raising it up to fourth and fifth ordered equations does not increase the accuracy of the solution and therefore the additional cost of such kernels do not make them worthwhile for use in these types of simulation. Other kernels were attempted on this problem, notably; a step function (the simplest kernel possible), a triangular shaped function and a zero-centred kernel (used sometimes in astrophysics). The results are not included, however, from these kernels as they proved

unsuitable and could not provide a consistent solution. Therefore it has to be concluded that the cubic spline kernel is a required level of complexity in order to be confidant of a numerical solution but, in contrast to statements made by Liu and Liu (2003), increasing further is not needed.

4.4.6 Final remarks on dambreak simulations

Hydra has been validated for the simulation of a breaking dam into an open channel. This validation took the form of a comparison against other SPH codes, traditional grid based simulation methods and experimental results from both myself and independent researchers. The results were generally good and demonstrated that Hydra has been converted successfully to a water based simulation code and can accurately model the behaviour of water both quantitatively and qualitatively. Other methods were also analysed all of which had their benefits. It was theorised that introducing disorder of particles deliberately by way of a glass start instead of a regular grid is a better way of generating initial conditions. It was also shown that a fairly simple cubic spline kernel provides acceptable results and no benefit is achieved for attempting to refine the kernel further. A limit was proposed for the SPH method regarding flow on a dry bed based on particle number and smoothing length. A water column collapsing onto a dry bed must maintain sufficient particle depth all the way down to the toe of the surge in order for the results to be trusted. This does not apply when a wet bed is modelled (as long as the bed is more than a few particles deep).

Even though the scenarios modelled were fairly simple when compared to real life dam/river geometry, no method could be described to have simulated all of the physics perfectly. The SPH methods could arguably be described to have simulated the simplest scenarios better than CFX but when the geometry becomes more complicated CFX catches up and starts to overtake SPH as the more accurate model. This is mainly because, despite over prediction of water splash height, horizontal momentum is not lost as easily. When using a regular grid,

CFX can not compete with SPH for free surface resolution (at equal computational cost). If a more adaptable mesh is used or cells are concentrated along the pre-predicted free surface location, the opposite is true. Of course, it must be mentioned that Hydra has only just been developed for this type of fluid simulation during this PhD work whilst CFX has been developed commercially for many years.

Chapter 5

More Complex Simulations

5.1 More advanced functions

5.1.1 Inlets and Outlets

In the previous sections the water version of Hydra has demonstrated an ability to provide numerical solutions to problems such as dam breaking, wave impact and sloshing. However there is a limitation to the problems that Hydra can tackle in its current state; they all require a set number of particles to be present and that this number of particles is present at the start and end of the simulation. Many problems requiring CFD solutions involve steady state solutions or geometry where fluid enters and leaves the domain. River channel flow, flow over a weir, and flow into a ships hull to mention just a few potential applications. To increase the usability of Hydra some modifications have been made. These comprised of allowing for inlets (e.g. a pump or upstream of a river) where particles can be introduced into the simulation part way through and outlets (e.g. a hole in a container or downstream of a river) where particles can exit the simulation at any time.

Inlets were programmed into Hydra fairly simply by allowing any place to be designated as an inlet in the domain. Usually this would be at the end of a tube or the edge of the domain but could in theory be anywhere in the model even suspended in mid air if needed. Before, Hydra would place particles as designated in the initial conditions created from the startup.F subroutine. Where ever the inlet was placed a line of particles of length equal to the length of the inlet would appear according to a given rate adding new particles throughout the

simulation. These new fluid particles can be started with a set velocity to drive them away from the inlet or they can be placed there with no velocity. Each timestep the code can check whether or not it is time to add another layer of particles. The only main difficulty that presents itself when particles are being added is that there is now a constantly changing fluid mass within the domain. The code responds to this by keeping a particle identification number attached to each particle. A sum of these particles can reveal the total mass that should be within the domain which enables mass conservation and does not necessitate the inclusion of a continuity equation into Hydra.

The programming of outlets has necessitated the creation of two new types of particle, namely “exit” particles and “none” particles. When a fluid particle approaches an outlet it is turned into an exit particle. An exit particle is exactly the same as a fluid particle with the only different property being the type it is assigned to. The region in which a particle becomes an exit particle is very small and directly in the flow towards the outlet. Fluid particles become exit particles in most cases but if there is a sudden pressure pushing back towards the centre of the domain for some reason then it is possible for an exit particle to become a fluid particle again. Once the actual outlet itself is reached by an exit particle, i.e. the edge of the domain, then the particle becomes designated a particle of type none. This particle is then removed and its effect on other particles due to being within a smoothing length is cancelled. The mass represented by the particle is also removed from the system. Unlike an exit particle, a “none” particle cannot return from the void and become an exit particle, it is gone forever. Note that particles that exit through the outlet are different to particles that escape either through splashing or wall penetration in that they are removed completely from the simulation and not allowed to float away or placed in a “safe area”.

5.1.2 Container Filling Simulation

In order to check the new inlets and outlets were functioning a simple test was devised consisting of the filling up of a container with a hole in it. An empty square container of side length 10 metres was created from boundary particles. At the top of the left hand side a tube was placed designed to simulate a water jet or tap. The inlet was placed at the left hand side of this tube and water pumped in through it. Once in the tube there were no other pressures or forces acting on the water other than the standard ones. Fluid particles are not imparted with velocity once they have entered the tube; they are pushed down the tube by the creation of new particles behind them (though they could be given initial velocity if a powerful jet is required). Some way up the right hand wall of the container a section of the boundary particles were removed and a second horizontal tube put in their place. The outlet was at the right hand end of this tube. As particles near the outlet they change types from fluid to exit, then once they have reached the end of the tube they become type none and are removed from the simulation. In Figure 5.1 are several snapshots of the filling of the container in progress.

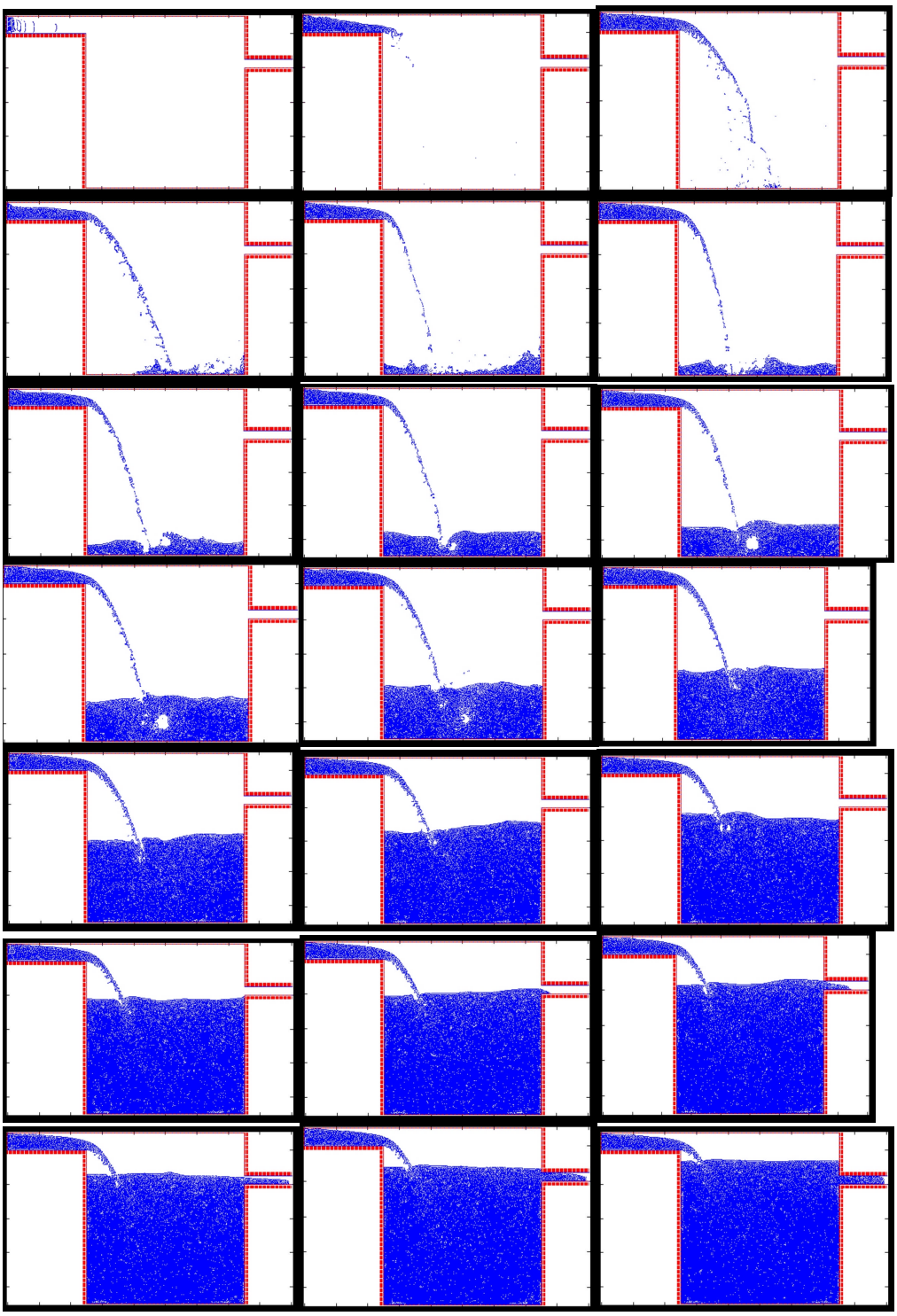


Figure 5.1. Snapshots of the container filling simulation. Images are not equally spaced in time.

The container slowly fills up and the simulation displays complicated internal motions and swirling consistent with a bucket of water being filled. Purely visual analysis of the

simulation indicates that it displays realistic water motion. The simulation lasts for ten seconds in real time, which is enough to fill the container and demonstrate the inlet and outlet are functioning properly. A slightly reduced rate of water input at the inlet would allow for a steady state solution where the container depth is unchanging with time. The walls show no sign of penetration and the main body of water is completely stable if highly turbulent where the jet freefalls into it.

The inlet creates water at a constant rate for the whole simulation. However a potential problem for future simulations was noticed late on in the simulation. As the container began to fill, the time taken to simulate a given amount of real time continually increased. This is due to the fact that more and more particles were being introduced constantly into the model all of which had to be kept track of by the computer. The total number of particles of all types in the simulation cannot be said therefore as it was constantly increasing but at the end well over 100000 were being tracked. For larger more complicated simulations this may make modelling too computationally expensive. A solution to this problem is proposed in section 5.1.3.

5.1.3 Particle recycling

In order to solve the problem of the simulation being slowed due to constantly increasing particle number a particle recycling system was devised. Instead of creating new particles with new identification numbers at the inlet as before a reservoir of fluid particles was created prior to simulation commencement. This consisted of a rectangle of particles where a side length is equal to the inlet in Figure 5.2.

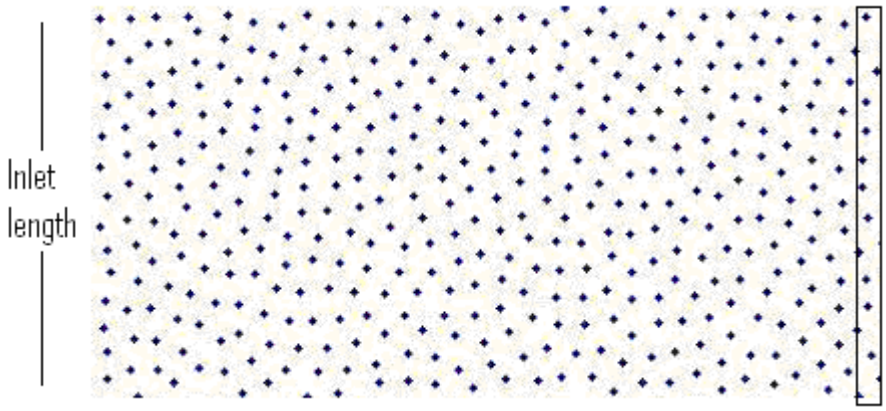


Figure 5.2. Demonstration of the particle recycling principle also showing catchment sweep.

All particles still require an ID number throughout the simulation however. When new particles need to be entered through the inlet a catchment area sweeps through the reservoir and selects the first particles it comes across (equal to one line of particles so the water flow rate is constant). These are then pumped into the domain as usual. When a particle becomes a type none and leaves the simulation it is recycled back into the reservoir. The ID number is not lost, nor is it retained in the memory of the computer, rather it is freed up to be used again in the simulation with a new particle. Technically it is not the particle itself which is recycled but the ID number. The particles in the reservoir are represented not as a separate domain but as numerical values in a computer file. Every particle ID that is lost is freed to be reapplied to a new particle about to be entered. This stops the total number of particles in the simulation from spiralling out of control and, once a steady state solution has been reached where input and discharge have equalled each other, particle number should become approximately constant. It is important to make the reservoir large enough so that sufficient particles can be selected from it until the solution becomes steady state or the flow of water through the inlet will dry up.

5.2 Weir Simulation

A weir is a Hydraulic structure designed to maintain water level upstream. It is often use as a means of measuring or sometimes controlling the flow or discharge rate of a river or stream. There has been some investigation of Hydraulic structures such as weirs from a CFD standpoint in the literature (e.g. Wahl et al, 2000; Chen, Dai and Liu, 2002) but the subject has been far from exhaustively researched. This research details an attempt at modelling a simple broad crested weir using the SPH code Hydra with validation against another numerical solution and experimental values. The experiment was described in detail in Hager and Schwalt (1994) and featured a long horizontal channel with a rectangular broad crested weir of length 500mm and height 401mm placed in the middle. The same scenario was modelled by Hargreaves et al (2007) using version 6.2 of Fluent, Fluent Inc. (2005). The mesh used in Fluent employed quadrilateral cells of various sizes. There were two sets of non-conformal grid interfaces and extensive use was made of geometric progressions in the meshing of the domain (Hargreaves et al 2007). This allowed for a reduction in the total number of cells needed.

Hydra set up the same geometry using boundary particle and ghost particles including ghost particles lining the inside of the weir. The channel was filled with water initially to speed up the emergence of a solution. An inlet of size 500mm was placed on the floor on the left hand side and created a slow bubbling up effect rather than a powerful jet. This was to represent a continuous supply of water flowing from upstream. An outlet was placed against the far right hand side of the domain to remove particles from the domain once they have traversed the entire channel. Extensive use of a particle recycling reservoir was made throughout this simulation. It is not possible to say exactly how many particles are used in the Hydra due to the number constantly changing but of the order of 10000 is estimated once a steady state solution is reached. A small pressure outlet on the weir is used in the grid based simulation to allow the weir to function correctly in the initial stages and prevent the water dribbling down the side of the weir. This is not required in SPH as air is essentially represented by the void.

Once the simulation is begun it creates a wave of water travelling down the channel. This crashes over the weir in a disordered fashion and slowly pushes water downstream towards the outlet. It takes considerable time to achieve a steady state solution where the upstream and downstream portions of the channel maintain the proper free surface levels. In total this is achieved after approximately 10 seconds of real time (many thousands of timesteps in the code) which interestingly is the same as is quoted for the Fluent simulations in Hargreaves et al (2007). A zoomed in image of the weir itself showing the particles and the velocity vectors associated with them can be seen in Figure 5.3.

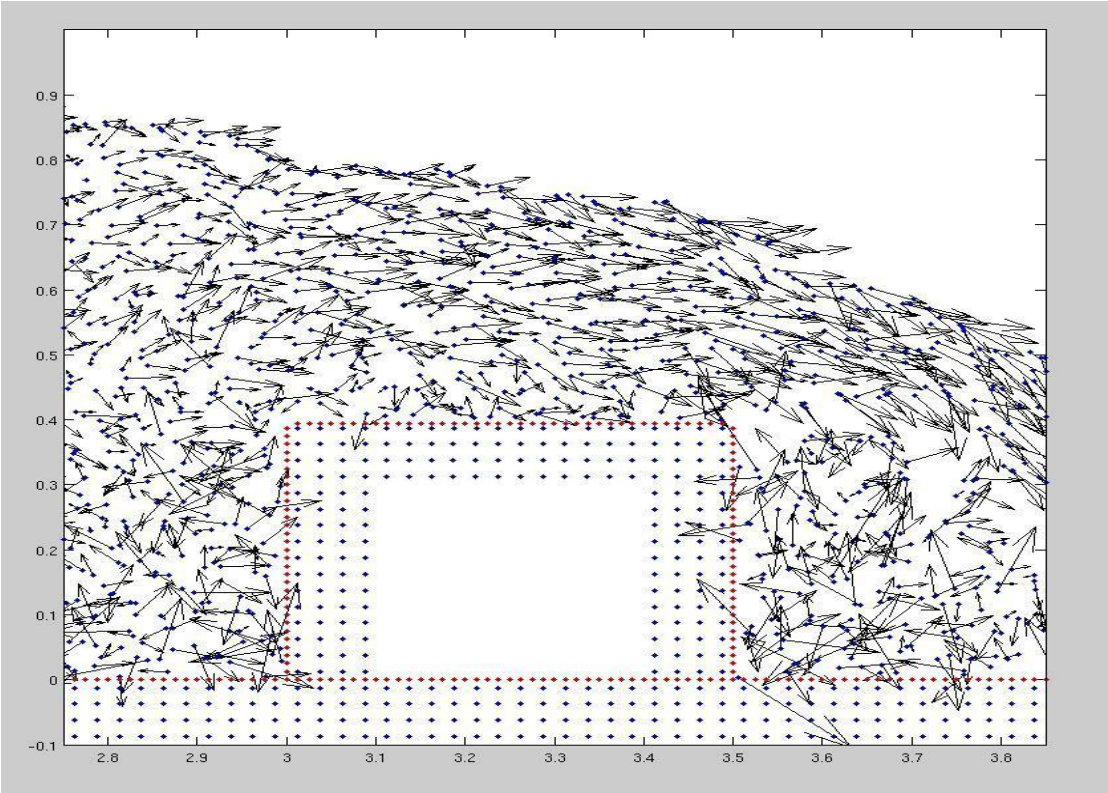


Figure 5.3. The weir showing the shape of the water flowing over it.

The particles and their velocity vectors show the direction of the bulk of the flow and the increase in velocity as the right hand side of the weir is reached. Considerable particle disorder is seen on both the upstream and downstream sides of the weir along the bottom. Observation of figure 5.4 from Hargreaves et al (2007) allows the free surface shape over the weir to be compared with Hydra.

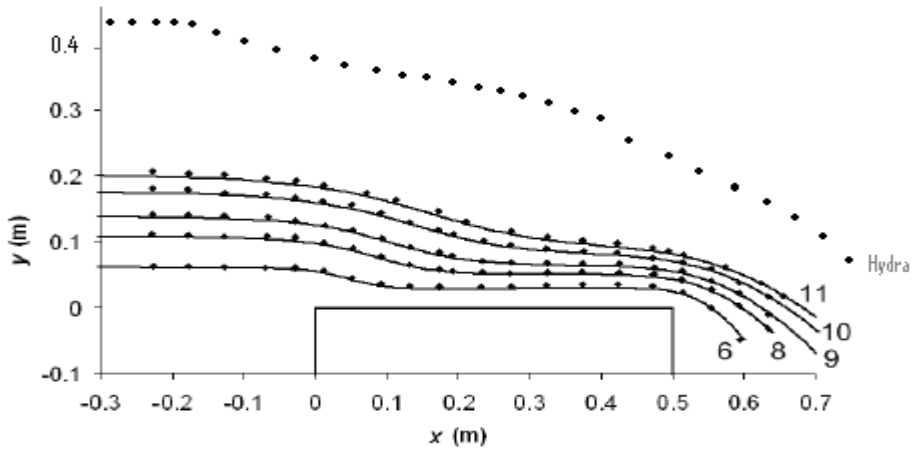


Figure 5.4. Results from Hargreaves et al (2007) showing the free surface shape over the weir with results from Hydra superimposed on top.

The shape of the free surface is closely matched by all three solutions. However the Fluent simulations are able to restrict the solution to a predetermined upstream channel depth and several depths are modelled by Hargreaves et al (2007). Hydra can only do this by varying the rate at which the particles are added by the inlet. Despite attempts to control this, the SPH particles seem to consistently over predict the free surface height upstream by around 30%. The particles upstream appear to take on an almost aerated effect as they travel down the channel. The reason for this is not known as the free surface shape is consistent with experiment and other numerical methods throughout the channel and the downstream section of the weir does not seem to be affected.

In the image in figure 5.6 the flow of the water over the edge of the weir can be seen and the initial part of the downstream section of the channel. An air gap can be seen near the top right hand corner of the weir. The momentum of the main flow carries water over this and the majority of the water goes on down the channel. This is represented by the long ordered velocity vectors from left to right of the image. The immediate downstream section of the channel shows a triangular section of unsteady and complex flow (Figure 5.6). This highly turbulent churning effect in the water is not unreasonable when you consider weirs in nature;

there is a considerable amount of white water as it is known up the right hand side of the weir and beneath the impact of the water falling into the channel. This is in agreement with the findings of Hargreaves et al (2007) (figure 5.5).

Figure 5.5. View of weir and white water from Hargreaves et al (2007).

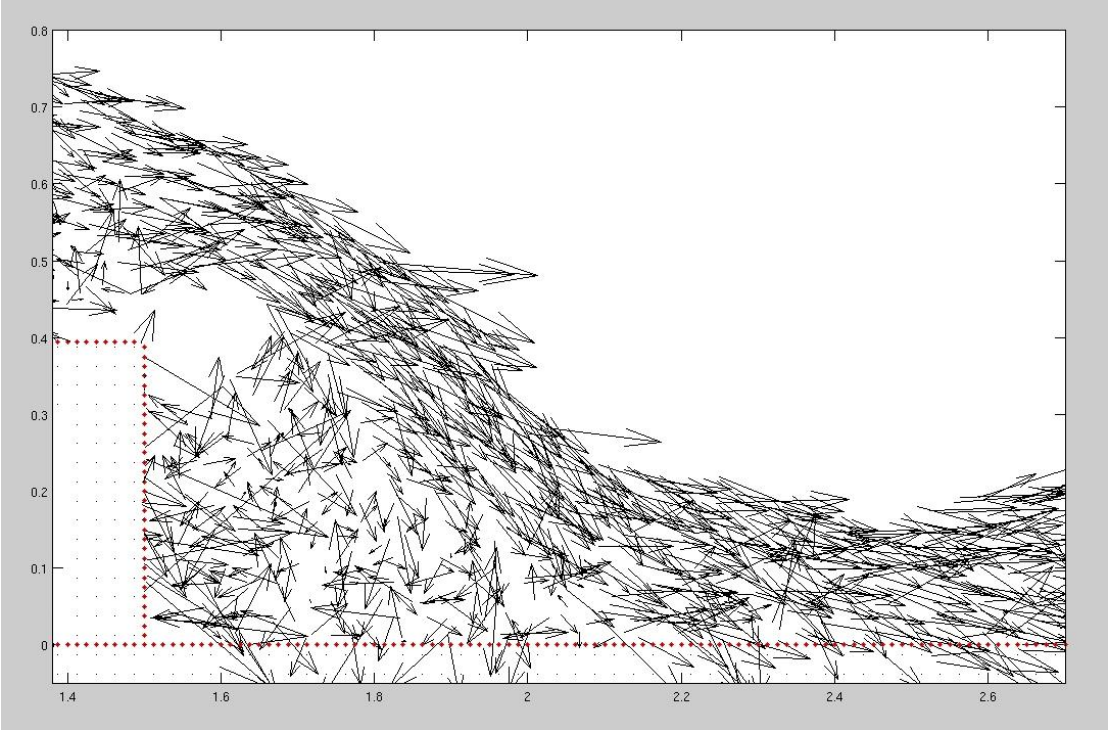


Figure 5.6. The right hand side of the weir and part of the downstream section of the channel in Hydra with velocity vectors.

In a new simulation the weir was substantially extended to examine the effects, if any, a longer stream had on the numerical solution. This provided a much longer upstream portion so that the inlet was nowhere near the actual weir. This is designed to be compared with run 11a from Hargreaves et al (2007). An image showing the weir can be seen in figure 5.7.

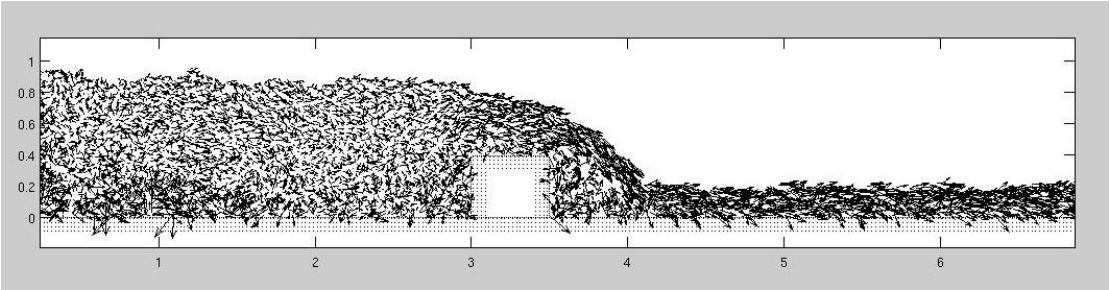


Figure 5.7. The entire extended weir simulation. Inlet, outlet, and free surface shape are visible complete with velocity vectors.

The simulation is successfully completed and a steady state solution is reached (after a longer period due to additional length). The same parameters are used to create this size of weir as the original. The particle recycling reservoir was increased in size to cope with the extra particles needed throughout the simulation. Once again the depth in the upstream region was overestimated in Hydra by the same amount as before; the increase in channel length appears to have no effect on this. The water is seen to accelerate down the face of the weir in the same manner as before with downstream velocity greater than upstream represented by the darker appearance of the downstream flow due to heavier velocity vectors. White water is generated along the right hand face of the weir and the downstream channel depth varies slightly as the water moves along. For the extended weir model in Hargreaves et al (2007) the discharge was fixed to a constant value and the predicted free surface level of the downstream flow was measured and compared to the experimental value obtained from Hager and Schwalt (1994). Similar to observations made by Hargreaves et al (2007) a completely steady state solution is never reached due to small variations in the downstream flow (which is completely reasonable from observation of real weirs) a close enough match is

reached to make a prediction in Hydra based on the average water depth over the downstream section. Hydra predicts a free surface level of 194mm which agrees closely with the Fluent prediction of 199.9mm (Hargreaves et al 2007) and the Hager and Schwalt (1994) experimental value of 202.1mm. Hydra has therefore slightly underestimated water depth downstream but overestimated upstream. One important consideration to bear in mind with the Hydra model is the amount of particles used. This was quite low and the simulation should be repeated with more particles – at least 20000 is recommended. These numerical models were all 2D in nature and this scenario was not attempted in 3D using the SPH method. One simulation was performed in 3D using Fluent by Hargreaves et al (2007) who found the use of a 2D model did not significantly affect the result and was substantially cheaper computationally.

Chapter 6

Conclusions

6.1 Summary

The project was designed to convert a fluid simulation code named Hydra, which uses the smoothed particle hydrodynamics method, designed for astrophysics based problems into one designed for water based problems. Hydra was then used to study the flow of water in a variety of scenarios comparable to open channel flow and flood inundation events; dambreaks and weirs were highlighted. These simulations provided the basis for a verification and validation study of the new water version of Hydra. The benefits of continued research into this alternative method of computational fluid dynamics were discussed and reasons why SPH or meshless methods in general could be used ahead of traditional CFD were highlighted.

The theory behind SPH and CFD was summarised and the study of flood inundation scenarios expanded to compare SPH with other CFD methods. Experimental techniques were also used and the advantages and disadvantages of these were also considered.

6.2 Conclusions

After an initial review of CFD principles, SPH theory and previous SPH simulations the Hydra code was used to demonstrate an accurate solution could be achieved to the Riemann shock tube problem. A demonstration of the orientation independence that is a useful

property of an SPH based code over some grid based codes was shown in a separate Sod shock performed at oblique angles. The initial aims of this research have been successfully achieved with the SPH based simulation code Hydra having been converted from astrophysics to water flow modelling. This was primarily achieved by recoding the equation of state, removing astronomical physics information for replacement by liquid physics and the implementation of solid boundaries or walls. This success was verified by the implementation of several custom built code tests which demonstrated the stability of the water when subject to no external forces and to gravitational forces. These tests included a lid driven shear cavity in 2D and 3D, a free floating water square, a container filled with water and a draining tank. A detailed calculation of the theoretical time taken to drain this tank was performed and the figure (11.8s) compared favourably with Hydra simulations. This test was also used to demonstrate resolution independence and volume conservation. The inclusion of solid boundaries forming walls, barriers and containers was implemented by the creation of two new types of particle. The first of these were the boundary particles which operated as points of repulsion marking the actual boundary itself and preventing fluid particles from moving across the boundary. These were coded to increase in repulsion strength as distance toward the point decreased. An improvement over the standard boundary particle approach was to have the repulsion force act normal to the orientation of the boundary. Hydra therefore has had three sub-types of boundary particles implemented; x direction, y direction and z direction. This has the advantage of improved friction treatment. The second type was the ghost particles used outside of boundary particles which prevented a sudden drop off in density outside of the boundaries thereby eliminating spurious boundary pressures. The use of ghost particles in this fashion is not a new idea but for the first time a formula was devised in this research to calculate the actual number of layers required by a simulation in order to remove the spurious boundary pressures. A simple calculation following this formula preformed at simulation set up will ensure that sufficient ghost particles are placed but unnecessary layers are removed speeding up the solution.

The newly converted Hydra was presented with several different dam breaking problems of varying sizes and including both total dambreaks and partial dambreaks. Dry channel beds and wet channel beds were considered with varying lengths of channel after the dam. The results from these simulations were compared with a wide variety of sources to get a detailed analysis of Hydra's performance. SPH simulations performed by other authors, solutions gained from traditional grid based simulation codes, experimental results and independent experimental results were all used for comparison. All numerical methods appear to capture the initial collapse of the water column well. All numerical methods slightly over predict the speed of the water surge as it travels down a dry channel bed when compared to equivalent experimental results. Hydra is no different in this respect but tended towards being slightly slower than most other numerical methods. This may be due to better boundary treatment within Hydra; closer constant contact between boundaries and fluid particles creates more friction with the bed floor. The difficulty in obtaining accurate data from dam breaks into an open channel experiments and how this could allow errors to creep into results, especially onto a dry bed, was noted. Typically inaccuracies in experimental data were found to produce an underestimation of flow speed down a channel. Effects created by the gate removal proved difficult to remove and these would cause a delay in the propagation of the wave down the channel. The ability of Hydra to track the free surface of the water down the channel was shown and the advantage that an SPH code has over a grid based solver in this respect discussed. A key advantage in using the SPH method is the free surface does not have to be resolved separately as it has to be in many grid codes; instead it lies where the particles are situated. Overall the solutions produced by Hydra compared favourably with current leading simulation packages (e.g. Fluent).

Dambreaks into a long channel were shown experimentally and in Hydra simulations. This had the advantage of allowing a full free surface validation down the entire channel length for the entire simulation which is not often observed in the literature. These experiments were performed with a longer channel length relative to the size and depth of the dambreak than has been observed in any of the literature. Analysis of these unique experiments/simulations

allowed the discovery of a previously unpublished limiting factor in the ability of an SPH code to simulate water travel down a dry channel bed. A mathematical limit to the length of channel that can be accurately simulated using a SPH code was proposed based on particle number, smoothing length and initial conditions. The principle of the limit revolves around the fact that at every point of the main flow there must be some particles which do not see either the top or the bottom of the flow in order for the code to provide a good density estimate. This provides a minimum number of particles which must be met before a simulation should be commenced. This newly discovered limit will allow an SPH researcher to predict the limits of his model before simulation and adjust if not sufficient saving time and reducing the chance of a misleading result. When a wet channel bed (especially a deep channel depth) is introduced downstream the over prediction in wave velocity by the numerical methods disappears. The results from Hydra indicate that it handles a wet bed with more proficiency than a dry bed especially when obstacles are introduced into the channel. The advantages of using an adaptable grid or varying the grid size so that more cells are present around the regions of interest within a traditional grid method were clearly shown with a wet bed dambreak. Hydra demonstrated and excellent ability to provide a solution to the wet bed dambreak problem that was equal in accuracy to any of the other fluid simulation packages used but the speed of attaining this solution was found to be inferior to CFX when a problem specific adaptable grid was used. Several different smoothing kernels used by various SPH researchers were tested to determine whether or not the increasingly complex kernels favoured by some actually increase solution accuracy. There was no evidence that this was the case and it was concluded that a relatively simple cubic spline kernel provides the same level of accuracy as any kernel tested.

Finally Hydra was further modified to include inlets and outlets in order to provide the possibility of steady state solutions such as continuous flow over a weir. This is a big improvement over most SPH water codes and adds a new dimension to the attractiveness of Hydra. The time taken to achieve solutions in a container filling scenario led to the invention of a particle recycling scheme to be developed for Hydra which allowed inlets and outlets to

provide steady state solutions in a much more efficient way and subsequently reduce simulation time. A weir was modelled and compared with data obtained from published results. The free surface shape down the channel and the churning water on the immediate downstream side of the weir was replicated well. The prediction of the downstream water level was very close to published data (194mm compared to 202mm). However the water depth upstream of the weir was over estimated in Hydra by a factor of 30%. Nevertheless, by virtue of the comparisons against experimental and published numerical data in this thesis, the potential to simulate a wide range of water flow simulations involving open channel flow has been demonstrated. The code developed in this research is a successful beginning to this goal but more research is required in order to broaden the scope of Hydra and allow it to be considered a true alternative to existing simulation packages such as CFX or Fluent. A final point that the conversion of an already proven code to water based simulations was much quicker to get into action than a new code built from scratch was made.

6.3 Future Work

Several recommendations for future research which could be used to enhance the suitability of Hydra, and the SPH method in general, for use in flood inundation simulations and open channel flow can be made. SPH has the potential to replace traditional grid based CFD methods in many applications but more research is needed in order to raise its profile amongst CFD researchers and convince them of its accuracy and adaptability. With regards to Hydra specifically, suggestions can be made for both the short and long term. A good continuation of the current models presented in this project would be a detailed study of wall overtopping and coastal defences. Various sizes of water waves can be simply generated by methods such as shown in this thesis and analysis of their impacts on different shapes and sizes of sea wall could be of use to many. The inclusion of sloped boundaries would be a good first step to look at more realistic beaches in the study of coastal waves. The ease at which different types of particles can be added to Hydra to represent different fluids/solid boundaries shows that

the inclusion of multiphase flows would not be problematic for Hydra. Oil spills are an obvious starting point here. Despite the 2D nature of the flooding simulations presented here, 3D was also demonstrated and this should be expanded upon. From a pure software coding point of view, additional features that could provide potential range of simulations possible with Hydra include the addition of turbulence models and the parallelisation of the code. This would be a crucial modification if large particle number 3D simulations were required.

With regard to the experimental work performed a problem in the experimental design was discovered where the opening of the gate affected the end results. A repeat of the dambreak experiment with an improved (perhaps gateless or automated) gate opening system would be beneficial in further validating Hydra.

References

Abbott, M. B. and Ionescu, F. (1967). *"On the Numerical Computation of Nearly Horizontal Flows"*. Journal of Hydraulic Research 5: 97-117.

Abdolmaleki, K., Thiagarajan, K. P. and Morris-Thomas, M. T. (2004). *"Simulation of The Dam Break Problem and Impact Flows Using a Navier-Stokes Solver"*. 15th Australasian Fluid Mechanics Conference.

Agertz, O., Moore, B., Stadel, J., Potter, D., Miniati, F., Read, J., Mayer, L., Gawryszczak, A., Kravtsov, A., Nordlund, Å., Pearce, F., Quilis, V., Rudd, D., Springel, V., Stone, J., Tasker, E., Teyssier, R., Wadsley, J. and Walder, R. (2007). *"Fundamental differences between SPH and grid methods"*. Monthly Notices of the Royal Astronomical Society, Volume 380, Issue 3, pp. 963-978.

Alcrudo, F. and Garcia-Navarro, P. (1993). *"A High-Resolution Godunov-Type Scheme in Finite Volumes for the 2d Shallow-Water Equations"*. International Journal for Numerical Methods in Fluids 16(6): 489-505.

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H. and Grisoni, L. (2007). *"SOFA – an Open Source Framework for Medical Simulation"*.

Ansys, Inc. (2007). Ansys - Autodyn, version 11.0

BAE systems report. Last accessed: September 2009, http://baesystems.com/ProductsServices/ss_tes_atc_sph.html

BAE systems report 2. Last accessed: September 2009, http://www.baesystems.com/Newsroom/NewsReleases/2006/press_09032006.html

Barnes, J. and Hernquist, L. (1991). *"Fueling Starburst Galaxies With Gas-Rich Mergers".* Astrophys. J. Lett. 370, L65.

Belvedere, G., Lanzafame G. (2002). *"SPH Simulations of Spiral Shocks in Discs around Black Holes".* Publications of the Astronomical Society of Japan, Vol.54, No.5, p.781-785.

Belytschko, T., Lu, Y. and Gu L. (1994). *"Element-Free Galerkin Methods".* International Journal for Numerical Methods in Engineering, 37, pp. 229-256.

CFX Version11. http://www.ansys.com/products/fluid-dynamics/cfx/default.asp

Chen, Q., Dai, G. and Liu, H. (2002). *"Volume of fluid model for turbulence numerical simulation of stepped spillway overflow".* J. Hydraul. Eng. 128(7): 683-688.

Cheng, A. and Cheng D. (2005). *"Heritage and early history of the boundary element method".* Engineering Analysis with Boundary Elements, 29, 268-302.

Cleary, P., Prakash, M., Ha, J., Sinnott, M., Nguyen, T. and Grandfield, J. (2004). *"Modeling of Cast Systems Using Smoothed-Particle Hydrodynamics",* JOM, March 2004, 67-70.

Cnet.com article. Last accessed September 2009. http://news.cnet.com/Supercomputer-simulates-black-hole-collision/2100-11397_3-6062605.html

Cologrossi, A. and Landrini, M. (2003). *"Numerical simulation of interfacial flows by smoothed particle hydrodynamics".* Journal of Computational Physics, vol. 191.

Couchman, H.M.P., Pearce, F.R. and Thomas, P.A. (1997). "*Hydra Code Release".* astro-ph/9603116.

Crespo, A.J.C., Gomez-Gesteira, M. and Dalrymple, R.A. (2007). *"Boundary Conditions Generated by Dynamic Particles in SPH Methods".* Computers, Materials and Continua, Vol 5, no.3, pp 173-184.

Crespo, A.J.C., Gomez-Gesteira, M., and Dalrymple, R.A., (2007). *"3D SPH simulation of large waves mitigation with a dike".* Journal of Hydraulic Research, vol.45, no.5, pp 631-642.

Dalrymple, R. A. and Rogers, B. D. (2006). *"Numerical Modeling of Water Waves with the SPH Method."* Coastal Engineering 53: 141-147.

Dauptain, A. (2007). *"Brutus – Lid driven cavity test case CAVITY".*

Dotti M., Colpi M. and Haardt, F. (2007). *"Inspiral of double black holes in gaseous nuclear disks".* astro-ph/0602013.

Eason, G. (2000). *"Improved Turbulence Modelling for Computational Wind Engineering".* PhD thesis report, School of Civil Engineering, University of Nottingham, UK.

Engineeringtalk.com article. Last accessed: September 2009, http://www.engineeringtalk.com/news/bsy/bsy122.html

Evrard, A. E.; MacFarland, T. J.; Couchman, H. M. P.; Colberg, J. M.; Yoshida, N.; White, S. D. M.; Jenkins, A.; Frenk, C. S.; Pearce, F. R.; Peacock, J. A.; Thomas, P. A. (2002). *"Galaxy Clusters in Hubble Volume Simulations: Cosmological Constraints from Sky Survey Populations".* The Astrophysical Journal, Volume 573, Issue 1, pp. 7-36.

Faber, J. and Rasio, F. (2002). *"Post-Newtonian SPH calculations of binary neutron star coalescence. III. Irrotational systems and gravitational wave spectra".* Phys. Rev. D 65, 084042.

Floodrisk.org. http://www.floodrisk.org.uk

Frazão, S., Alcrudo, F., Mulet, J., Noël, B., Testa, G. and Zech, Y. (2007). *"The Impact European Research Project on Flood Propagation in Urban Areas: Experimental and Numerical Modelling of the Influence of Buildings on the Flow".* Flood Risk Management in Europe: 191-211.

Fread, D. (1980). *"DAMBRK: The NWS Dam-break Flood Forecasting Model".* Hydrologic Research Laboratory, National Weather Service, NOAA.

Frenk, C. S., White, S. D. M., Bode, P., Bond, J. R., Bryan, G. L., Cen, R., Couchman, H. M. P., Evrard, A. E., Gnedin, N., Jenkins, A., Khokhlov, A. M., Klypin, A., Navarro, J. F., Norman, M. L., Ostriker, J. P., Owen, J. M., Pearce, F. R., Pen, U. L., Steinmetz, M., Thomas, P. A., Villumsen, J. V., Wadsley, J. W., Warren, M. S., Xu, G. and Yepes, G. (1999). *"The Santa Barbara Cluster Comparison Project: A Comparison of Cosmological Hydrodynamics Solutions".* Astrophys. J. 525: 554-582.

Gazzola L., King E. J., Pearce F. R. and Coles P. (2007). *"The growth of baryonic structure in the presence of cosmological magnetic pressure".* MNRAS, 375, 657.

Gerardi, G., Molteni, D. and Teresi, V. (2005). *"3D SPH Simulations of Shocks in Accretion Flows around black holes".* arXiv:astro-ph/0501549v3.

Ghia, U., Ghia, K. N. and Shin, C.T. (1982). *"High Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method".* J. Comput. Phys. 43.

Gingold, R. and Monaghan, J. (1977). *"Smoothed particle hydrodynamics - theory and application to nonspherical stars"*. Monthly Notices of the Royal Astronomical Society, 181: 375-389.

Gittins, D., Clarke, C. and Bate, M.(2003). *"Hydrodynamical simulations of a cloud of interacting gas fragments"*. Mon. Not R Astron Soc.

Godunov, S. K. (1959). *"A Finite Difference Method for the Numerical Computation and Discontinuous Solutions of the Equations of Fluid Dynamics"*. Mat. Sb. 47: 271-306.

Gomez-Gesteira, M. and Dalrymple, R. (2004). *"Using a Three-Dimensional Smoothed Particle Hydrodynamics Method for Wave Impact on a Tall Structure"*. Journal of Waterway, Port, Coastal and Ocean Engineering, 130(2), 63-69.

Gomez-Gesteira, M., Cerqueiro, D., Crespo, C. and Dalrymple, R. (2004). *"Green water overtopping analyzed with a SPH model"*. Journal of Ocean Engineering, vol 32, pp. 223–238.

Goodwin, P. R., Matthew, G. S. and Wright, N. G. (1989). *"Prediction of Flood Wave Propagation Due to Dam Failure Using the Preissmann Scheme"*. Int Conf on Hydraulic and Environmental Modelling of Coastal, Estuarine and River Waters, Bradford, UK, Gower Technical Press.

Groenenboom, P. (2008). *"New features and applications of the hybrid SPH/FE approach in PAM-CRASH"*. ERCOFTAC SIG SPHERIC IIIrd International Workshop.

Ha, J., Cleary, P., Prakash, M., Alguine, V., Nguyen, T., and Scott, C. (2003). *"SPH MAGMAsoft and water analogue modellings of die filling of a servo piston"*. Proc. 3rd Int. Conf. on CFD in Minerals & Process Industries, Melbourne Australia.

Hager, W. and Schwalt, M. (1994). *"Broad Crested Weir"*. J. Irrigation and Drainage, 120(1): 13-26.

Hargreaves, D. M., Morvan, H. P. and Wright, N. G. (2007). *"Validation of the Volume of Fluid Method for Free Surface Calculation: The Broad-Crested Weir"*. Engineering Application of Computational Fluid Mechanics 1(2): 136-146.

Hirt, C. W. and Nichols, B. D. (1981). *"Volume of Fluid Methods for the Dynamics of Free Boundaries"*. Journal of Computational Physics 39: 201-225.

HR Wallingford report (2007). *"Use of physical modelling measurements to guide multi-phase simulations for wave/structure interaction"*.

Hu, X. and Adams, N. (2006). *"A multi-phase SPH method for macroscopic and mesoscopic flows"*. J. Comput. Phys., 213: 844.

Issa, R. (2005). *"Numerical assessment of the smoothed particle hydrodynamics gridless method for incompressible flows and its extension to turbulent flows"*. PhD thesis report, Department of Mechanical, Aerospace and Manufacturing Engineering, University of Manchester, Manchester, UK.

Jones, D. A. and Belton, D. (2006). *"Smoothed Particle Hydrodynamics: Applications within DSTO"*. Maritime Platforms Division Defence Science and Technology Organisation.

Kay, S. T., Pearce, F. R., Jenkins, A., Frenk, C. S., White, S. D. M., Thomas, P. A. and Couchman, H. M. P. (2000). *"Parameter tests within cosmological simulations of galaxy formation"*. Monthly Notices of the Royal Astronomical Society, 316: 374-394.

Kay, S. T., Pearce, F. R., Frenk C. S. and Jenkins, A. (2002). *"Including star formation and supernova feedback within cosmological simulations of galaxy formation".* MNRAS, 330, 113.

Kitsionas, S., Whitworth, A. and Klessen, R. (2007). *"SPH simulations of star/planet formation triggered by cloud-cloud collisions".* Proceedings of the International Astronomical Union (2007), 3:271-278.

Koshizuka, S. and Oka, Y. (1996). *"Moving particle semi implicit method for fragmentation of compressible fluid".* Nuclear Science Engineering, 123: 421-434.

Liem, R. and Köngeter, J. (1999). *"The influence of initial flow conditions on the propagation of dam break waves".* Proc. of XXVIII IAHR Congress, Graz, Austria.

Liu, M. B., Liu, G. R., Lam, K. Y. and Zong, Z. (2003). *"Smoothed particle hydrodynamics for numerical simulation of underwater explosion".* Computational Mechanics, 30, pp. 106–118.

Liu, G. R. and Liu, M. B. (2003). *"Smoothed particle hydrodynamics: a meshfree particle method".* World Scientific.

Lucy, L. B. (1977). *"A numerical approach to the testing of the fission hypothesis".* Astron. J., 82 (12), 1013-1024.

Muanwong, O., Thomas, P. A., Kay, S. T. and Pearce F. R. (2002). *"The effect of cooling and preheating on the X-ray properties of clusters of galaxies".* MNRAS, 336, 527.

Martin, J. C. and Moyce, W. J. (1952). *"Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane".* Philosophical Transactions of the Royal

Society of London. Series A, Mathematical and Physical Sciences, Vol. 244, No. 882, pp. 312-324.

Matsui, H., Habe, A. and Saitoh, T. (2007). *"Supermassive Black Hole Binary and Nuclear Star Burst".* R. S. de Jong, Island Universes, 387–390.

Maxwell, T. T. (1977). *"Numerical Modelling of Free Surface Flows".* PhD Thesis report, Imperial College, UK.

McCarthy, I., Bower, R., Balogh, M., Voit, M., Pearce, F., Theuns, T., Babul, A., Lacey, C. and Frenk, C. (2007). *"Modelling Shock Heating in Cluster Mergers: I. Moving Beyond the Spherical Accretion Model".* Mon. Not. Roy. Astron. Soc. 376: 497-522.

Moes, N., Dolbow, J. and Belytschko, T. (1999). *"A Finite Element Method for Crack Growth without Remeshing".* International Journal for Numerical Methods in Engineering Volume 46 (1): 131-150.

Monaghan, J. J. and Lattanzio, J. C. (1985). *"A Refined Particle Method for Astrophysicsal Problems".* Astronomy and Astrophysics, 149, 135.

Monaghan, J. J. (1989). *"On the problem of penetration in particle methods".* Journal of Computational Physics, 82:1-15.

Monaghan, J. J. (1992). *"Smoothed particle hydrodynamics".* Ann. Rev. Astron. Astrophys., 30, 543-574.

Monaghan, J. J. (1994). *"Simulating Free Surface Flows with SPH".* Journal of Computational Physics, 110(2), 399-406.

Monaghan, J. J. and Kos, A. (1999). "*Solitary Waves on a Cretan Beach".* Journal of Waterway, Port, Coastal and Ocean Engineering, 125(3), 145-154.

Monaghan, J. J. (2002). *"SPH compressible turbulence".* Advanced Methods for Computational Fluid Dynamics, 335: 843-852.

Monaghan, J. J., Kos, A. and Issa, R. (2003). *"Fluid Motion Generated by Impact".* Journal of Waterway, Port, Coastal and Ocean Engineering, 129, 250-259.

Monaghan, J.J. (2005). *"Smoothed Particle Hydrodynamics".* Rep. Prog. Phys., 68, 1703.

Morris, J. P. (1996). PhD thesis, Monash University, Melbourne, Australia.

Morris, M. (2000), CADAM Final Report, European Comission: 60 p.

Muller, M., Schirm, S. and Teschner, M. (2004). *"Interactive Blood Simulation for Virtual Surgery Based on Smoothed Particle Hydrodynamics".* Technology and Health Care, 12(1), 25-31.

Müller, M., Solenthaler, B., Keiser, R. and Gross, M. (2005). *"Particle-Based Fluid-Fluid Interaction".* Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2005).

Navarro, J. F. and Benz, W. (1991). *"Dynamics of cooling gas in galactic dark halos".* ApJ, vol. 380, 320-329.

Pearce, F. R., Jenkins, A., Frenk, C. S., White, S. D. M., Thomas P. A., Couchman, H. M. P., Peacock, J. A. and Efstathiou, G. (2001). *"Simulations of galaxy formation in a cosmological volume".* MNRAS, 326, 649.

Pfrommer, C., Springel, V., Enßlin, T. and Jubelgas, M. (2006). *"Detecting shock waves in cosmological smoothed particle hydrodynamics simulations".* Mon. Not. R. Astron. Soc, 367, 113.

Preissmann, A. (1961). *"Propagation des Intumescences dans les Canaux et Rivieres".* First Congress of the French Assocation for Computation, Grenoble, France.

Preliminary report, university of Lisbon. Last accessed: September 2009, http://www.dec.fct.unl.pt/projectos/impacto/Public_Papers/Report%20on%20Ceramic.pdf

Randles, P. and Libersky, L. (1996). *"Smoothed Particle Hydrodynamics: Some recent improvements and applications".* Comput. Methods Appl. Mech. Engineering. 139 375-408.

Ritchie, B. and Thomas, P. (2001). *"Multiphase Smoothed Particle Hydrodynamics".* Mon. Not. R. Astron Soc. 323, 743-756.

Sanders, B. F. (2001). *"High-Resolution and Non-Oscillatory Solution of the St. Venant Equations in Non-Rectangular and Non-Prismatic Channels".* Journal of Hydraulic Research 39(3): 321-330.

Sedov, L. I. (1959). *"Similarity and Dimensional Methods in Mechanics".* New York: Academic Press, 1959.

Shao, S. (2006). *"Incompressible SPH simulation of wave breaking and overtopping with turbulence modelling".* Int. J. Numer. Meth. Fluids, 50, 597–621.

Sleigh, P. A., Gaskell, P. H., Berzins, M. and Wright, N. G. (1998). *"An Unstructured Finite-Volume Algorithm for Predicting Flow in Rivers and Estuaries"*. Computers & Fluids 27(4): 479-508.

Sod, G. A. (1978). Journal of Computational Physics, 27, 1.

SPHysics code v1.4, http://wiki.manchester.ac.uk/sphysics

Stevens, D., Power, H. and Morvan, H. (2008). *"An order-N complexity meshless algorithm based on a local Hermitian interpolation"*. FERREIRA, A., ed. ECCOMAS Special Issue on Meshless Methods. Springer.

Swri.org article. Last accessed September 2009.
http://www.swri.org/3pubs/brochure/d20/CFDoilgas/CFDOil&Gas.pdf

Tasker, E., Brunino, R., Mitchell, N., Michielsen, D., Hopton, S., Pearce, F., Bryan, G., Theuns, T. (2008). *"A test suite for quantitative comparison of hydrodynamics codes in astrophysics"*. Mon. Not. R. Astron. Soc. Volume 390, Issue 3, pp. 1267-1281.

Thacker, R. J., Tittley, E. R., Pearce, F. R., Couchman, H. M. P. and Thomas, P. A. (2000). *"Smoothed Particle Hydrodynamics in cosmology: a comparative study of implementations"*. Monthly Notices of the Royal Astronomical Society, 319: 619-648.

Veen, D. J. and Gourlay, T. P. (2008). *"SPH Study of High Speed Ship Slamming"*. ERCOFTAC SIG SPHERIC IIIrd International Workshop.

Versteeg, H. K. and Malalasekera, W. (1995). *"An Introduction to Computational Fluid Dynamics – The Finite Volume Method"*. Addison Wesley Longman Ltd.

Vignjevic, R. (2004). *"Review of Development of the Smooth Particle Hydrodynamics (SPH) Method".* Proceedings of 6th DCSS.

Violeau, D. and Issa, R. (2007). *"Numerical modelling of complex turbulent free surface flows with the SPH method: an overview".* Int. J. Numer. Meth. Fluids, 53: 277–304.

Violeau, D., Piccon, S. and Chabard, J. P. (2001). *"Two Attempts of Turbulence Modelling in Smoothed Particle Hydrodynamics".* Proceedings of the 8th symposium on flow modelling and turbulence measurements. Advances in fluid modelling and turbulence measurements. World scientific: Singapore, 339-346.

Virgo Consortium http://www.virgo.dur.ac.uk/index.php?subject=millennium

Wada, K., Kokubo, E. and Makido, J. (2006). "*High-Resolution Simulations of a Moon-Forming Impact and Post-Impact Evolution".* Astrophys. J. 638, 1180-1186.

Wahl, T., Repogle, J., Wahlin, B. and Higgs, J. (2000). *"New developments in design and application of long throated flumes"*. Joint conference on water resource engineering and water resources planning and management. Minneapolis.

Wiersma, R., Schaye, J., Theuns, T., Vecchia, C. and Tornatore, L. (2009). *"Chemical enrichment in cosmological, smoothed particle hydrodynamics simulations".* Mon. Not. Royal Astron. Soc., July 2009.

Yokohama press report. Last accessed: February 2009, http://www.yrcpressroom.jp/ir_en/report/AR/AR2002.pdf

Appendix A

Hydra kernel normalisation

To normalise a kernel the main part of the kernel must be integrated over its range and the integrals added together. It is crucial to remember that the normalisation factor of the kernel (the first part common to all the lines) is different depending on the number of dimensions being simulated in. If say, a simulation is performed in 2D and then repeated in 3D the kernel normalisation factor must be altered or the solution will be incorrect. The kernel (W) used in Hydra has been reproduced here:

$$W(r,h) = Normalistionfactor\begin{cases} 4-6x^2+3x^3 & 0 \le x \le 1 \\ (2-x)^3 & 1 \le x \le 2 \\ 0 & else \end{cases}$$

Where $x = \frac{r}{h}$ is the ratio of the particle separation to the smoothing length.

When normalising in one dimension the integral is taken over the range zero to two, indicated by the conditional limits imposed on the kernel, with no additional parameters, i.e.

$$\int_0^2 W.dx$$

The full calculation in detail:

$$\int_0^1 4 - 6x^2 + 3x^3 .dx + \int_1^2 (2-x)^3 .dx$$

$$\int_0^1 4 - 6x^2 + 3x^3 .dx + \int_1^2 8 - 12x + 6x^2 - x^3 .dx$$

$$4x - 2x^3 + \frac{3}{4}x^4 + 8x - 6x^2 + 2x^3 - \frac{x^4}{4}$$

$$2\frac{3}{4} + 4 - 3\frac{3}{4} = 3$$

However when normalising the kernel in one dimension the result must be doubled to account for both sides of the particle. This is equivalent to integrating over the range -2 to 2 instead of 2 to 0. Therefore the result is 6. The normalisation factor is one over the integral.

→ Normalisation factor is $\frac{1}{6}$

Normalisation in 2 dimensions takes the form:

$$\int_0^2 W \times 2\pi r.dx$$

The normalisation in detail is as follows

$$\int_0^1 (4 - 6x^2 + 3x^3) \times 2\pi x.dx + \int_1^2 (2-x)^3 \times 2\pi x.dx$$

$$\int_0^1 8\pi x - 12\pi x^3 + 6\pi x^4 + \int_1^2 (8 - 12x + 6x^2 - x^3) \times (2\pi x).dx$$

$$4\pi x^2 - 3\pi x^4 + \frac{6\pi}{5}x^5 + 8\pi x^2 - 8\pi x^3 + 3\pi x^4 - \frac{2}{5}\pi x^5$$

$$\frac{11}{5}\pi + \frac{3}{5}\pi = \frac{14}{5}\pi$$

In two dimensions there is no requirement to double the result so taking the 1 over the integral provides

→ Normalisation factor is $\frac{5}{14\pi}$

This is the normalisation used throughout this thesis as all of the simulations were carried out in two dimensions.

For completeness the normalisation of Hydra's kernel in 3 dimensions has also been performed. When normalising in three dimensions the integral takes the form:

$$\int_0^2 W \times 4\pi r^2 .dx$$

The calculation in full:

$$\int_0^1 (4 - 6x^2 + 3x^3) \times 4\pi x^2 .dx + \int_1^2 (2 - x)^3 \times 4\pi x^2 .dx$$

$$\int_0^1 16\pi x^2 - 24\pi x^4 + 12\pi x^5 + \int_1^2 (8 - 12x + 6x^2 - x^3) \times (4\pi x^2).dx$$

$$\left(\frac{16\pi x^3}{3} - \frac{24\pi x^5}{5} + 2\pi x^6\right) + \left(\frac{32\pi x^3}{3} - 12\pi x^4 + \frac{24\pi x^5}{5} - \frac{2}{3}\pi x^6\right)$$

$$2.53\pi + 1.47\pi = 4\pi$$

Taking one over the integral gives

→ Normalisation factor is $\frac{1}{4\pi}$

So the full kernel used in a 3D simulation would be written as:

$$W(r,h) = \frac{1}{4\pi}\begin{cases} 4 - 6x^2 + 3x^3 & 0 \le x \le 1 \\ (2-x)^3 & 1 \le x \le 2 \\ 0 & else \end{cases}$$

Appendix B

Calculation of time taken to drain water tank for wall hole test

In order to calculate the time taken to drain water through a hole in a tank the principles of conservation of mass and the energy equation to relate the velocities at the outlet and the rate of change of the free surface.

Take the flow area at the tank hole to be $C_c d^2$

where C_c is the coefficient of contraction and assumed to be 0.6.

The ideal flow velocity at the hole (vi) is given by

$$\left(\frac{V^2}{2g} + h \right) = \frac{v_i^2}{2g}$$

$$v_i = \sqrt{V^2 + 2gh}$$

Where V is the velocity of the flow at the free surface, g is gravity and h is the height of the free surface above the hole.

But the real velocity at the hole (v) must be multiplied by a coefficient of velocity Cv which is assumed to be 0.99 Therefore

$$v = C_v \sqrt{V^2 + 2gh}$$

$$v^2 = C_v^2 \left(V^2 + 2gh \right)$$

From continuity the rate of change of the level of the free surface must be related to the velocity of the jet at the hole by

$$VD^2 = vd^2 C_c$$

Combining these equations gives us:

$$V^2\left(\frac{D^4}{C_c{}^2 d^4}\right) = C_v^2\left(V^2 + 2gh\right)$$

$$V^2\left(\frac{D^4}{C_v{}^2 C_c{}^2 d^4} - 1\right) = 2gh$$

where D is the diameter of the tank

Therefore

$$V = \frac{d^2}{D^2} C_v C_c \sqrt{2gh}$$

By using the fact that

$$V = -\frac{dh}{dt}$$

It becomes apparent that

$$-\frac{dh}{dt} = \frac{d^2}{D^2} C_v C_c \sqrt{2gh}$$

Integrate between t = 0 and T and integrate between h = H and h = 0

$$\int_H^0 -\frac{dh}{\sqrt{h}} = \frac{d^2}{D^2} C_v C_c \sqrt{2g} \int_0^T dt$$

We can produce a formula to calculate the time taken (T) to drain the tank

$$T = \frac{2D^2}{d^2 C_v C_c}\sqrt{\frac{H}{2g}}$$

$$T = \frac{2 \times 1^2}{0.3^2 \times 0.99 \times 0.6}\sqrt{\frac{2}{2 \times 9.81}}$$

$$T = 11.8 \text{ s}$$