Simulating Breaking Waves Using Smoothed Particle Hydrodynamics

Gareth Llewellyn Vaughan

A thesis submitted for the degree of
Doctor of Philosophy
at the University of Waikato, Hamilton
New Zealand

June 8, 2005

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Abstract

Detailed computational modelling of fluid motion in the surf zone (such as under breaking waves) is difficult, because model domains must be long and narrow, and very high resolutions are required to obtain sufficiently detailed simulations. Existing modelling methods have deficiencies: one class of methods are restricted to simulating waves only to the point where the lip of the wave hits the water, while a second class experience significant numerical errors. A third class of methods are the Lagrangian particle-based methods, and while some of these have been applied to simulating breaking waves, their general usefulness for such work is not yet established.

Here a model is created (named MARIAN) using smoothed particle hydrodynamics (SPH), an example of a Lagrangian particle-based method. This model is tested, and then used to simulate a number of processes relevant to coastal modelling, namely the propagation of waves in a numerical wave tank, the breaking of solitary waves on plane slopes, and the breaking of cnoidal waves over a real bathymetry. From comparisons of the results, merits and demerits in the model and method can be identified, and general conclusions as to how useful the model is for the task can be established. It is found that while some simulations are very good (a bursting dam, for example), and others are average (a seiching basin), some simulations important for work in the coastal zone are not good (the propagation of waves in a numerical wave tank) and others are very poor (breaking waves). Clearly, there are limits as to what MARIAN can be used to simulate, and what these limits are is not apparent.

From the model developed in this study it should be concluded that Lagrangian particle-based models designed for simulating breaking waves have shortcomings, and based on this work, other methods would currently be preferred. However other researchers have recently had more success by using variations on the methods employed here. In particular by using better viscosity formulations they resolved breaking waves reasonably well. In the work performed here it was ascertained that some methods used in the literature cause SPH models to fail in enforcing the fundamental principles of physics. While the current state of the model created here is unsatisfactory, there are promising improvements that could be made, and further research is likely to resolve some of the problems discussed here. Topics that would benefit from further research include finding ways to reduce runtimes and increase resolutions (by parallelizing source code, improving algorithms and identifying the free surface), and ways to increase accuracy (improving the quality of kernels employed, including the "missing boundary integrals", and ensuring consistency) while still enforcing the fundamental principles.

Acknowledgements

Finishing this thesis has only been possible with the help of many people. For the guidance provided by my supervisors Terry, Karin, Alfred and Richard I am especially grateful. The completion of my field work was only possible because of Dirk's contribution on the multitude of trips to Whangamata. I am appreciative for advice and suggestion coming from discussions with Alistair, Brett, Brad, Vern and Arne, whose suggestions initiated many improvements in many areas of this thesis. To my wife Marian, many thanks for support and for tolerating me in my pursuit for the completion of this thesis. I greatly appreciate the support given to me by The University of Waikato in awarding me a doctoral Scholarship. Finally, thanks to the examiners for their comments which inspired significant improvements.

A.M.D.G.

Contents

List of Figures

List of Tables

List of Acronyms

BEM	Boundary Element Method, see Section 2.2.2
CFD	Computational Fluid Dynamics
CSPH	Corrected SPH, see Section 3.3.
CXSPH	SPH using both the Corrected and "X" variations
FEM	Finite Element Method
GFDM	Generalised Finite Difference Method, see Section 2.2.3
LBIE	Local Boundary Integral Equation, see Section 3.1.2
MAC	Marker And Cell, see Section 2.2.4
MPS	Moving Particle Semi-implicit, see Sections 2.2.3 and 3.4.5
NOAA	National Oceanic and Atmospheric Administration, see Section 6.1
NZGM	New Zealand Map Grid, see Section 6.1.1
PSPH	Projection method for SPH, see Section 2.2.3
RTK GPS	Real Time Kinematic Global Positioning System, see Section 6.1.1
SPH	Smoothed Particle Hydrodynamics
SWAN	Simulating WAves Nearshore, see Section 6.1
VMOSS	Vertical Movement Only Settling Subroutine, see Section 4.1
VOF	Volume Of Fluid, see Section 2.2.1
WCSPH	Weakly Compressible SPH, see Section 2.2.3
XSPH	The "X" variant of SPH, see Section 3.2.3

Introduction

A New Way of Simulating Breaking Waves

Particle-based Lagrangian methods have significant advantages, and also disadvantages in comparison to other modelling techniques. They are not subject to numerical diffusion, and when simulating fluid flow, no special treatment is required to account for complex free surface flows, such as those generated by breaking waves. Before 2000 relatively little had been published on the application of these methods to free-surface fluid-flows (Monaghan, 1994; Koshizuka *et al.*, 1995; Monaghan and Kos, 1999), but this is changing rapidly. While significant advances have been made recently (Lombardi *et al.*, 1999; Morris, 2000; Bonet and Kulasegaram, 2001; Rabczuk and Xiao, 2004), there may be important features of such techniques that remain to be discovered.

In comparison, Eulerian methods require special techniques to account for the free surface (the Volume of Fluid technique, for example, Guignard *et al.*, 2001), do experience numerical diffusion (Fries and Matthies, 2003), and also require long run-times, but have been used for creating high resolution simulations of breaking waves (Chen *et al.*, 1999; Guignard *et al.*, 2001). Yet another class, those models based on potential flow theory, are limited to irrotational flows in simply connected domains, and when simulating breaking waves they must stop when the lip has hit the water in front of the wave (Svensden and Grilli, 1990). However they can resolve the free surface as a sharp interface and have typically short run-times (Grilli *et al.*, 1989). The question as to whether particle-based Lagrangian methods are suitable for simulating processes within the coastal zone remains open.

The Rationale

The capability to simulate breaking waves is of some importance in the area of coastal oceanography where the issues of wave-induced erosion and its control (Komar, 1998), and the design of wave impact resistant coastal structures are significant (Dean and Dalrymple, 1991). With predictions of global climate change, and consequently higher sea levels, more coastal regions will become vulnerable to these processes in the future (Komar, 1998). Of increasing interest and importance in this area are artificial surf reefs which can be used to control coastal erosion and create breaks for surf board riding (Black and Andrews, 2001). In designing artificial surf reefs the accurate prediction of wave shape is paramount. The best modelling method for approaching such tasks will only become apparent through comparisons of the performance and accuracy of the various techniques.

The Aim

The primary aim of this project is to investigate the suitability of a particle-based Lagrangian model for applications in studies of the surf-zone, in particular to simulating breaking waves. This needs to be established by comparison to existing models that use other methods, with an emphasis on the accuracy they can achieve and how quickly simulations can be completed.

Research Objectives

A review concentrating on the physical processes of the surf-zone and on existing computational models of breaking waves is undertaken as a background to understanding the issue before developing a particle-based Lagrangian model. Numerical methods in fluids are briefly discussed as will the mechanics of particle-based Lagrangian methods. This is done to establish the standard for numerical simulations in the area, and gain an understanding of the merits and demerits of the various methods. These reviews constitute Chapters 2 and 3.

Chapter 4 presents a particle-based Lagrangian model that can be used to simulate breaking waves, the creation of which is a major objective of the work performed here. A fundamental concern is that the model should operate without significant programming errors. This is assessed by comparison to examples available in the literature. The tests conducted here are also described in Chapter 4.

Before waves can break on a beach they have to get there. An important test of the performance of this model is its ability to simulate the propagation of waves. In Chapter 5 the propagation of solitary and cnoidal waves in a numerical wave tank is investigated. Chapter 5 concludes with an examination of solitary waves breaking on plane beaches which have a variety of slopes. The results are critiqued and reasons for various aspects of the performance are suggested.

But coastal oceanography is concerned predominantly with real beaches, rather than idealised laboratory constructions. It may be that there are aspects of simulating waves breaking on real beaches that do not apply when simulating ideal laboratory scenarios. These issues need to be established if an assessment is to be made as to how useful a particle-based Lagrangian model can be. To this end a case study is made of waves breaking on a real-world beach,namely Whangamata bar. Parameters describing the "barrels" of waves breaking on the bar are determined from photographs and the methods used to do this are presented in Chapter 6. The data collected from Whangamata bar will be compared to a number of SPH simulations of a breaking wave, constructed using the bathymetry of Whangamata bar, also presented in Chapter 6.

In the final chapter, conclusions are drawn as to the suitability of the model created here for simulating processes in the coastal zone. Also, new publications are frequently becoming available describing improvements to methods and techniques that could be employed here, and some of these have the potential to greatly enhance the quality of the model created here. These are discussed, suggesting future research stemming from the work conducted here.

A Brief Note on Terminology

In this thesis the term *runtime* refers to the time taken in the real-world for a simulation to run to completion, while simulation duration refers to the time internal to the simulation. For example a wave might take 60 seconds to complete the breaking process so a simulation of this wave would have a *simulation duration* of 60 seconds but might take two weeks to run to completion so the runtime is two weeks. The notation $O(N)$ means that some property, a runtime, for example, scales with N, so doubling N doubles the runtime.

Chapter 2

Waves

The first step in achieving the aim of this thesis, the development of a particle-based Lagrangian model of a breaking wave, is establishing the current understanding of how and why waves break. In addition a rationale is required as to the appropriateness of the method for the task.

In this chapter the existing understanding of breaking waves is reviewed. The goal of this chapter is to review the relevant scientific understanding of how waves break and to appraise the achievements of modellers who have simulated breaking waves.

Significant areas examined include the types and sources of waves, the methods of classifying breakers, the physics of breaking waves, the effect of bathymetry on wave breaking and the effect of wave breaking on the sea floor. Existing wave models are introduced and critiqued, with special attention being paid to their limitations and to the phenomena that they can account for.

2.1 How Waves Break

The way that waves break is strongly dependent on the nature of the waves and on the beach over which the waves are breaking (Komar, 1998). An understanding of how and why waves break is important for assessing the adequacy of the model developed in this thesis. It provides a framework for knowing what input waves are appropriate and how the wave should develop, as well as providing a first-order check for the correctness of any model.

Empirical descriptions of the ways in which waves break are described and the dimensionless numbers that have been used to parameterise this behaviour introduced. Then the effects of bathymetry and beach slopes are described.

2.1.1 Wind Waves and Swell

The waves that break on beaches are, for the most part, gravity waves generated by wind and are periodic with periods from 1 s to 20 s (Mei, 1989). Of no interest here are waves driven by the Coriolis effect or surface tension. Here a brief description of processes during the generation of waves and swell is given, as is a discussion of processes significant to waves and swell.

The Generation of Waves

Initially, wind waves appear as very short capillary waves, driven by surface tension, and provided the wind does not die, they soon evolve to become wavelets. These are very short, small waves driven by gravity. With the wind removed, capillary waves die away very quickly, much faster than wavelets do. If they continue to be driven by wind, wavelets increase in height till they are too high, and they break. The broken waves split into waves with a variety of wavelengths which

are also pumped by the wind. The longer waves can grown to greater heights, which they do, and then by the same process, are split into longer waves yet. Eventually limiting waves heights are met, and the waves can grow no more.

As they are being generated, wind waves are referred to as *sea*, notable for being highly complex (Komar, 1998), a state that can be understood as the simultaneous presence of many waves with quite different periods. After leaving the area of the ocean in which they were generated, a sea will transform to become a *swell*, typified by more organised waves with reasonably uniform heights and periods.

Dispersion

Waves on the surface of the ocean are *dispersive* meaning that the phase speed is dependent on wavelength (Kinsman, 1984), that is, longer waves travel faster. This is in contrast to non-dispersive light waves, for example, for which the speed is constant, irrespective of wavelength.

This means that once it has transformed from being a sea, a swell will continue to change as it propagates away from the area in which it was generated. The shorter the period of a wave, the lower its phase velocity so short waves lag behind those with longer periods (Komar, 1998). When waves generated by a storm arrive at a point on a coastline it is the long period waves that arrive first, and the time when the short period waves arrive is determined by how far away they were generated.

Viscosity

Also of significance is the reduction in wave height due to viscosity, an effect which damps the amplitudes of short period waves at a greater rate than it does long period waves (Komar, 1998). Viscosity is not important when waves are traveling only short distances, but over large distances it means longer period waves are more likely to survive the trip from the area they were generated than short period waves.

Wave Groups

So the larger the distance from the area of generation, the better a swell can be approximated by a single train of waves with constant period. Variations in the heights of different waves within a real wave train may be significant due to the presence of wave groups (Mei, 1989) or if the winds that generated the waves were unsteady. Wave groups are also termed *sets* by surfers, and are the cause of *surf beat* (Mei, 1989).

A wave group travels at the group velocity which is slower than the phase velocity Komar (1998). In the open ocean the group velocity is approximately half the phase velocity, while in shallow water it is approximately equal to the phase velocity. Individual wave peaks within a wave group in the open ocean can be observed forming at the trailing edge of a group, moving to the front, and then disappearing. Different wave groups consist of waves with different wavelengths and have different group velocities. Dispersion still applies and the groups consisting of longer waves travel faster.

2.1.2 Wave Theories

There are several different wave theories in common usage, these being the Airy, Stokes, cnoidal and stream function wave theories (Isobe, 1988; Dean and Dalrymple, 1991). These wave theories are useful under different conditions dependent on the depth of the water, the wave height and wave length. Figure 2.1 shows diagrammatically the areas of application for the various theories but it must be noted that these divisions were constructed giving most preference to the simplest

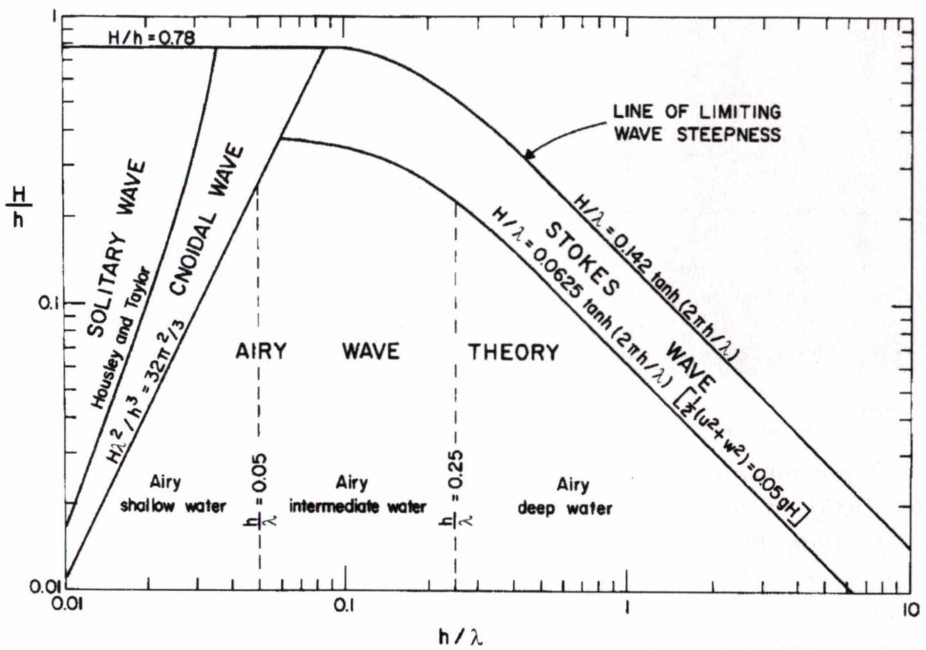

Figure 2.1: *Areas of application for the different wave theories as determined by the ratio H/h and H/λ (from Komar, 1998). Here H is the wave height, h is water depth and λ the wavelength.*

theory (Airy) and least to the most complex (cnoidal). Stream function wave theory (Dean and Dalrymple, 1991) is notably absent from this chart, excluded because it is not an analytic wave theory (Komar, 1998). There are other such diagrams (Isobe, 1988, for example) which take into account stream function wave theory and also higher order Stokes and cnoidal waves.

The functional forms of the free surface elevation derived from the different waves theories are given in Table 2.1.2. Examples are plotted in Figure 2.2.

The different wave theories describe waves with different profiles and are limited to waves in which certain assumptions apply. Airy waves (linear waves) are limited to waves for which the wave height is small in comparison to both the depth and wavelength (Komar, 1998), an assumption that does not apply within the surf zone where wave heights and water depths may be of the same order. By not making the small wave height assumption Stokes-wave theory can be derived (Komar, 1998), and incidentally, Airy-waves can equivalently be thought of as first order Stokes-waves. Stokes-waves have flatter troughs, higher peaks and steeper wave faces than Airy-waves. In very shallow water when wave heights are close to water depth the Stokes theory fails (Komar, 1998) and cnoidal wave theory must be used. Solitary waves are a special case, being a cnoidal wave for which the wavelength is infinite.

Stream function wave theory (Dean and Dalrymple, 1991) is a fully nonlinear numerical method for finding permanent wave forms given various problem parameters (namely the wave period and height, and the water depth). It requires the use of either tables or a computer, but does not experience limitations like those that constrain the analytic theories. It involves finding solutions to the governing equations that minimise the error in the free surface boundary condition (Dean and Dalrymple, 1991).

Permanent Waves

The different wave theories of the previous section describe waves that are periodic in space (Dean and Dalrymple, 1991). These waves are also referred to as being *permanent*, that is, they do not change shape during propagation (Isobe, 1988). The derivations of these wave forms

Theory	Surface Elevation	θ
Airy	$\eta(x,t) = \frac{H}{2}\sin(\theta)$	$\frac{2\pi}{\lambda}(x-ct)$
2nd order Stokes	$\eta(x,t) = \frac{H}{2}\cos(\theta) + \frac{3H^2\lambda^2}{64\pi^2h^3}\cos(2\theta)$	$\frac{2\pi}{\lambda}(x-ct)$
1st order Cnoidal	$\eta(x,t) = y_t - h + H\mathrm{cn}^2(\theta,m)$	$\frac{2K(m)}{\lambda}(x-ct)$
Solitary	$\eta(x,t) = H\mathrm{sech}^2(\theta)$	$\left(\frac{3H}{4h^3}\right)^{\frac{1}{2}}(x-ct)$

Table 2.1: *Expressions for the free surface elevations for the different waves theories (Goring, 1978; Isobe, 1988). $K(m)$ is the complete elliptic integral of the first kind and $\mathrm{cn}(\theta,m)$ is one of the Jacobi elliptic functions, both in terms of the elliptic parameter m. x is the horizontal coordinate in space, η the deviation in surface elevation from the still water level, c is the phase velocity, y_t is the distance from the sea floor to the trough of the wave and other variables are as before. Units are not important here provided all variables and constants are consistent.*

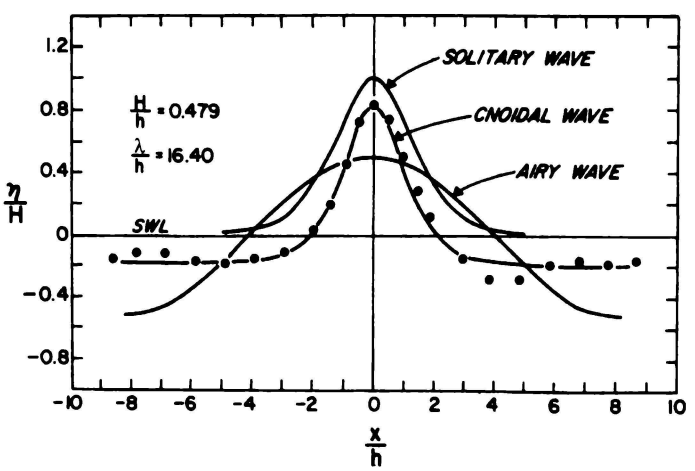

Figure 2.2: *Examples of wave profiles generated from the different wave theories (from Komar, 1998). SWL means still water level.*

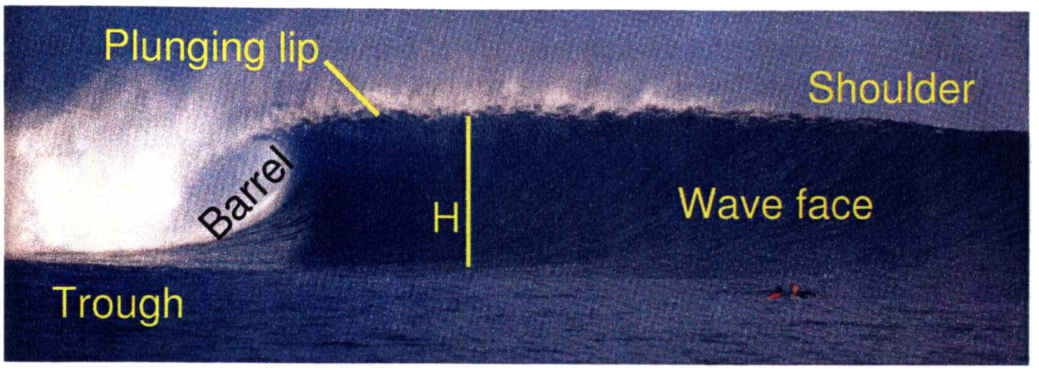

Figure 2.3: *Terminology for parts of breaking waves. Here H is the height of the wave. (Photo: Waves, EMAP Australia Pty Ltd.)*

require that viscosity and friction (at the seabed) be neglected and that certain other assumptions can be justified, with different sets of assumptions being required to derive different permanent waves. Waves that are not permanent can be constructed as the sum of a number of permanent waves, where each permanent wave can have unique periods and heights (Dean and Dalrymple, 1991). The constituent waves travel with different phase velocities and so the constructed wave will experience dispersion (Dean and Dalrymple, 1991). Generally, waves in deep water are well approximated by Airy waves while in shallow water cnoidal waves fit better (Komar, 1998).

2.1.3 Types of Breakers

As a wave approaches a coastline the depth of the water decreases and the wave experiences a substantial change in shape. Exactly how the waves shape changes is dependent on several factors: the initial shape of the wave; the depth of the water and the slope of the sea floor; the strength and direction of local winds and currents (Douglass, 1990); and other bathymetric features (Mead, 2000). The effects of winds and currents are not examined here.

In this section empirical observations of how waves break are recounted and the reasons why waves break are outlined. The existing methods of classifying breaking waves are examined and the effect of bathymetric features investigated.

Classifying How Waves Break

The exact manner by which waves break can vary considerably between different waves and from beach to beach. However waves on like beaches break in the same manner (Svensden and Putrevu, 1995). The most common method of classifying breaking waves is by the use of the qualitative descriptors *spilling*, *plunging*, *collapsing* and *surging* (Aagard and Masselink, 1999), representing bands in a continuous spectrum. Consequently the use of these descriptors can be a little subjective since divisions between the bands require arbitrary definition, a issue that means classifying particular waves is difficult (Komar, 1998).

As a wave approaches a beach its *face* (see Figure 2.3 for definitions) will steepen, creating a *lip* that will eventually be propelled beach-ward from the face of the wave, so creating a *spilling* or *plunging* wave. The lip of the wave will proceed to impact either on the face of the wave or the trough preceding the wave.

As the lip is projected from the crest a barrel is formed. For plunging waves the falling lip may be streaked with foam, air entrained in the flow, and once the lip has hit the water a considerable amount of air is entrained within the turbulent flow that has been generated. For

spilling waves the process is the same but the lip is more likely to impact high on the wave face (so the barrel is much smaller) and a turbulent flow of foam down the face occurs (Mei, 1989).

However some waves neither spill nor plunge, instead they may collapse or surge, the latter representing a fully reflective beach. Surging waves initially form like plunging waves but the base of the wave will accelerate up the beach slope creating a foamy flow (Komar, 1998).

Mead and Black (2001) note that while the majority of waves that are of interest to surfers are plunging waves, there is a range of waves that fall into this category. The differences between any two plunging waves can include the shape of the wave face and the orientation and shape of the barrel. The size and shape of the lips of different plunging waves may also vary.

The Cubic Curve of Longuet-Higgins

Using potential theory Longuet-Higgins (1982) developed analytic expressions describing the barrel shape and orientation of plunging waves. This approach provides a method to parameterise the way that a wave breaks but it cannot be applied to waves with distorted faces, such as that shown in Figure 2.4. Longuet-Higgins (1982) and also Mead and Black (2001) have fitted these curves to digitised points around barrels in photographs of breaking waves (from both the laboratory and real-world). These photographs were selected for being as close as possible to profile shots of the breaking waves (see Figure 2.5).

Longuet-Higgins' cubic curve (Longuet-Higgins, 1982) is given in terms of the free parameter μ and the Cartesian coordinates x and y

$$\begin{aligned} \frac{x}{a} &= 3\mu^2 - \frac{1}{3} \\ \frac{y}{b} &= -\mu^3 + 2\mu \end{aligned} \tag{2.1}$$

and is plotted in Figure 2.6. The parameters a and b allow independent scaling of the cubic curve in two dimensions so the cubic can be matched to the physical vortex dimensions (w and l in Figure 2.5) of any barrel. The angle of the vortex can be found by a relatively simple transformation of coordinates.

The weakness in the approach used by (Mead and Black, 2001) to examine real-world breaking waves is the uncertainty in where the digitised points should be placed, a fairly subjective process. Also problematic is the uncertainty over whether the wave crest (at the position the curve is being placed) is parallel with the camera axis (Ranasinghe *et al.*, 2001). Indeed the approach may not work for all waves since in some cases the shoulder of the wave will obscure the view down the barrel, a consequence of refraction.

Worth noting are the $\sqrt{3}$ aspect-ratio ellipses (New, 1983, for example) which can be fitted to profiles of breaking waves in a similar manner to Longuet-Higgins' cubic. Examples of $\sqrt{3}$ aspect-ratio ellipses fitted to simulated waves are shown in Figure 2.13. However these ellipses do not provide as close a description of the breaking waves as Longuet-Higgins' cubics (New, 1983).

The Iribarren Number

The Iribarren number (also called the surf similarity parameter) evaluated at the break point (ξ_b) is given by (Mei, 1989)

$$\xi_b = \frac{s}{\sqrt{H_b/\lambda_\infty}} \tag{2.2}$$

where s is the beach slope, H_b is the height of the wave at the break point and λ_∞ is the open-ocean (deep water) wavelength. The Iribarren number (ξ_∞) in terms of the deep water wave

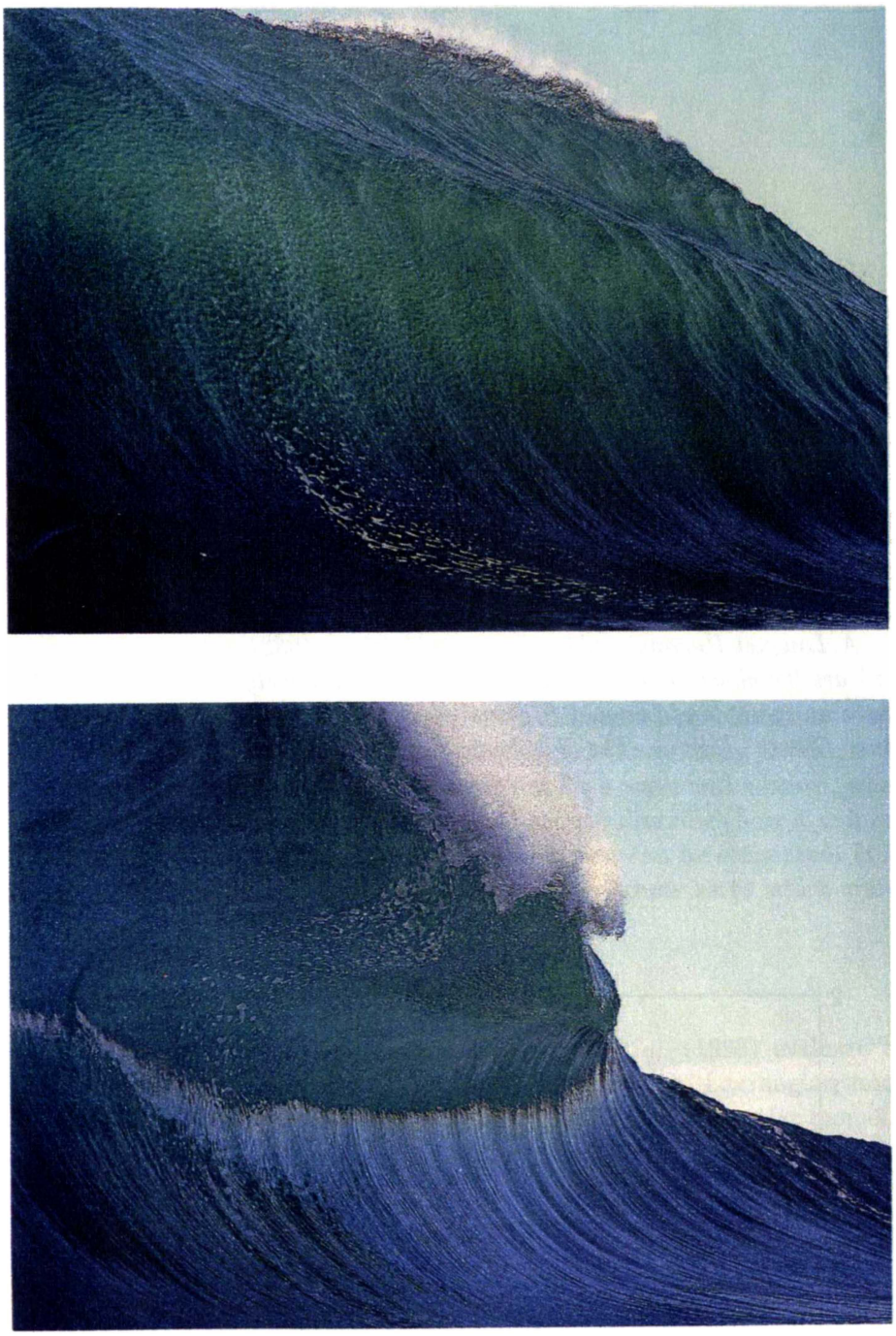

Figure 2.4: *A pair of images from a sequence (the top image came first) showing an extreme, barreling wave with a distorted face. A photo of this wave a short time later is shown in Figure 2.5. (Photo: Litmus, The Val Dusty Experiment Pty Ltd., Australia)*

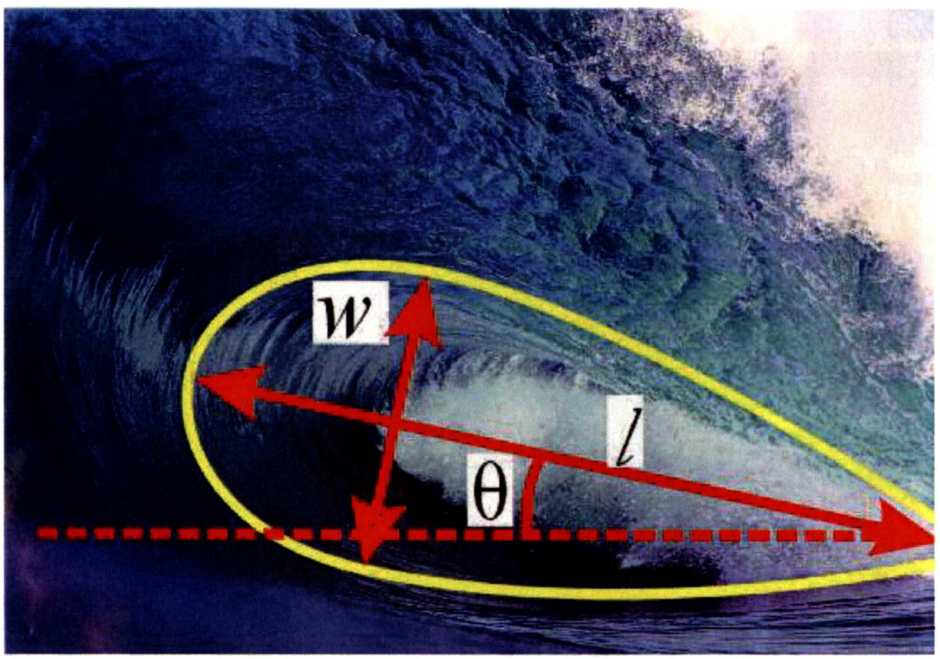

Figure 2.5: *A Longuet-Higgins' cubic (Longuet-Higgins, 1982) fitted to the barrel of a wave. Here* w *and* l *are the physical vortex width and length respectively, and* θ *is the vortex angle. The yellow curve is an example of Longuet-Higgins' cubic curve and the dashed red line is a horizontal reference line. (Photo: Litmus, The Val Dusty Experiment Pty Ltd., Australia)*

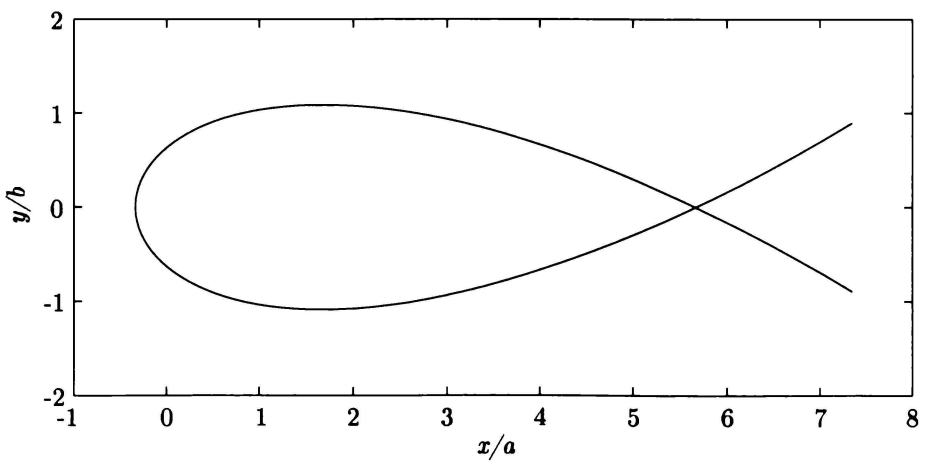

Figure 2.6: *Longuet-Higgins' cubic from Equation 2.1 (Longuet-Higgins, 1982).*

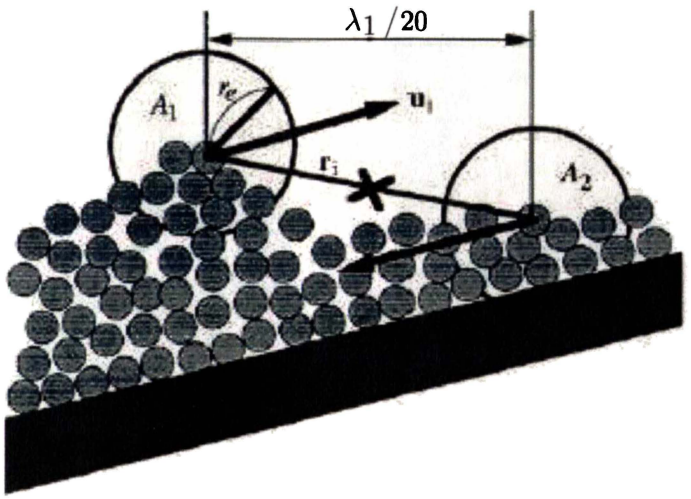

Figure 2.7: *The definition for angular momentum used by Koshizuka* et al. *(1998). The first point (at the centre of the area A_1) is placed at the top of the wave and the second point (at the centre of the area A_2) is on the free surface and is a distance $\lambda_\infty/20$ toward the beach from the first point. The angular momentum can then be found from Equation 2.4.*

height (H_∞) is given by

$$\xi_\infty = \frac{s}{\sqrt{H_\infty/\lambda_\infty}}. \tag{2.3}$$

Galvin (1968) and also Battjes (1974) correlated the Iribarren number against the descriptors (plunging, spilling, surging and collapsing) and established the limiting Iribarren numbers for each descriptor. So by determining the Iribarren number for a wave and a beach, whether or not a wave will break can be determined, the descriptor that best describes how it will break can be found and the *degree* to which a wave plunges, spills et cetera can be estimated. It is, however, an empirical result based on the subjective opinions of researchers as to which waves broke in which manner.

The Angular Momentum of Koshizuka et. al.

In order to classify breaking waves objectively, Koshizuka *et al.* (1998) evaluate the angular momentum when simulating breaking waves with their particle-based Lagrangian model in which they employ the moving particle semi-implicit method (MPS). Their angular momentum ($\boldsymbol{\omega}$) is defined by

$$\boldsymbol{\omega} = \frac{1}{N} \sum_{i \in \{A_1, A_2\}} \boldsymbol{r}_i \times \boldsymbol{u}_i. \tag{2.4}$$

and is depicted geometrically in Figure 2.7. Here A_1 and A_2 are areas defined by their radii r_e and the positions of their centres which are separated by $\boldsymbol{r}_i$ and $\boldsymbol{u}_i$ are the velocities of all N particles within A_1 and A_2.

A_1 is chosen to be at the top of the wave while A_2 is chosen (somewhat arbitrarily it seems, and without much discussion) to be $\lambda_\infty/20$ beach-wards of A_1 and apparently at the free surface (Koshizuka *et al.*, 1998). The angular momentum is defined as being about a particular point, in this case that point is half way between the centres of A_1 and A_2. This definition makes sense for some cases, but where the depth at the trough is small, errors may occur, in particular with collapsing and surging waves. In practice it will probably be quite dependent on the choice of the positions and sizes of A_1 and A_2.

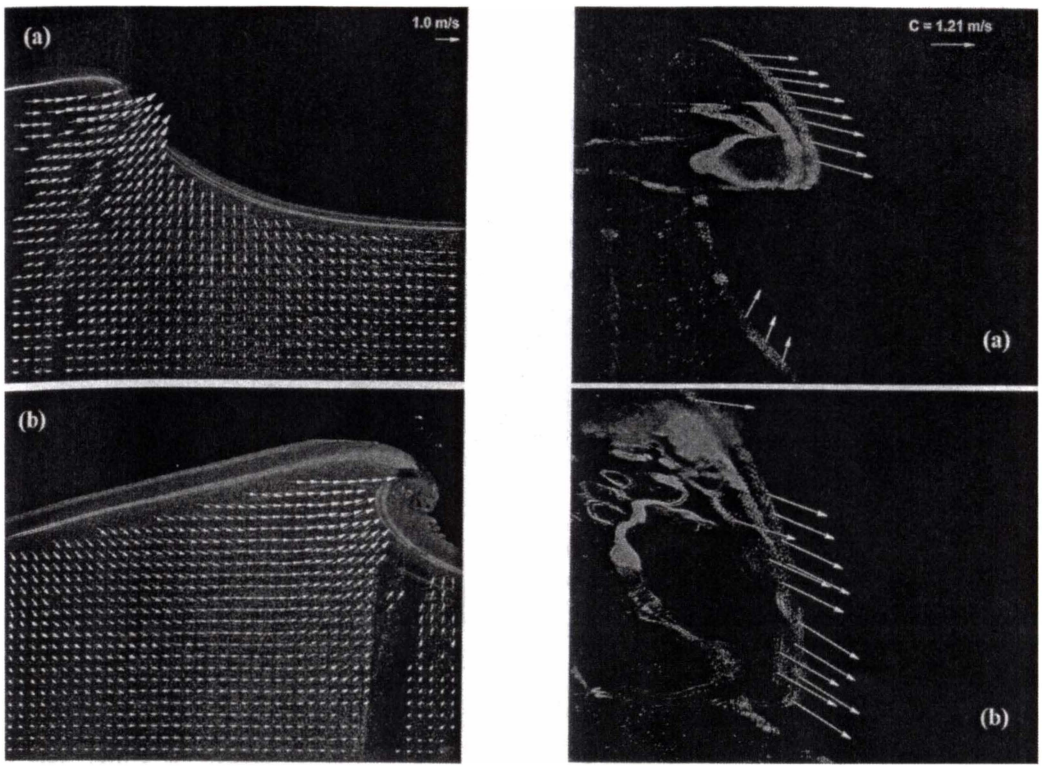

Figure 2.8: *Velocities within a shoaling wave, from Chang and Liu (1997), a result of experiments using laser particle image velocimetry.*

This parameter is distinct from the Iribarren number in that it describes how a wave *is* breaking, rather than predicting how it *will* break. This approach is good for particle-based Lagrangian methods and could be extended for gridded Eulerian methods, but is unlikely to be easily implemented for experiments or field work.

The Physics of Breaking Waves

Waves break in the surf zone because the water becomes too shallow for the stable wave form to exist (Aagard and Masselink, 1999). Alternatively Sawaragi (1995) summarises the three mechanisms for breaking in progressive waves proposed by Sir G. G. Stokes:

i) particle velocities at the wave crest exceed the phase velocity;
ii) the wave peaks and has a cusped crest with angle of 120 ° and
iii) the wave profile loses symmetry and its face becomes vertical.

Velocities measured within a shoaling wave using the Particle Image Velocimetry technique (Chang and Liu, 1997) are shown in Figure 2.8. Note that the flow in the trough is nearly horizontal and toward the wave crest, while on the face of the wave it is directed up and beach-ward. Velocities within the lip are large, predominantly horizontal and beach-ward. Velocities near the sea-floor are small by comparison. Characteristic of breaking waves is that particle velocities within the lip are close to, or exceed, the phase velocity (Sawaragi, 1995; Komar, 1998).

A common misunderstanding is that waves break when friction at the sea floor slows the base of a wave and the unsupported crest topples forward, rather waves break when they become overly steep (Komar, 1998). Experimental evidence reported in the literature (Komar, 1998) shows friction has a negligible effect on wave breaking. The effect of bed roughness and viscosity

on a flow is the generation of a boundary layer at the surface (Schlichting, 1955), which is present whether the flow is unidirectional or oscillatory. Whether the boundary layer is laminar or turbulent depends on the fluid properties and on the flow velocity. Both Isobe (1988) and Mei (1989) say that this effect is only significant near the bottom in shoaling waves. However, Isobe (1988) showed that bed roughness does affect run-up in broken waves. So, provided the boundary layer is small in comparison to the water depth, bed roughness is considered negligible when the wave is shoaling,but may be significant when the broken wave reaches shallower water.

By using linear wave theory and the condition that energy flux be conserved during the breaking process certain behaviour is predicted (Komar, 1998). These predictions have also been observed in the field and can be considered as providing an approximation to the behaviour when nonlinearity is important. A wave that enters shallow water experiences a process termed *shoaling*, in which the wave progressively steepens and increases in height, until the wave is too steep and it breaks (Komar, 1998). As the wave height increases the particle velocities in the crest increase also and when equal to the phase velocity the wave breaks (Aagard and Masselink, 1999). Shoaling waves also experience decreases in wave length and phase velocity while their periods remain constant (Komar, 1998). Since the steepness of a wave is found from the ratio of its height to its wavelength the shortening of the wavelength and the increase in height cause a direct increase in wave steepness.

Breaking Criteria

As a wave enters shallower water it experiences depth induced breaking. In the following discussion the position at which a wave breaks is defined as being the point at which its height is greatest (Shuto, 1988). For solitary waves with height H on beaches with gentle slopes (and local depth h_b at the point of breaking) a theoretical breaking criteria, γ_b, can be derived (Komar, 1998)

$$\gamma_b = \left(\frac{H}{h}\right)_{max} = \frac{H_b}{h_b} \tag{2.5}$$

where the subscript *max* means the maximum value that the ratio takes is desired. The maximum value for the ratio occurs at the break point and is indicated by the subscript b. Experiments show that γ_b is strongly dependent on beach slope and its value can vary between 0.73 and 2.8 (Komar, 1998). In general we take

$$\gamma_b = 0.78, \tag{2.6}$$

a result that is often applied to waves of all types, rather than just solitary waves (Komar, 1998).

Aagard and Masselink (1999) present an empirical relation between γ_b and the Iribarren number

$$\gamma_b = 1.1\xi_b^{1/6} \tag{2.7}$$

while Komar (1998) quotes

$$\gamma_b = 1.2\xi_b^{0.27}. \tag{2.8}$$

A number of theoretical (relying on combinations of solitary and Airy wave theories) and semi-empirical expressions have been developed relating the height of a wave at breaking, H_b and the deep water wave parameters (Komar, 1998), all of the form

$$\frac{H_b}{H_\infty} = \frac{C}{(H_\infty/\lambda_\infty)^n} \tag{2.9}$$

in which C is a constant varying from 0.303 to 0.563 and n (also a constant) varies from $\frac{1}{5}$ to $\frac{1}{3}$. The difference in the choice of constants determines how well the expression agrees with different sets of data.

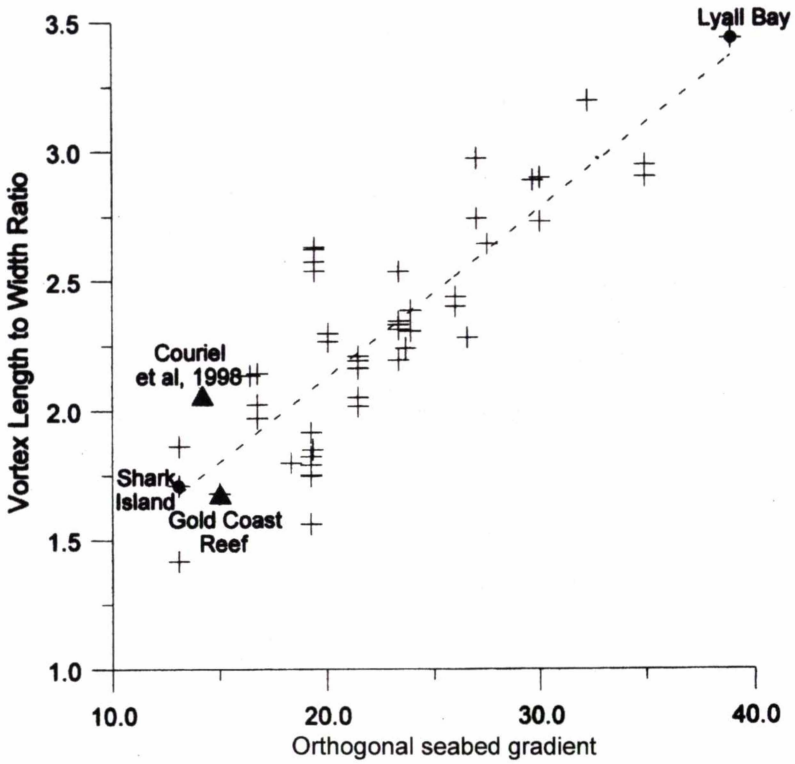

Figure 2.9: *Barrel length to width ratio ($\frac{l}{w}$) versus orthogonal seabed gradient (Mead and Black, 2001). The dashed line is Equation 2.10. The gradient is given as the denominator from the ratio, so a gradient of 10.0 is a 1:10 slope.*

So given H_∞ and λ_∞ for a wave in the open ocean the height it will have at the break point can be estimated (using equation 2.9), and then after finding γ_b (using equation 2.7 or 2.8 with the definition for the Iribarren number), the depth at which the wave will break can be estimated (using equation 2.5). Galvin (1968) or Battjes (1974) can then be referred to in order to estimate whether the wave will plunge, spill etc.

2.1.4 Breaking Waves and Bathymetry

Most of the literature on the effect of bathymetry on breaking waves is concentrated on plane beaches and the effect of slope on which descriptor (plunging, spilling etc.) is best suited to waves breaking on the beach.

By fitting equation 2.1 to digitised positions around barrels in photographs of breaking waves Mead and Black (2001) were able to describe vortex (barrel) dimensions as in terms of the vortex length to width ratio and the vortex angle, l/w (see Figure 2.9). Mead and Black (2001) then correlated the vortex length to width ratio against the *orthogonal seabed gradient*, and found a line of best fit

$$\frac{l}{w} = 0.065 \times \text{orthogonal seabed gradient} + 0.821. \tag{2.10}$$

Clearly a seabed has a different gradient in different directions, and the orthogonal seabed gradient is the gradient in the direction along the wave orthogonal. A wave orthogonal is a curve to which the local crest of the wave is always perpendicular. Vortex angles can vary considerably for a single surf break and appear unrelated to seabed gradients (orthogonal or otherwise) and vortex length to width ratios.

Notably, Mead and Black (2001) could only *estimate* the positions of breakers in their photographs with respect to the bathymetries they had collected. So there are significant uncertainties associated with the orthogonal seabed gradients they used. There is scope for improvement of Equation 2.10 by experiments in which the breaker positions are identified more accurately.

2.1.5 Generating Waves Using Pistons

In laboratory wave flumes, waves are created by the use of wave makers, which include (among others) the piston and flap types (Dean and Dalrymple, 1991). The piston type comprises a plane, vertical surface that is moved in the direction normal to the plane. A flap type is also a plane surface, but is hinged at the bottom edge. Dean and Dalrymple (1991) note that the piston type is better for generating waves in shallow water, while in deep water the flap type is preferable, in each case because the piston or flap (respectively) motions match well the water particle trajectories.

Waves can be created in numerical wave tanks by the same method as in the laboratory. The emphasis in this work is on waves in the coastal zone so piston type wave makers are the more important. Other wave making methods were not considered here. Having decided upon a piston-type wave maker its motion must be controlled if it is to create waves with the desired forms. Poor choice of piston trajectories can mean non-permanent waves are generated.

Goring (1978) developed a method by which permanent wave forms (such as Airy, Stokes, solitary and cnoidal waves, but with an emphasis on the second two) can be generated from piston-type wave makers. Goring (1978) developed the technique to create laboratory waves with a piston-type wavemaker that was controlled using a function generator that could be programmed using paper tape. The function generator had a very limited (by today's standards) memory and so only first order wave theories were used. Notably, Dalrymple and Knio (2001) used Goring's method to generate waves in their SPH model.

In Goring's method the piston is moved in order that its motion matches the depth averaged wave velocities (Goring, 1978). However it should be noted velocities are not being matched at the same point (irrespective of time), but rather at the same water particle, irrespective of time. The method requires expressions for the functional form of, and the dispersion relation for, the wave.

A significant alternative approach was provided recently by Katell and Eric (2002) who, rather than using the Boussinesq solution for the solitary wave, used the Rayleigh solution. In fact the two solutions are the same but for the values of two constants. Katell and Eric (2002) noted that solitary waves generated by Goring's method dispersed more quickly and concluded that the rate of dispersion could not be explained by friction and viscosity alone, rather it indicated the waves were not permanent and were experiencing dispersion. Experimental evidence corroborated the conclusions of Katel and Eric. They did not, however, extend the method to investigate other types of permanent waves.

2.2 Numerical Models of Breaking Waves

The computational modelling of breaking waves is difficult because of the range of spatial scales at which the solution must exist. The wave will typically be less than 10 m high and yet have a wavelength of the order of 100 m. Water depth near breaking might be between 1 m and 5 m while the horizontal distance over which the wave breaking occurs could be up to 1 km. The smallest features we might like to resolve could be the thickness of the plunging lip, perhaps 0.1 m in size. The range in scales required in such a simulation could cover 4 orders of magnitude and present a formidable problem since achieving a solution will mean high resolutions and consequently high run-times (Ferziger and Perić (1996, p. 255) note this point in a discussion on the modelling of

turbulence). Researchers have chosen a variety of spatial domains for use in their models, using different water depths and bathymetries.

There are many existing computational models of breaking waves and only those that can simulate the time-dependent breaking of waves in the vertical plane are examined here. These contain three significant families: volume of fluid models, potential theory models and Lagrangian particle models. In this section the three major families are introduced. The performances of the models are discussed in terms of their merits and demerits, and the constraints upon simulations that they can perform. A few non-conformists are also discussed briefly, namely a Lagrangian finite element model and a pair of marker and cell (MAC) models.

2.2.1 The Volume of Fluid Method

The Volume of fluid (VOF) method is technique that can be used to locate and track free surfaces in Eulerian (gridded) models, determine how the free surface will evolve in time and enable the application of boundary conditions (Hirt and Nichols, 1981). Without VOF Eulerian methods can encounter problems at free surfaces, including unrealistic diffusion (Harlow and Meixner, 1961; Hirt *et al.*, 1970).

VOF is primarily controlled by the introduction of a volume fraction (or colour function), a field of values defined at each grid cell (Chen *et al.*, 1999). The system being simulated might contain empty cells (along with full and partly full cells), or it might contain 2 different fluids so every cell contains either one or the other fluid, or a mixture of both (Chen *et al.*, 1999). Once the volume fractions have been computed the free surface within each cell can be expressed as a line segment. There are a number of ways to generate the line segments, some of which are discussed in Ashgriz and Poo (1991).

But the VOF method is really just a way to account for the free surface, and any model employing the method requires the selection of a set of governing equations, equations derived from the fundamental principles of physics. There are two significant options that are discussed here, the Reynolds equations and the Navier-Stokes equations.

VOF and the Reynolds Equations

The model created by Lin and Liu (1998a,b) and the commercial model used by Bradford (2000) use the Reynolds equations. In these equation variables are expressed as, taking the general property ψ as an example, $\psi = \overline{\psi} + \psi'$, in which $\overline{\psi}$ is the slowly varying (mean flow) contribution, and ψ' is the rapidly varying (turbulent) contribution. This approach introduces new variables for all of the ψ's, and more equations are required if solutions are to be found.

One possibility (there are many) is that the new equations are provided by the k-ε model, an approach utilised by Lin and Liu (1998a,b) in their breaking-wave model. The standard form of the k-ε model is the creation of equations for the turbulent kinetic energy, k, and the turbulent kinetic energy dissipation rate, ε. The merit in this approach is that flows are *turbulence averaged* and results can be easily compared to experimental data since most experiments are performed in Eulerian (fixed in space) coordinates.

VOF and the Navier-Stokes Equations

An alternative to the Reynolds equations for use with VOF are the Navier-Stokes equations, as used by Chen *et al.* (1999) in their model of a breaking wave in deep water, and by Guignard *et al.* (2001) in their model for waves breaking on beaches. If this approach is used and eddies are simulated down to the *Kolmogorov* scale, where turbulent kinetic energy are dissipated into heat (Oran and Boris, 1987), then it is an example of *direct numerical simulation* (Ferziger and Perić, 1996). Neither Chen *et al.* (1999) nor Guignard *et al.* (2001) go to quite that detail, in their

models the smallest eddy simulated is determined by the cell size, and any turbulence occurring at a smaller scale than that is termed *sub-scale*. The dissipation to heat at the sub-scale level is left to the artificial viscosity terms, meaning that their models provide *large-eddy simulations* (Oran and Boris, 1987). If this approach is used the assumption has been made that (Oran and Boris, 1987) sub-scale turbulence does not feed back to large scale eddies.

Chen *et al.* (1999) state that the inclusion of surface tension in VOF models is important because it ensures some numerical instabilities are avoided.

The Origins of VOF

One notable early numerical method is the *marker and cell* method (see Section 2.2.4) (Harlow and Welch, 1965) which preceded and evolved into, VOF. The marker and cell method is a composite Euler-Lagrange method that uses massless Lagrangian markers within a gridded Eulerian model to track the free surface, the motion of the markers being determined from the variables defined on the grid. The free surface can be found simply by identifying cells with markers that lie next to cells without markers. It was found that large numbers of markers were required to track the free surface which resulted in long run-times (Hirt and Nichols, 1981) and greater storage (not the problem today that it was in 1981). VOF requires lower run-times and storage than marker and cell and has become somewhat of a standard for the Eulerian modelling of free surface flows. Examples of the application of the marker and cell method to breaking waves are discussed in Section 2.2.4.

Some Important Features of VOF Simulations

For a two dimensional domain with $N \times N$ points a VOF model needs to perform computations at all of the N^2 points, and consequently much longer run-times are required than for potential theory models, which use only the $4N$ points around the boundary to achieve the same resolution.

VOF easily handles fragmentation and coalescence of the fluid without requiring special treatment. This is demonstrated clearly in Figure 2.10, which is taken from Chen *et al.* (1999), and shows splash-up occurring after the breaking of an initially unsteady deep water wave. VOF does not require a simply connected domain or the condition of irrotationality, as potential theory methods do (Guignard *et al.*, 2001), and so simulations are not limited to the time at which the falling lip hits the water.

Notable of VOF models is that they experience numerical diffusion, suggested by Guignard *et al.* (2001) as the reason for the loss of energy observed in some simulations.

The VOF models of Lin and Liu (1998a,b) and Bradford (2000)

Lin and Liu (1998a) and Bradford (2000) used VOF models based on the Reynolds equations to simulate waves breaking on beaches using 900×77 and 800×80 cells respectively. However neither Lin and Liu (1998a) nor Bradford (2000) present details of the free surface profile of their breaking waves, instead concentrating on the nature of turbulence within the surf-zone.

Lin and Liu (1998b) use the same model as Lin and Liu (1998a) in their paper on turbulence within the surf-zone (they used a 600×60 grid for most of the work) but included detail of the plunging lip of a breaking wave from a simulation with 900×77 cells, a simulation that is notable for peculiar shape of the lip (Figure 2.11), a feature that they do not explain.

The VOF model of Chen *et al.* (1999)

The resolutions achieved by Chen *et al.* (1999) in their model of breaking waves in deep water (up to 512×512 cells, about half of which contain air) are quite impressive but the authors do

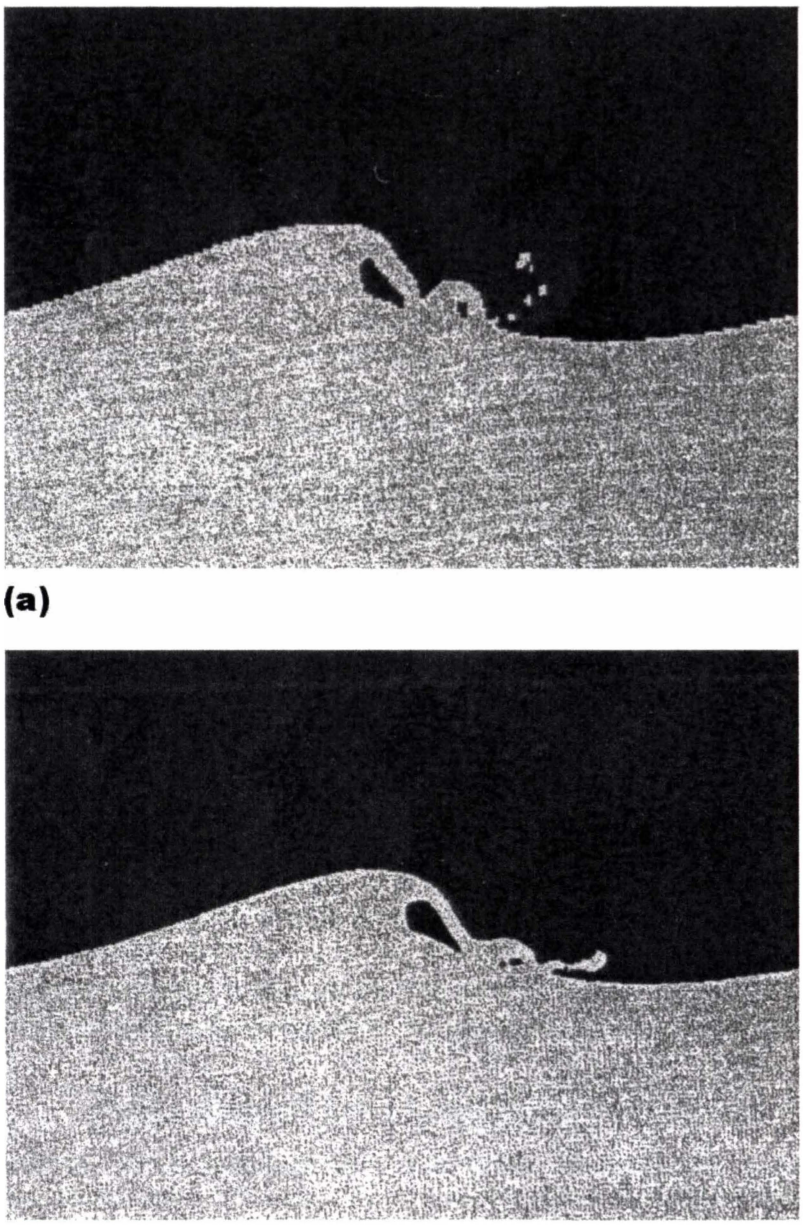

Figure 2.10: *A deep water breaking wave demonstrating splash-up with fluid fragmentation and coalescence from the VOF model of Chen* et al. *(1999). The two plots are from the same simulation performed at different resolutions,* (a) *at 256×256, and* (b) *at 512×512. The slight differences in the two are due to errors in simulating very small features.*

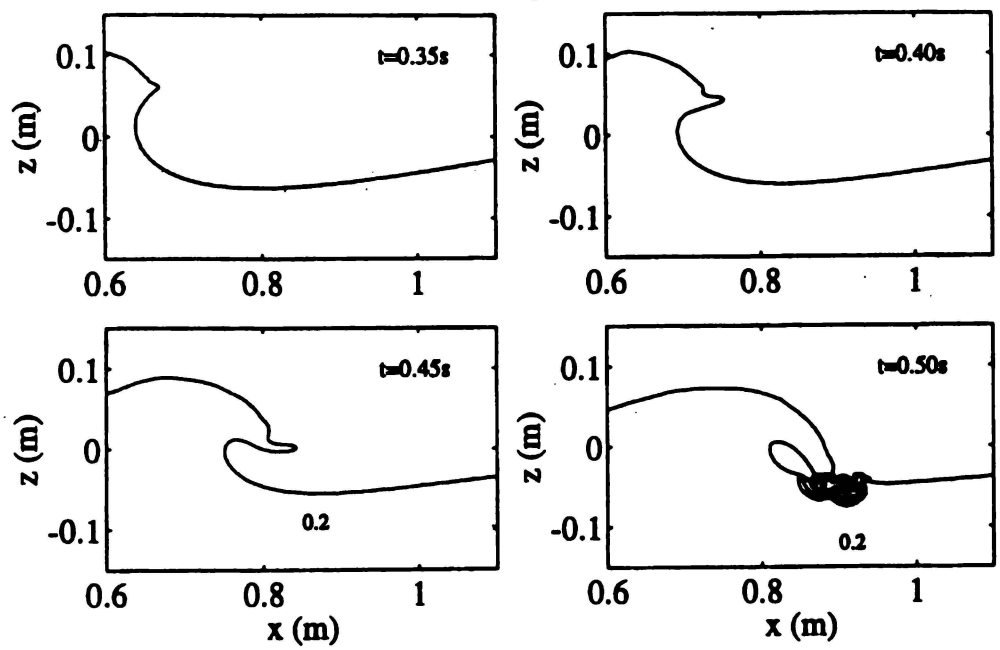

Figure 2.11: *Detail of the lip (from Lin and Liu, 1998b) from a simulation run using a VOF model. The contour lines in the bottom right plot are turbulence intensities.*

not state the run-times required nor the computer upon which simulations were performed. The spatial domain is square with a constant depth and the vertical boundaries are periodic. This means the wave used by Chen *et al.* (1999) is large compared to the horizontal size of the domain. Results from Chen *et al.* (1999) are shown in Figure 2.12.

Chen *et al.* (1999) start their simulations with an initially unstable wave and allow the simulation to progress. As a consequence their wave breaks soon after initiation and at 8 s the wave energy has almost completely dispersed. They use a time-step of 10^{-4} s so require 80 000 time-steps. If they were to simulate a wave approaching a beach and shoaling, the propagation of the turbulent bore and run-up over the swash zone, then more time-steps would be required, and hence increasing time to run the simulation.

Most breaking wave models are constrained with regard to the bathymetries that can be used, often being limited to constant slopes or flat bottoms. The focus of this thesis is the breaking of shallow water waves on beaches with real bathymetries. The use of real bathymetries means that periodic boundaries cannot be used. On typical beaches the horizontal distance over which waves shoal and break and the distance the turbulent bore covers are large in comparison to depth and typical wave sizes. While the numbers of cells used by Chen *et al.* (1999) could be used for a breaker on a beach, the number of cells which a wave comprises will be smaller, and so the resolution of the wave would be lower.

Figure 2.13 shows results from Chen *et al.* (1999) with detail of the wave lip forming a clear lobe, explained as the effect of surface tension which was included in the governing equations. In the same Figure Chen *et al.* (1999) make a comparison with BEM results generated both with and without surface tension. Without surface tension the lip of the wave remains sharp, while with surface tension it is rounded. Figure 2.14 shows the acceleration within the lip for the VOF simulation and the upward acceleration at the tip due to the presence of air within the barrel (Chen *et al.*, 1999).

Flow Science Inc. (2003) claim some VOF models employ what they term *pseudo-VOF*, an

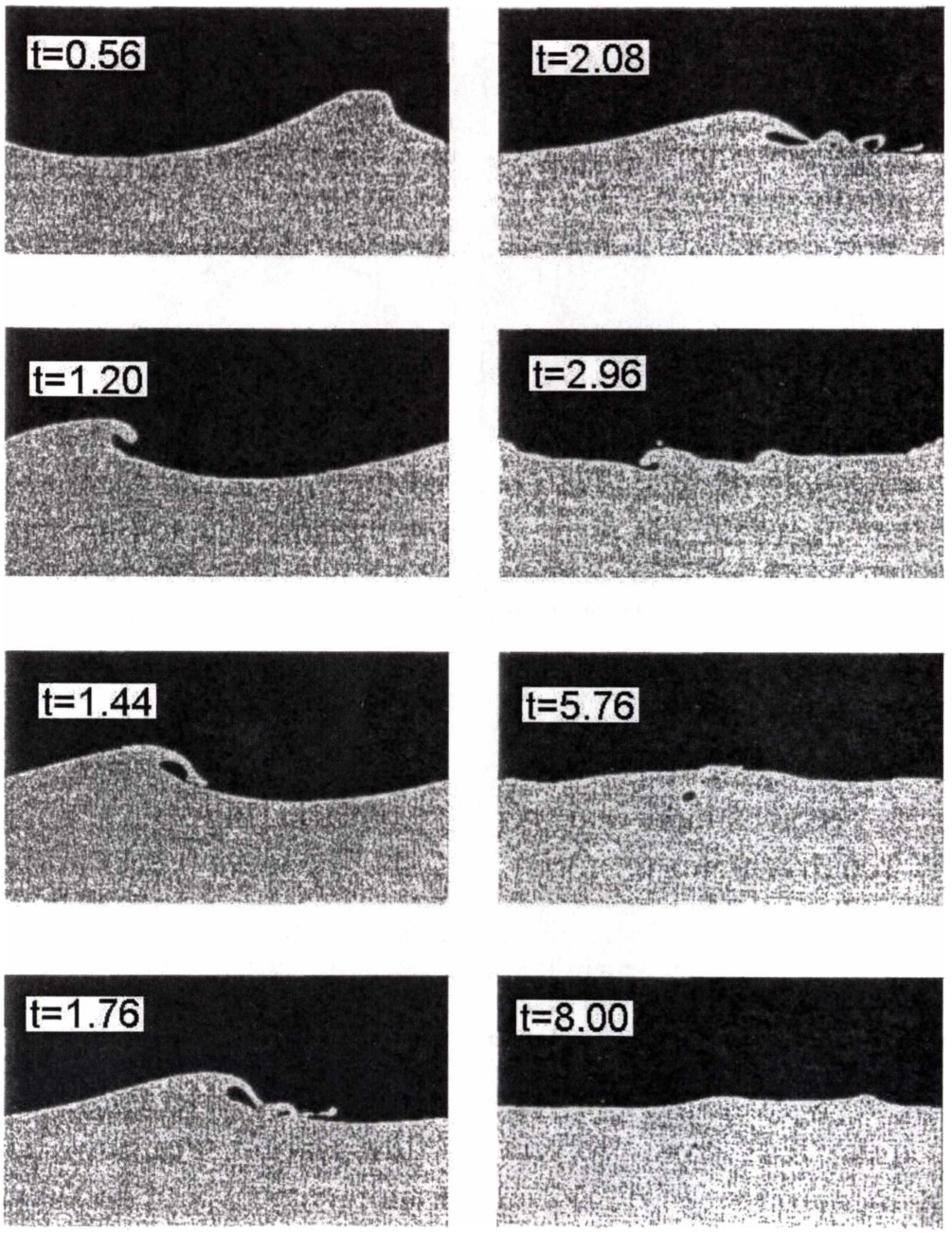

Figure 2.12: *Snapshots of a VOF simulation (Chen* et al., *1999) of a breaking deep water wave. This simulation used a resolution of 512×512, periodic vertical boundaries and is one wavelength wide in the horizontal.*

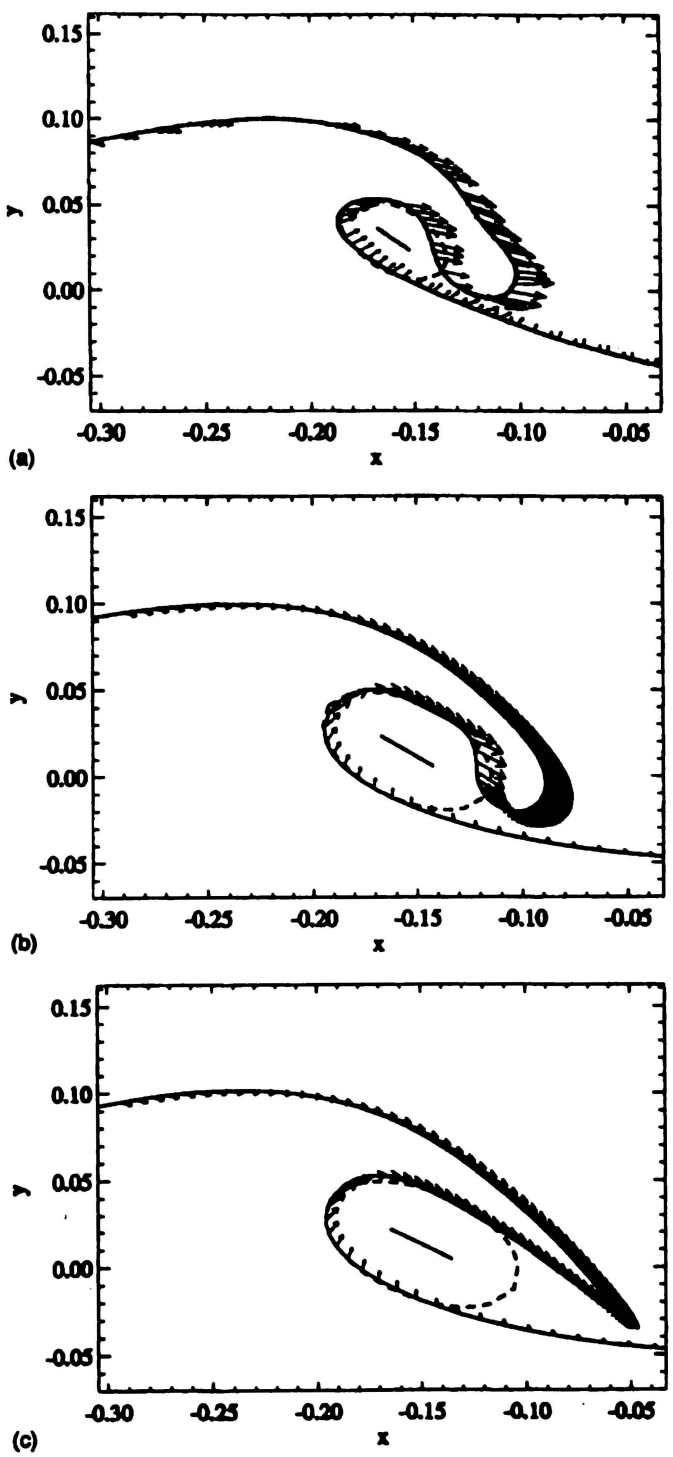

Figure 2.13: *A comparison of a VOF simulation* (a), *with BEM results that include the effect of surface tension* (b), *and that do not* (c), *(Chen* et al., *1999). The lobe generated by the presence of surface tension is clear.*

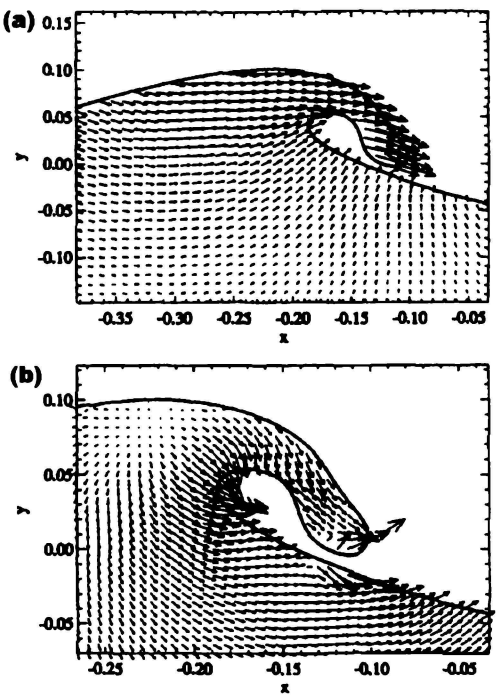

Figure 2.14: *Detail of the plunging lip in a simulation from Chen* et al. *(1999), in which the vectors are velocities in* (a), *while in* (b) *they are accelerations. Notice the effect of air on the accelerations at and under the lip in* (b).

approach involving incorrect treatment of boundary conditions at the free surface. The consequences of the use of *pseudo-VOF* are demonstrated in Figure 2.15 by the unphysical generation of a growth at the tip of the jet, a growth that is independent of the choice for the density of air. This effect does not explain the lobe in the lip of the falling wave from Figure 2.13 since the lobe also appears in the BEM simulation that employs surface tension.

The VOF model of Guignard et al.

The model used by Guignard *et al.* (1999, 2001) is similar to that used by Chen *et al.* (1999) but it is designed to simulate waves breaking on beaches. Figure 2.16 shows output from Guignard *et al.* (1999), and although they used a model domain with 80×2500 cells (only slightly smaller than the maximum 512×512 used by Chen *et al.* (1999)) the resolution on the face of the wave is markedly lower. This is a consequence of the size of the wave in comparison to the size of the domain.

Guignard *et al.* (2001) note the long times required to run their VOF model, and suggest the coupled BEM/VOF model described in Guignard *et al.* (1999) as a solution. They state that such an approach would "*...allow us to both reduce cell size and the numerical diffusion* ..." (Guignard *et al.*, 2001, p. 73), meaning that either higher resolutions or shorter run-times could be achieved by this approach. They proceeded to develop such a coupled model and this is further discussed in Section 2.2.2.

A comparison of results with experimental data (Guignard *et al.*, 2001) is shown in Figure 2.17. This figure shows similarities but the simulation has a smaller barrel with a different shape later in the simulation. The lip of the computational wave precedes that of the experimental wave by a significant amount, a feature that the authors do not explain.

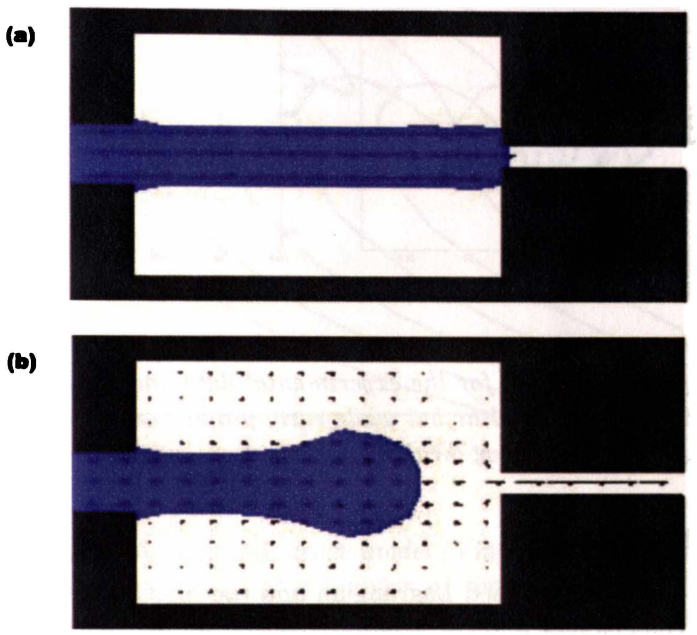

Figure 2.15: *A jet of water entering an air-filled cavity using* (a) *VOF and* (b) pseudo VOF, *from Flow Science Inc. (2003), who assert that the lobe is a numerical effect due to the use of* pseudo VOF.

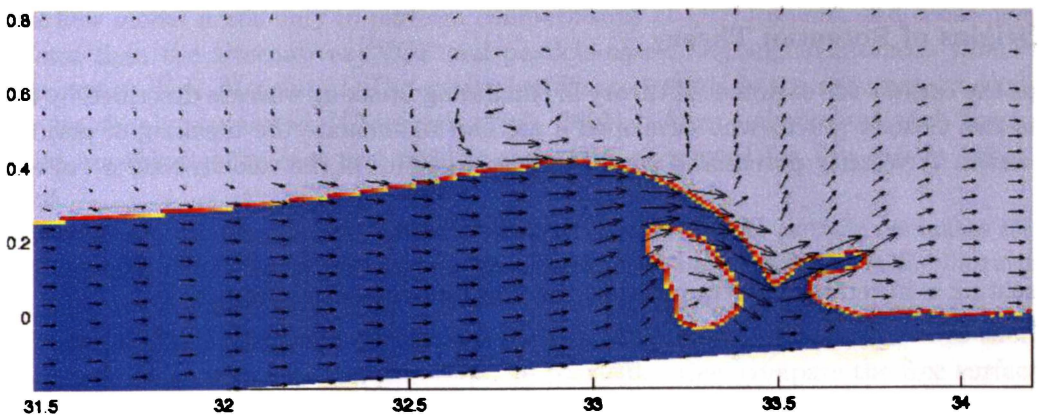

Figure 2.16: *Output from the VOF model described in Guignard* et al. *(1999). The water is dark blue, air is light blue and cells with a mixture of air and water are yellow-red. Note the eddy generated in the air behind the breaking wave.*

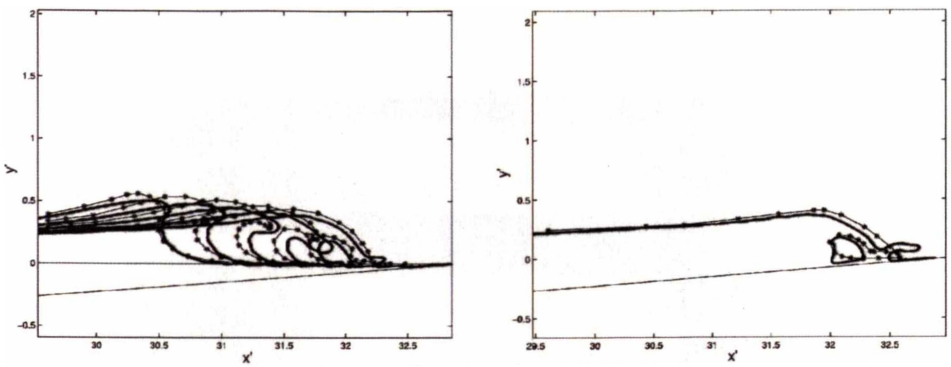

Figure 2.17: *A comparison of output from a VOF model (Guignard* et al., *2001) and experimental data. The lines with asterisks are for the experimental data and the heavy, full line for the VOF simulation. The similarities are clear but while early profiles match well, later ones do not. The VOF barrel is smaller and the wave breaking evolves more quickly.*

Summary of VOF models

Like models using the particle-based Lagrangian and the MAC methods, VOF models can simulate the highly turbulent flow and the shoaling transformation that occur in the wave breaking process. The ability to account for the presence of air and surface tension at relatively high resolutions are noteworthy merits of the VOF method. The effects of numerical diffusion have been observed in some simulations.

2.2.2 Potential Theory Methods

A number of researchers have created models of breaking waves based on potential theory. This family of methods are typically fast and accurate, but cannot be used in certain situations. In this section the basic theory of potential flow models is introduced and the capabilities of current models are described and critiqued.

The Origins of Potential Theory

Perhaps the earliest use of potential theory in simulating breaking waves is described by Longuet-Higgins and Cokelet (1976) who developed a method to simulate the breaking of periodic deep water waves. A velocity potential ϕ can be defined in terms of the velocity vector ($\boldsymbol{v}$) as

$$\boldsymbol{v} \equiv \nabla\phi. \tag{2.11}$$

Representing $\boldsymbol{v}$ in terms of ϕ requires that the flow be irrotational, that is

$$\nabla \times \boldsymbol{v} = \mathbf{0} \tag{2.12}$$

(where $\mathbf{0}$ is the zero-vector) and also that the domain be simply connected (Guignard *et al.*, 2001). Using this result, the principle of conservation of mass can be written as Laplace's equation for the potential

$$\nabla^2\phi = \nabla.(\nabla\phi) = 0. \tag{2.13}$$

From this point, solutions to 2.13 subject to boundary conditions can be found by the use of Green's functions (just one of a number of possible approaches) to generate an integral (called the boundary integral equation) over the boundary of the model domain (Grilli *et al.*, 2001). This

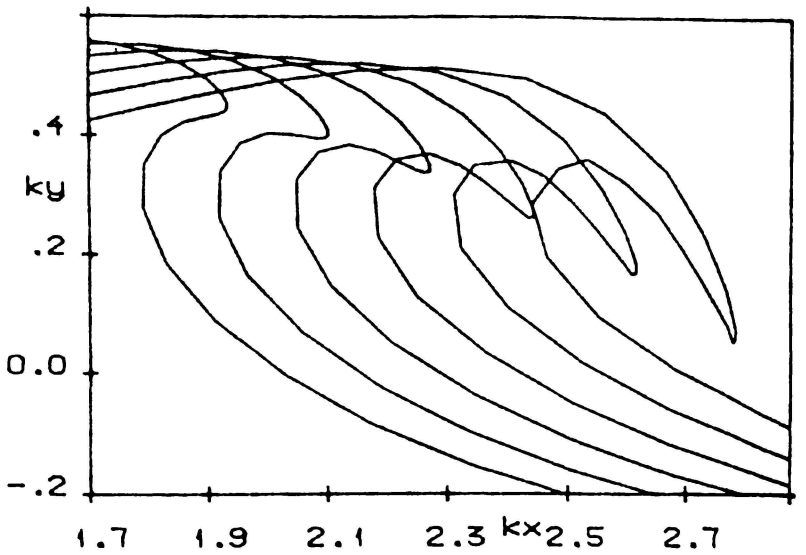

Figure 2.18: *Results from an early potential theory model (Vinje and Brevig, 1981) showing detail of the plunging lip of a deep water wave at a number of times. The breaker was generated by specifying an initially unstable sinusoidal free surface.*

approach is called the *Boundary Element Method* (BEM). Alternatively, rather than writing the velocity potential in the form given in Equation 2.11, a complex potential can be defined (limiting the method to two dimensions) and Cauchy's theory can be used to find the solution (Vinje and Brevig, 1981). Results from Vinje and Brevig (1981) are shown in Figure 2.18.

The Merits and Demerits of Potential Theory Models

The feature that makes potential theory models so useful is that computations are required only at the boundaries of spatial domains. For a two dimensional domain with $N \times N$ points a potential flow model needs only to perform computations at $O(N)$ points, and consequently are *much* faster than the alternatives (VOF and particle-based Lagrangian methods which require computations at $N \times N$ points). The most significant demerit is the requirement that model domains be simply connected (Guignard *et al.*, 2001). Also, only irrotational flows can be considered, meaning turbulence and friction cannot be accounted for, but this is not a serious constraint.

Comparisons of numerical simulations and experimental results provide estimates of the accuracy of models. For BEM models many comparisons have been discussed in the literature, and some of these are outlined here.

Grilli *et al.* (1997a) compared simulations (from a BEM model) of solitary waves shoaling on plane-sloped beaches with experimental data. In particular they compare the free surfaces from the two sources, a comparison from which they concluded that their BEM model was extremely accurate. A plot of this comparison is shown in Figure 2.19.

Svensden and Grilli (1990) compared BEM simulations of the run-up and reflection of solitary waves on steep slopes against experimental and analytic results. Again the comparison was favourable and the authors concluded that there was agreement to within experimental uncertainty, except where slopes were lower and friction was suggested as having a significant role in the experimental results. Grilli and Horrillo (1997) simulate the generation and propagation of fully nonlinear periodic waves and compare their results to experiments, finding that the two

agree very well.

Thus BEM models are capable of accurately simulating shoaling solitary waves, and also of simulating the generation and propagation of nonlinear periodic waves. This, with their relatively high computation speed mean that when we wish to simulate breaking waves, potential theory models are easily the preferred option, but only if our interest lies in the wave before the lip has hit the face, and it is only if we are interested in the period after this event that VOF or particle-based Lagrangian methods are the better choice.

The Capabilities of Potential Theory Models

Longuet-Higgins and Cokelet (1976) examined waves in infinitely deep water and used periodic vertical boundaries. Vinje and Brevig (1981) did not constrain themselves to deep water (however they do require constant depths) and also used periodic vertical boundaries. Longuet-Higgins and Cokelet (1976) examined breakers generated by the application of a pressure distribution at the free surface, while Vinje and Brevig (1981) started with initially unstable sinusoidal waves.

Grilli *et al.* (2001) note that more recently the method was extended to account for arbitrary waves in finite depths over irregular bathymetries. Grilli *et al.* (2001) discuss the use of adaptive gridding and time-stepping, used to ensure that the model provides the required resolution at all points within the model domain and that their model remains computationally stable, that errors are controlled.

Grilli *et al.* (1997b) note that in their potential theory model "arbitrary waves" can be generated. Grilli and Horrillo (1997) discuss the details of making waves with a *flap* wavemaker in their *numerical wave tank*, while Grilli and Svendsen (1990) use a piston wave maker. Since waves can also be specified in the initial state of a system (Vinje and Brevig, 1981), a number of methods are available for creating any of a variety of wave forms in potential theory models.

Grilli *et al.* (1997b) note that their potential theory model can be used with complex bathymetries. Grilli and Svendsen (1990) used (probably the same model) to simulate the transformation of waves over a beach with a submerged breakwater. Grilli *et al.* (2001) contains details of a potential theory model in 3 dimensions, a model that was used to simulate waves breaking over a submerged ridge, an example of which is shown in Figure 2.20. While the literature does not contain descriptions of simulations over a wide range of complex bathymetries, it is clear that potential theory models have this ability.

A Coupled BEM/VOF Model

To avoid the limitations of requiring a simply connected domain Guignard *et al.* (2001) created a coupled potential theory - VOF model. They employed a BEM model to simulate waves up to a certain point where the results were used as input to a VOF model. After the point of coupling wave energy in the BEM model was dissipated by the use of an absorbing beach. The means by which the models are coupled is shown in Figure 2.21. When the wave reaches the position shown by profile a in the BEM model (plot (c) Figure 2.21), boundary conditions for the VOF model are generated and computation begins. The merit in coupling models is the ability to simulate waves after breaking while taking advantage of the lower cost and higher accuracy of the BEM method in the deeper part of the domain.

Summary of Potential Theory Models

Models based on potential theory are fast and accurate and for modelling breaking waves they are the pre-eminent tool. Waves of arbitrary form can be input either by wavemakers or by the specification of the initial condition, and complex bathymetries can be used. These methods are limited to simulations in which the flow is irrotational and the computational domain remains

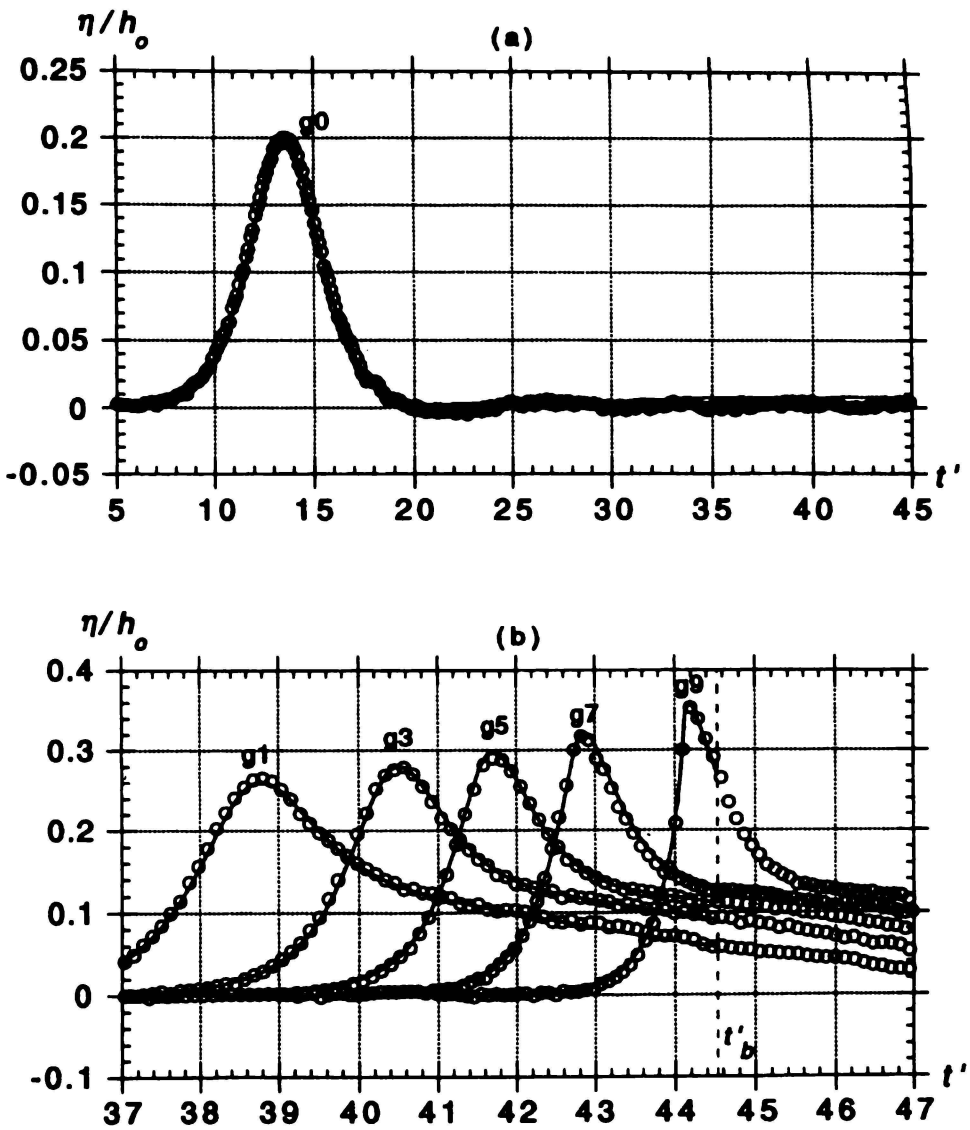

Figure 2.19: *A comparison of a BEM simulation with experiment (Grilli* et al., *1997a). Plot* (a) *shows the solitary wave they generated (using a method similar to Goring's method), and* (b) *shows various stages of shoaling.*

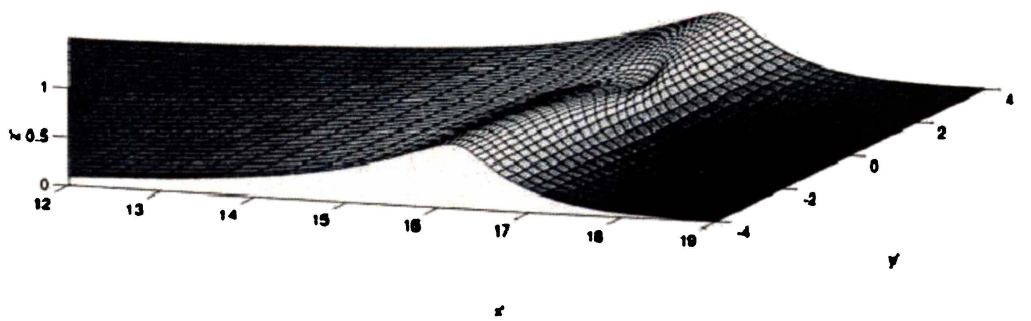

Figure 2.20: *Results from the three dimensional BEM model of Grilli* et al. *(2001), showing a solitary wave breaking over an submerged ridge. The coordinates are dimensionless, scaled by a constant depth.*

simply connected, so they can't be used after the lip of a breaking wave has hit the wave face. The coupling of potential theory models with VOF models is a significant step forward.

2.2.3 Particle-Based Lagrangian Methods

The first particle-based Lagrangian method was Smoothed Particle Hydrodynamics (SPH) (Gingold and Monaghan, 1977; Lucy, 1977) and was developed for simulating astrophysical fluids. The first application (Monaghan, 1994) of SPH to simulating free-surface fluid-flows included simulations of breaking waves, a bore and a bursting dam. The basis for the method is the idea of Lagrangian control masses, for which field variables are determined by the use of *kernel estimates*, which, along with other details of how SPH works, are presented in Chapter 3.

SPH is considered to be one of the earliest members of the family of meshless methods, named because no mesh is required, either Lagrangian or Eulerian. Another early member was the generalised finite difference method (GFDM) of Liszka and Orkisz (1977), a method that was not applied to fluids (but could have been if it were used with the idea of convecting Lagrangian particles). Since then a number of other meshless methods have been developed, among which are the element free Galerkin, diffuse element, h-p clouds and the *partition of unity* methods which are reviewed by Duarte and Oden (1995), Belytschko *et al.* (1996) and Liszka *et al.* (1996). Another notable example is the moving particle semi-implicit method (MPS) (Koshizuka *et al.*, 1995), a variation on SPH.

Some of the meshless methods have been applied to simulating incompressible free-surface flows, but as far as the author is aware, only MPS and SPH have been applied to breaking waves. These two methods are discussed here, noting the merits and demerits but without examining the methods in detail.

Numerical Errors in MPS and SPH

Numerical models always experience three kinds of errors: *modelling errors*, *discretisation errors* and *convergence errors* (Ferziger and Perić, 1996). Modelling errors are the difference between the actual flow and the exact solution which is more often than not an unachievable ideal. Convergence errors are the difference between the exact solution to the discretised governing equations, and the solution found by iterative methods. Discretisation errors are generated when the terms of the exact conservation equations are discretised.

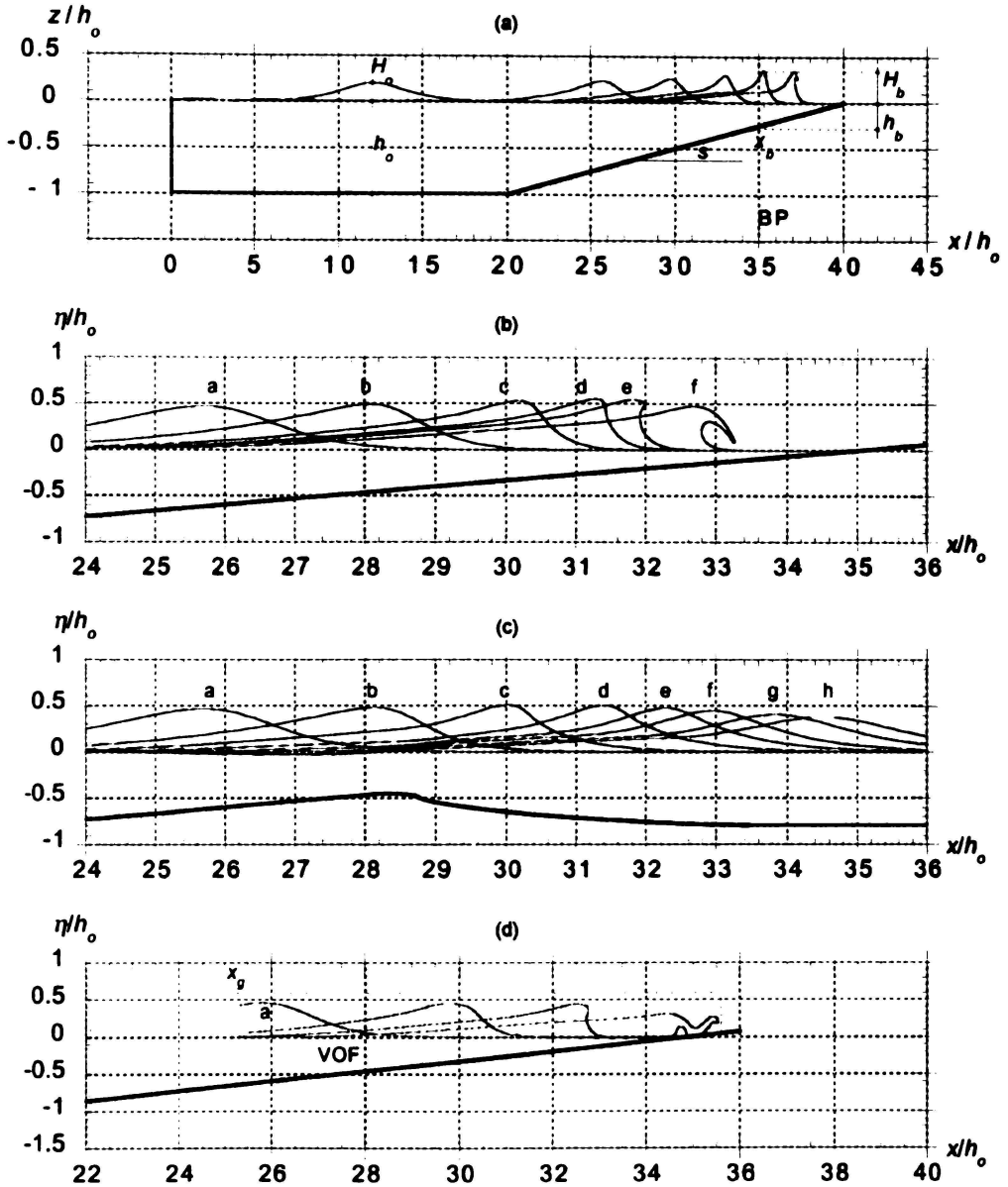

Figure 2.21: *Output from the coupled BEM/VOF model of Guignard* et al. *(2001) showing how the models were coupled.* (a) *shows the entire model domain;* (b) *shows how the solitary wave would shoal in a BEM model;* (c) *shows the domain modified with an absorbing beach used in the BEM part of the model;* (d) *shows the domain used by the VOF part of the model and the breaking wave.*

Numerical diffusion is an example of a discretisation error that occurs when convection terms (those with the form $\boldsymbol{v}.(\nabla\psi)$, where ψ would be velocity ($\boldsymbol{v}$) in the conservation of momentum equation or density (ρ) in the conservation of mass equation) in the governing equations are discretised (Gingold and Monaghan, 1982; Oran and Boris, 1987). In reality numerical diffusion is just one of a number of discretisation errors and may or may not be the most significant. In MPS and SPH the governing equations are written in terms of Lagrangian derivatives and so do not contain convective terms, hence these methods are free of numerical diffusion.

Governing Equations

Both MPS and SPH are based on finding solutions to equations derived from the principles of conservation of mass and momentum. Similar to the VOF models (Chen *et al.*, 1999; Guignard *et al.*, 1999, 2001), MPS and SPH are examples of direct numerical simulation, in which the smallest eddy that can be resolved is determined by the size of the particles used.

MPS, SPH and Pressure

SPH and MPS are similar in how the governing equations are discretised, but with a notable difference in how pressure is calculated. In SPH the fluid is compressible and pressure is determined by the use of an equation of state (Monaghan, 1994). In MPS a Poisson equation for pressure is used (Koshizuka and Oka, 1996). The use of a Poisson equation for pressure in MPS means the governing equations are implicit, so solutions must be found by iterative means (Koshizuka *et al.*, 1995). Consequently MPS models require more computations per time-step than those using SPH.

This is a significant difference since it means SPH simulates the propagation of sound through the fluid (Monaghan, 1994) and stability of the method (from the Courant condition) requires much shorter time-steps (than required by MPS). Monaghan (1994) steps around this requirement by artificially reducing the speed of sound, greatly increasing the allowable step-size, but even so steps are required to be shorter than those in MPS. The use of a lower speed of sound represents a modelling error that has the effect of generating larger variations in density. But there are relatively few situations in coastal oceanography where sound is an important feature.

So the trade off is clear: more short, quick and less accurate steps (SPH) or fewer long, slow and more accurate steps (MPS). A danger in the SPH choice is that any errors dependent on the number of time-steps will accumulate at a greater rate.

Variations from the MPS approach can be found in the literature including the work of Cummins and Rudman (1999). Their method is termed PSPH, from a *Projection method for SPH*, and also solves the Poisson equation for pressure, but using a different method. They concentrate on comparing PSPH against traditional SPH, or as they call it, *Weakly Compressible SPH*, WCSPH. They conclude PSPH is more accurate for simulating incompressible flow because it avoids the short step-size of WCSPH, but claim that more research is required in order that the method be able to account for a free surface.

MPS, SPH and Unbounded Phenomena

The *most* notable merit of the family of meshless methods is their suitability for simulating unbounded systems, or nearly unbounded systems. In such simulations the process of generating Eulerian grids (many cells in which will be empty) becomes extremely expensive (Gingold and Monaghan, 1977). Also important is ability of meshless methods to simulate systems for which much of the domain is empty (Hernquist and Katz, 1989), while maintaining high resolutions in the areas of interest. The evolution of a galaxy is an example of a simulation that might require an unbounded, predominantly empty domain.

Breaking waves have the characteristic that the domain is large in comparison to the desired resolution (not quite galactic scales though) but don't have a lot of empty space. Unbounded phenomena are rare in coastal oceanography, so these properties are of little benefit here.

Boundaries in SPH

There are several methods used to create boundaries in SPH models, all of which involve placing stationary particles with in the domain. These boundary points apply a repulsive force to free fluid particles and so contain the fluid. A *ghost point* boundary employs the Lennard-Jones potential (but any potential could be used) to repel free fluid particles (Monaghan, 1994). However when this method is employed undesirable *corrugated* boundaries are produced that disrupt flows. Monaghan and Kos (1999) modified the method to ensure that the force is always normal to the boundary, and here these are called *normal/tangential* boundaries.

Koshizuka and Oka (1996) and Dalrymple and Knio (2001) use a different approach, placing several rows of *fluid* particles along the boundaries, particles that are not allowed to move. This also has the merit of reducing the error in some SPH formulations (Koshizuka and Oka, 1996), a point which is discussed further on Page 59.

Complex Simulations and Flows With Free Surfaces

Another merit of meshless methods is the ease with which they deal with complex flows (Welton, 1998). However Eulerian methods such as VOF can also handle complex flows and whether meshless methods can do so more easily is questionable. This merit is important in any comparison with meshed-Lagrangian methods (such as the model of Radovitsky and Ortiz (1998)) and to potential theory methods.

SPH simulations containing several fluids, each with different densities, have been discussed in the literature (Monaghan and Kos, 1999). While SPH simulations of breaking waves in systems with both air and water have not been seen by the author, the capability clearly exists.

Also notable regarding SPH is that problems with a free surface require no special treatment in order that boundary conditions be applied, whereas VOF is all about special treatment, something that adds considerably to run-times in such methods. MPS is a little different to SPH on this score and requires that *surface particles* be found in order that the pressure at the free surface can be applied (Koshizuka *et al.*, 1995), a boundary condition not required in SPH.

However with both MPS and SPH the free surface is not defined by a sharp interface as is the case for potential theory and VOF models. Rather, in any simulation, there are areas with particles and areas without, and all that can be said is that the free surface lies in between the two, somewhere close to the particles.

In spite of the complexity of flows that can be simulated with SPH, the models remain relatively simple and it is easy (comparatively) to write source code (Benz, 1990; Monaghan and Kos, 1999). Notably, neither of these two references state what other source codes they compared SPH source against in order to justify this claim.

Run-times in MPS and SPH Simulations

Two factors affect the number of computations required per time-step in SPH and MPS simulations: the number of points and the cost per point. Like VOF, the number of points at which the solution is sought in an $N \times N$ domain is N^2, in comparison to the $4N$ of potential theory models. This means many more computations are required every time-step. In addition, both MPS and SPH need to spend time finding *neighbours* and determining the distances to them, for each particle, the process of determining which particles are involved in calculating the kernel estimates of various properties.

The Accuracy and Other Properties of Kernels

Monaghan (1982) notes that kernel estimation is equivalent to the process of interpolation, and different kernels have different properties with regard interpolation. The choice of kernel is a significant contributor to discretisation errors in SPH models. A kernel spherically symmetric in space (an even function), is capable of at least linear interpolation Monaghan (1982). By careful design kernels can be created that reduce the interpolation error, but other properties of kernels are also significant.

Monaghan and Lattanzio's 1985 ubiquitous cubic spline is an example of a kernel that is fast but less accurate. Alternatively Monaghan (1982) suggest the use of Gaussian and super-Gaussian kernels but while they are more accurate they come at greater computational cost.

Monaghan's cubic spline is an example of a kernel with *compact support*, that is, it has a value of zero outside a certain distance (called the smoothing length) so particles outside the support are not required, accounting for the speed of this kernel. All particles for which the kernel is not equal to zero are called neighbours. Gaussian and super-Gaussian kernels have infinite supports so *every* particle in the system is a neighbour, hence the high cost of such kernels. However given the limited precision of floating point numbers, these kernels have what is effectively a finite support, and requiring lower accuracy from kernel estimations will further reduce this support.

Artificial Clustering

Flat-topped kernels (the gradient in the kernel is zero at its centre) are handy because if two particles end up in the same spot kernel estimates remain well defined. The result of using flat-topped kernels is that a maximum occurs in the repulsive pressure force experienced between 2 adjacent particles, a point that is further discussed in Section 3.2.2. The consequence of this is that artificial clustering may occur. Clustering occurs when particles cannot repel one another with great enough force to stay apart, instead they form a *cluster* of particles with an unphysically high density. Koshizuka *et al.* (1998) uses a kernel inversely proportional to distance (so it is infinite at the origin) that does not have this problem but the accuracy of this kernel is not disclosed. Monaghan (1982) also notes that kernels can be chosen with this consideration in mind.

MPS and SPH models of Breaking Waves

Particle-based Lagrangian methods that have been applied to breaking waves include the SPH models of Monaghan (1994) and of Dalrymple and Knio (2001) and the MPS models of Koshizuka *et al.* (1998) and of Gotoh and Sakai (1999). The predominant theme in all of these works is establishing that the respective methods work, and that they have great potential. As yet neither method is part of the main-stream CFD arsenal for simulating incompressible fluid flow with free surfaces and research is still very much focused on development.

Relatively low numbers of particles are used in the simulations described in these papers, being less than 7500 in all cases. This in comparison to the 75 *million* used by Springel *et al.* (2001) in their simulations of interacting galaxies. This is in part a result of the relatively high numbers of time-steps required of breaking wave models, but is possibly also an indication of the lack of super-computers available to coastal oceanographers.

Wavemaking in MPS and SPH

There are two methods for making waves in MPS and SPH models: the imposition of a wave form in the initial state or the use of wave makers (pistons etc.). Dalrymple and Knio (2001) used

the initial state approach to generate a solitary breaker on a beach with a plane slope, a breaker that is shown in Figure 2.23. The problem with this approach is that in SPH the initial state can have spurious energy present due to the initial particle pressures being incorrect (Monaghan, 1994), a problem often dealt with by preceding a simulation with a ***settling period*** where excess energy is removed. The imposition of a wave form precludes the use of settling periods and the spurious energy may affect results, particularly for low energy simulations.

Dalrymple and Knio (2001) used a piston wave maker with Goring's method (to control the piston trajectory) to generate solitary waves (as well as the initial state approach). Goring's method is discussed in Section 2.1.5. Monaghan (1994) used a piston wavemaker but did not state if special techniques (such as Goring's method) were used to control the trajectory of the piston. An example of a wave generated by Monaghan (1994) is shown in Figure 2.22.

Koshizuka *et al.* (1998) used a variation on the piston in which the amplitudes of boundary particles on the wave maker decrease with depth, an example of which is shown in Figure 2.24. Basically the velocities and positions of the wave particles match the values derived from theory.

Coarse Simulation Features in MPS and SPH

The way that a wave breaks using a model should be similar to a real-world breaker. MPS and SPH simulations are broadly similar to expected behaviour. If a wave breaks in either a plunging or a spilling manner water accelerates up the face of the wave and is projected forward, generating the lip. Under the lip a barrel is formed, the size and shape of which is determined in part by the Iribarren number for the wave. This behaviour should be seen in MPS and SPH simulations and observing this provides a first-order check.

The SPH model of Monaghan (1994) and the MPS models of Koshizuka *et al.* (1998) all show waves in which water accelerates up the face of the breaking wave and is projected outward at the lip with the highest velocities seen in the wave. This feature is confirmed in plots of the velocity vectors which are reproduced in Figures 2.22 and 2.24, and can also be seen in the results of Gotoh and Sakai (1999). This also occurs in the model of Dalrymple and Knio (2001), judging by the existence of the lip and barrel evident in Figure 2.23. So SPH and MPS have been shown to be capable of simulating the coarse features of breaking waves.

Barrels in MPS and SPH Models

A question regarding SPH and MPS models is what range of wave conditions they work for. The way that a wave breaks using a model should reflect real-world behaviour to some degree of accuracy. It is not clear from the literature that SPH and MPS models simulate breaking waves accurately.

Monaghan (1994) simulated periodic waves (of uncertain profile) on a plane beach with a relatively steep slope of 1:10. Monaghan notes that this wave should spill, and his results are consistent with that, containing only a small barrel, shown in Figure 2.25. The wave is well under the maximum wave height of 0.78×depth. A comparison of the results of this model with experimental data for breaking waves is not provided.

Results from a simulation by Dalrymple and Knio (2001) containing a wave generated by using Goring's method are not presented with enough detail to make any conclusions with regard barrel shape. Included in the same reference are the results of an initially unstable solitary wave generated using the initial state approach, shown in Figure 2.23. This wave breaks with a clear (but not extreme) barrel, so it is either a plunging wave or a low Iribarren number spilling wave.

Gotoh and Sakai (1999) show a wave breaking on a plane slope, a wave that they claim categorise as plunging, as is expected given the Iribarren number. A barrel is evident in their plot but it is obscured by unphysical fragmentation of the water body. The wave attains a steep face prior to breaking so it is not spilling. In one frame a barrel is definitely present, but it is

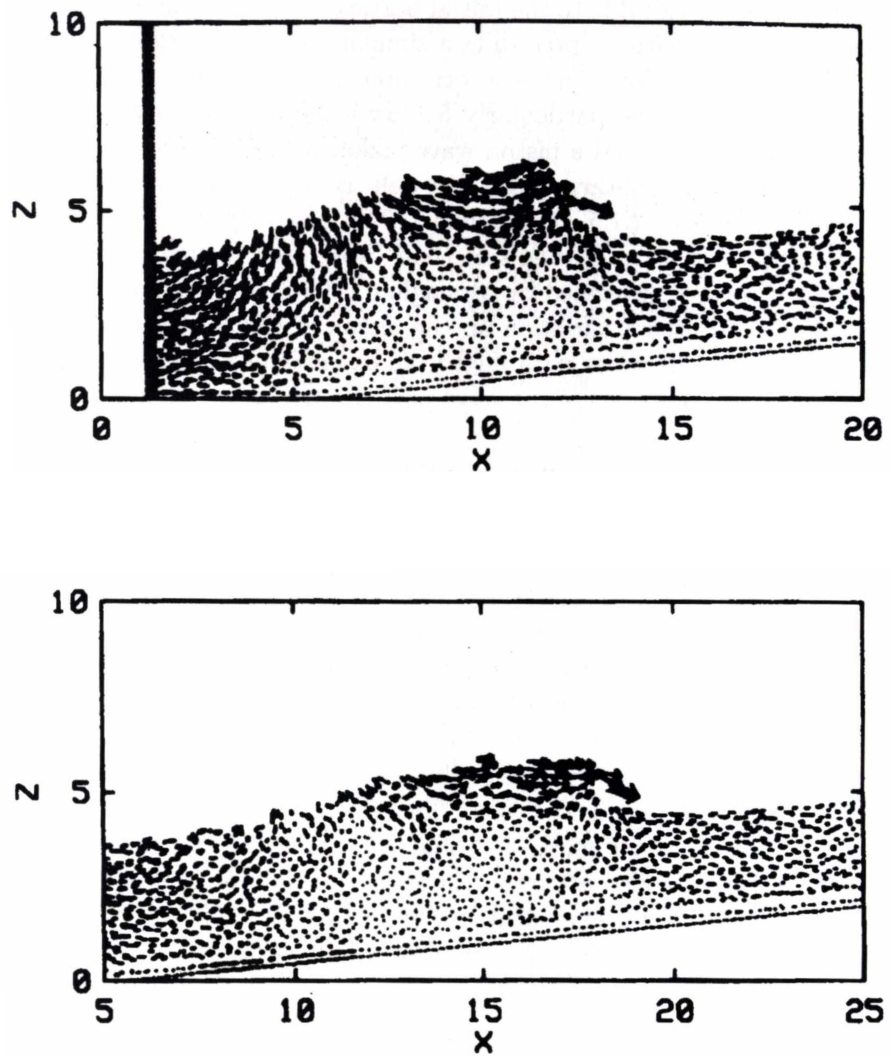

Figure 2.22: *A breaking wave generated in the SPH model of Monaghan (1994). The piston can be seen on the left of the top frame. The arrows are velocity vectors.*

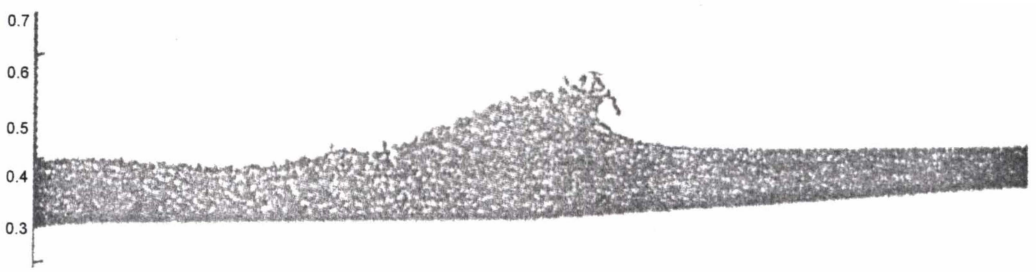

Figure 2.23: *A breaking solitary wave generated by the initial state approach from Dalrymple and Knio (2001). This wave was initially higher than the maximum stable wave height for the depth of the water. Curvature at the left of the image is due to photocopying and scanning.*

Figure 2.24: *An example of a breaking wave generated by Koshizuka* et al. *(1998) using their MPS model. A large section of the domain has been removed for these images but the wave making section remains at the left. The arrows are velocity vectors.*

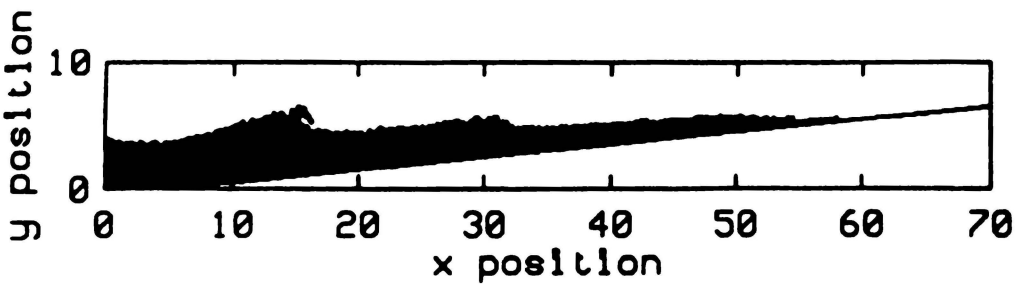

Figure 2.25: *The barrel from Monaghan's (1994) spilling wave.*

not in later frames and hence the simulation appears to depict a collapsing wave. Results from Koshizuka *et al.* (1998) are, if anything, worse. The height of their plunging wave is small with respect to the resolution and the presence of a barrel is a moot point.

Only two of the Lagrangian particle-based simulations discussed in the literature that I have examined show a clear barrel, those being from the SPH models of Monaghan (1994) and of Dalrymple and Knio (2001). One of these is for a spilling wave and the other for an extreme, initially unstable wave. MPS simulations (Koshizuka *et al.*, 1998; Gotoh and Sakai, 1999) do not categorically demonstrate an ability to simulate plunging waves.

To summarise this section, the relevant literature does not definitively demonstrate the ability of SPH and MPS models to accurately simulate a wide variety of breaking waves (but that is not to say they cannot). This could be explained by the low resolutions used for these simulations, the possible existence of numerical effects limiting the applicability of such techniques or simply that not enough simulations have been performed to rigorously test this point.

Summary of Lagrangian Particle-Based Methods

Lagrangian particle-based methods such as SPH and MPS have already been applied to breaking waves, but the literature contains few examples and simulations described in these have low resolution. This is partly a result of the relatively high cost of such simulations (and the subsequent steps taken to avoid this cost), a significant weakness of the methods. The ability to simulate sound in SPH models is not of great use in many cases and has the negative effect of requiring shorter time-steps (than if the fluid were incompressible). Also a problem with Lagrangian particle-based methods is that free surfaces are not defined by sharp curves as is the case for potential theory and VOF models. The strengths of SPH and MPS are the absence of numerical diffusion, the ease of dealing with free surfaces and complex flows. They are also fun, notes Monaghan (1992) "*Although a meal can be enjoyed without understanding the processes of digestion, numerical methods should be both understood and enjoyed.*".

2.2.4 The Marker and Cell Method

Sakai *et al.* (1986) and Takikawa *et al.* (1997) created models (quite possibly these are actually the same model) based on the Marker and Cell (MAC) method, models that they used to simulate breaking waves on plane sloping beaches (both used beaches with a 1:20 slope). Input waves were generated in a coupled FEM (finite element method) model and passed to the MAC models (which solve the Reynolds equations). Takikawa *et al.* (1997) used 70×122 cells and Sakai *et al.* (1986) used 92×50 (for a total of 8540 and 4600 cells respectively) and a large number of markers were used (the exact number was not given in either reference).

The large number of markers used means plots of marker positions appear to have high a resolution, as can be seen in Figure 2.26. This high resolution is necessary in order that the free surface be accurately resolved but gives a false impression that the simulation is accurate. The underlying low resolution grid means any computations performed on the Lagrangian markers have significant discretisation errors. This is because markers are convected by reference to the velocity field computed on the Eulerian grid.

The relatively low resolution of the Eulerian grid (and hence the simulation) are an indication of the high computational cost of the method. It should also be remembered that VOF was developed because MAC was considered too computationally expensive by some researchers. While both VOF and MAC use Eulerian grids to calculate the flow fields, VOF needs only to follow the free surface whereas MAC tracks the entire fluid body (and requires a lot of markers to accurately define the free surface), hence the higher computational cost.

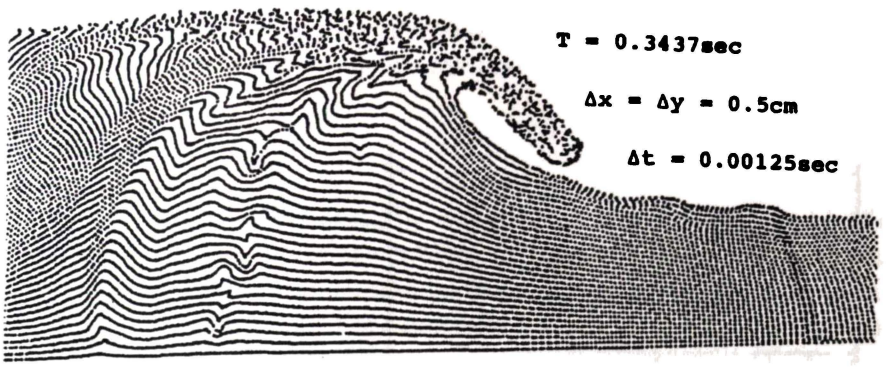

Figure 2.26: *Output from the MAC model of Sakai* et al. *(1986). Note that the apparently high resolution generated from the markers masks the low resolution of the grid.*

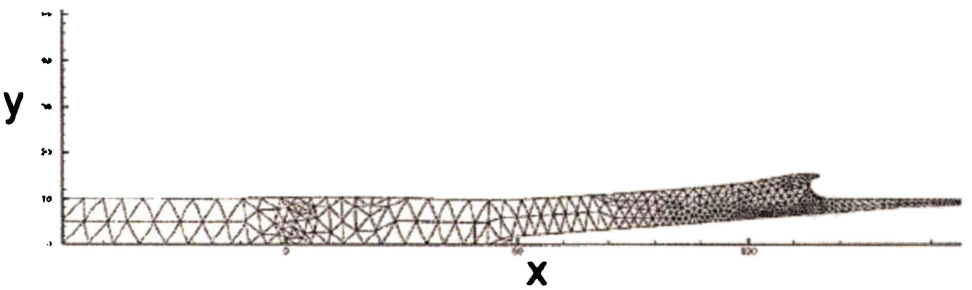

Figure 2.27: *An example of the mesh from the adaptive-meshing Lagrangian model of Radovitsky and Ortiz (1998).*

2.2.5 A Lagrangian Finite Element Model

Radovitsky and Ortiz (1998) created a fully Lagrangian finite element model that they used to simulate breaking waves on beaches with plane slopes. In this model the nodes for each element are convected with the flow, and so the nodes are Lagrangian.

A problem with this approach is that for some flows (like breaking waves), the mesh becomes distorted, and errors increase. Radovitsky and Ortiz (1998) use adaptive meshing to ensure high resolution where necessary, and to avoid an overly distorted mesh. Some researchers have re-gridded meshes when they become too distorted, a step which means the method is semi-Lagrangian. The re-gridding process is computationally expensive and introduces error.

It should be noted that for a computational domain the Lagrangian finite element method requires computations with of the order of $O(N^2)$ nodes, comparable to VOF and Lagrangian particle-based models, whereas a potential theory simulation of the same system would require $O(N)$. An example of the grid is shown in Figure 2.27.

The merit in the approach used by Radovitsky and Ortiz (1998) is that solutions are defined over the entire domain, rather than just at grid points (as is the case for Eulerian Finite difference methods). However the model cannot handle turbulent flow and so they stop their model at the point where the falling lip impacts the face of the wave. Detail of the lip of a wave is shown in Figure 2.28. Note the low resolution and the unexplained *lifting* that has occurred in the lip in the last frames.

So the model used by Radovitsky and Ortiz (1998) suffers similar constraints to those ex-

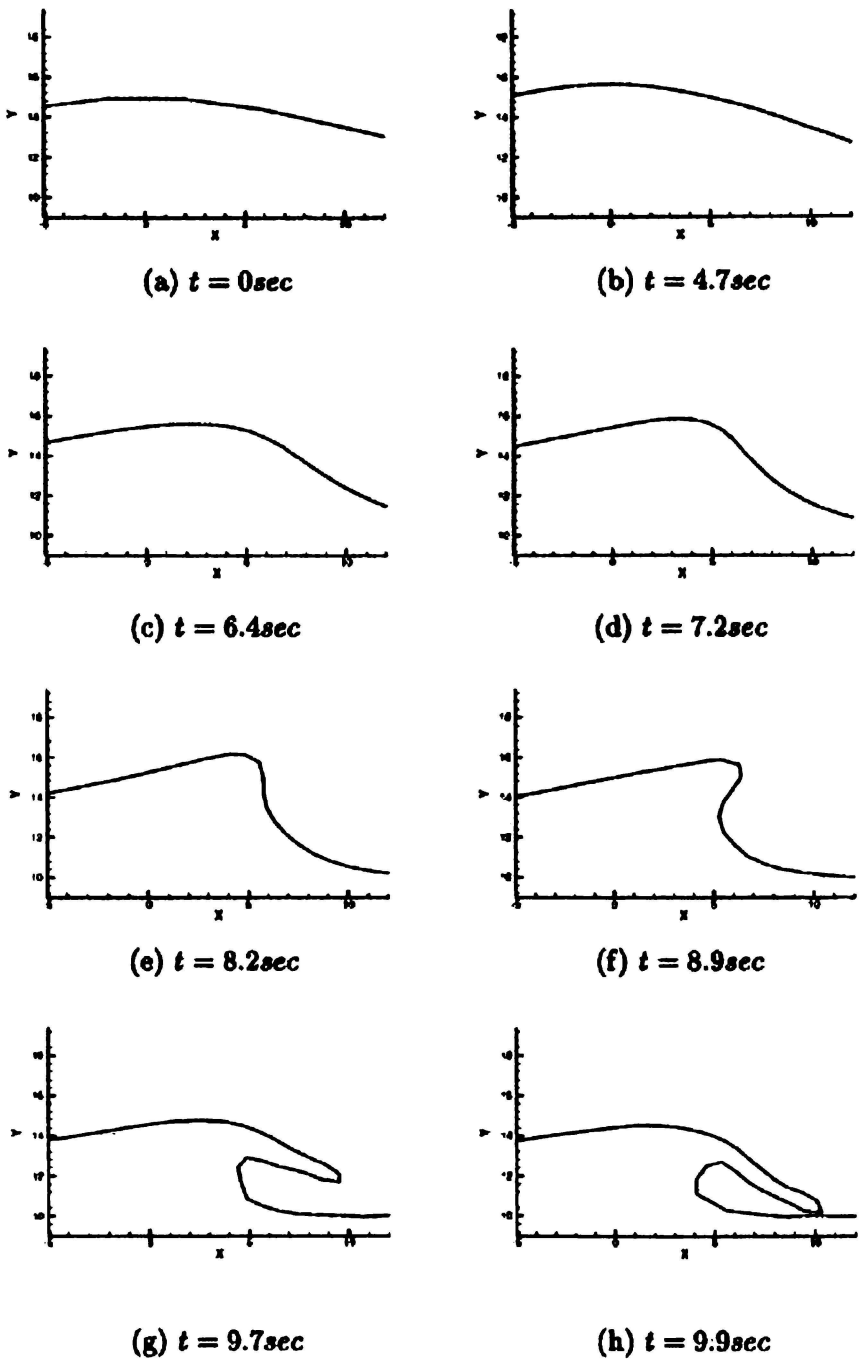

Figure 2.28: *Detail of the lip of a wave from the Adaptive Finite Element Model of Radovitsky and Ortiz (1998).*

perienced by potential theory models (which can't simulate past the lip hitting the wave-face). Also, it will take approximately as long as a VOF model to run (up to the time when the lip hits the wave face). These two issues mean this method is a poor choice.

2.3 Conclusions

The current understanding of how and why waves break is well established but the more subtle details of exactly what determines the profiles of breaking waves are not. In particular the effects of complex bathymetries and the behaviour of real-world breaking waves has not been ascertained. There is an abundance of numerical models in the field already and while many would be capable of answering these questions they have not been used to do so. The merits and capabilities of the various methods used in simulating breaking waves have been examined. It is clear that existing particle-based Lagrangian methods can be used to create models of breaking waves, but it has not yet been shown that they can do so accurately for a wide range of waves. With MAC and VOF these methods are the only ones to have been used for simulating waves after breaking.

Chapter 3

Computational Fluid Dynamics

Given the amount of literature on the computational modelling of breaking waves described in the previous chapter it is evident that computational modelling of breaking waves is by no means new. In reviewing the literature on this topic the previous chapter establishes the current standard of breaking wave models. The suitability of a particle-based Lagrangian method for this task must be ascertained by a comparison to existing models. In the previous chapter existing wave models were reviewed to provide a framework for this comparison.

The aim of this chapter is to develop an understanding of Lagrangian particle-based methods, emphasising the differences from other methods. The merits and demerits of the Lagrangian particle-based methods are examined and following this step a decision can be made regarding the basis for the model to be created here.

A review of some basic ideas from computational fluid dynamics is given, with a focus on basic principles. Particle-based Lagrangian methods that have been used to study fluid flows are examined. A nomenclature is included at the end of this chapter for the reader's convenience. Variables and notation that appear in only one subsection are not included in the nomenclature.

3.1 Methods in Numerical Modelling

All methods in computational fluid dynamics (CFD) are based on the discretisation of the governing equations derived from the fundamental principles of physics (Anderson, 1995; Ferziger and Perić, 1996). The form of the governing equations when discretisation is performed determines whether a method is Eulerian, Lagrangian or Composite Eulerian-Lagrangian.

Here, the term *discretisation* refers to discretisation in space, while discretisation in time is referred to as time-stepping. Many of the errors that plague CFD are a result of discretisation or the time-stepping method (Oran and Boris, 1987).

In this section the terms Eulerian and Lagrangian are introduced, discretisation and time-stepping methods are examined, and some discussion on errors and run-times is presented.

3.1.1 Eulerian and Lagrangian Formulations

The terms Eulerian and Lagrangian represent different formulations, different methods by which a fluid can be modelled. The selection of a formulation is based on how well it fits the problem in hand, and on the merits and demerits particular to its application. While the choice of formulation is independent of the choice of discretisation method to some degree, some constraints do exist. An appropriate topic with which to start discussion of the formulations is the Lagrangian derivative.

The Lagrangian Derivative

Suppose a fluid has a property $\psi(\boldsymbol{r}, t)$ and a velocity field $\boldsymbol{v}(\boldsymbol{r}, t)$. The Lagrangian derivative is defined by (Batchelor, 1967)

$$\frac{d\psi}{dt} = \lim_{\tau \to 0} \frac{\psi(\boldsymbol{r} + \boldsymbol{v}\tau, t + \tau) - \psi(\boldsymbol{r}, t)}{\tau}. \tag{3.1}$$

After performing a Taylor expansion on the term $\psi(\boldsymbol{r} + \boldsymbol{v}\tau, t + \tau)$ about $\psi(\boldsymbol{r}, t)$, and keeping only the first order terms, the Lagrangian derivative becomes

$$\frac{d\psi}{dt} = \frac{\partial \psi}{\partial t} + \boldsymbol{v}.\nabla\psi. \tag{3.2}$$

This relatively simple operator has an obscene number of names, also being known as the Stokes, material, substantive or convective derivative. It represents the time derivative of a property following the motion of the fluid.

The Eulerian Approach

Eulerian approaches are those in which the governing equations are discretised with respect to a mesh (perhaps a finite difference grid or a finite element mesh), a mesh that is stationary in time. To draw an analogy, Eulerian methods are like looking at a river while standing on the banks. Governing equations for Eulerian models will consist of a number of equations (derived from the conservation principles) with the form given by

$$\frac{\partial \psi}{\partial t} = -\boldsymbol{v}.\nabla\psi + \text{other terms} \tag{3.3}$$

in which ψ is a general property (it might be density, energy or velocity), and the first term on the right hand side is the convective term. The derivative on the left is a partial derivative with respect to time, with position held constant.

Discretisation methods available for use in Eulerian models are very well researched and include the control volume approach, and the finite difference and finite element approaches (Ferziger and Perić, 1996). The finite difference technique is notable because it can be understood intuitively, and can be related to the definition for the derivative introduced in first-year calculus. Finite element methods are also popular and have the advantage over finite difference methods of being more readily adapted to non-uniform meshes (Ferziger and Perić, 1996).

Many people have the ability to write source codes based on Eulerian meshes, code is widely available, and it can be tailored to suit a particular application. It is the detailed literature and familiarity of these discretisation techniques that makes Eulerian methods so common, and so well accepted. A notable demerit of Eulerian methods is that they suffer from numerical diffusion, an error generated by the discretisation of convective terms in the balance equations (Oran and Boris, 1987).

The Lagrangian Approach

In Lagrangian approaches the governing equations are written in terms of points that are convected with the fluid. To continue the analogy, Lagrangian approaches are like jumping into the river and examining it as one is carried along. The governing equations for Lagrangian models consist of a number of equations with the general form given by Equation 3.3

$$\frac{d\psi}{dt} = \text{other terms}, \tag{3.4}$$

in which the "other terms" are the same as those in Equation 3.4 and the derivative is a Lagrangian derivative.

Discretisation techniques include the Lagrangian Particle-based methods, meshless finite element method, and Lagrangian potential flow models. The particle-based approach can be termed the control mass approach since the fluid domain is broken into *pieces* which have mass constant in time. Both meshless finite element and Lagrangian potential flow involve points that are convected with the fluid, points from which the mesh and boundary (respectively) are constructed.

Lagrangian methods are not so widely used or studied as Eulerian methods, so source code, literature, and expertise are not as common. With the exception of the meshless methods, these methods experience severe limitations in the range of scenarios to which they can be applied. In comparison to Eulerian methods the governing equations for Lagrangian models contain no convective terms and so experience no numerical diffusion once discretised, as can be seen by comparing Equations 3.3 and 3.4.

The Composite Eulerian-Lagrangian Approach

Composite methods are those where a mixture of Lagrangian and Eulerian ideas are used, but to qualify for this group, a method needs to make reference to an Eulerian mesh. In the river analogy these methods are like jumping into the river to *go with the flow*, then jumping back out, then back in, then back out ... These methods are typically used in order to avoid problems associated with totally Eulerian or totally Lagrangian approaches, but often end up suffering the demerits of both.

Examples of such methods are the marker and cell, particle and cell and those methods that use Lagrangian meshes which are redrawn when they become too distorted. The marker and cell method (Harlow and Welch, 1965) uses massless markers that interact with an Eulerian grid in order to calculate free surface positions. The particle and cell method (Harlow, 1964) uses control masses that interact with an Eulerian grid in order to calculate flow fields. Lagrangian meshes are useful only where distortions to the mesh remain small, otherwise errors escalate, a problem that some researchers avoid by drawing a nice new mesh (Radovitsky and Ortiz, 1998).

A fourth alternative is available: the arbitrary Euler Lagrange approach. This involves solution points that are convected with arbitrary velocities (Hirt, 1970).

Composite Eulerian-Lagrangian methods are perhaps the least studied of all the methods. Research into totally Eulerian and totally Lagrangian methods has progressed to the point that they can be used to solve most problems.

3.1.2 Discretisation Methods

The right hand sides of Equations 3.3 and 3.4 contain terms with continuous derivatives, equations that are usually too complex to solve analytically, and to solve them using computational methods they must be discretised. Discretisation involves rewriting the governing equations in discrete, rather than continuous, forms. There are a number of ways that this can be done, some of which are discussed here. The motivation for this review is to demonstrate how the particle-based methods differ from other methods.

The Finite Difference Method

The use of the finite difference methods requires the specification of a grid, which in the simplest case, a structured grid, simply consists of orthogonal lines: a Cartesian grid (Ferziger and Perić, 1996). Such grids are simple and easily conceptualised but for more complex flows, those with curved boundaries for example, they are not so good since they require high resolutions to

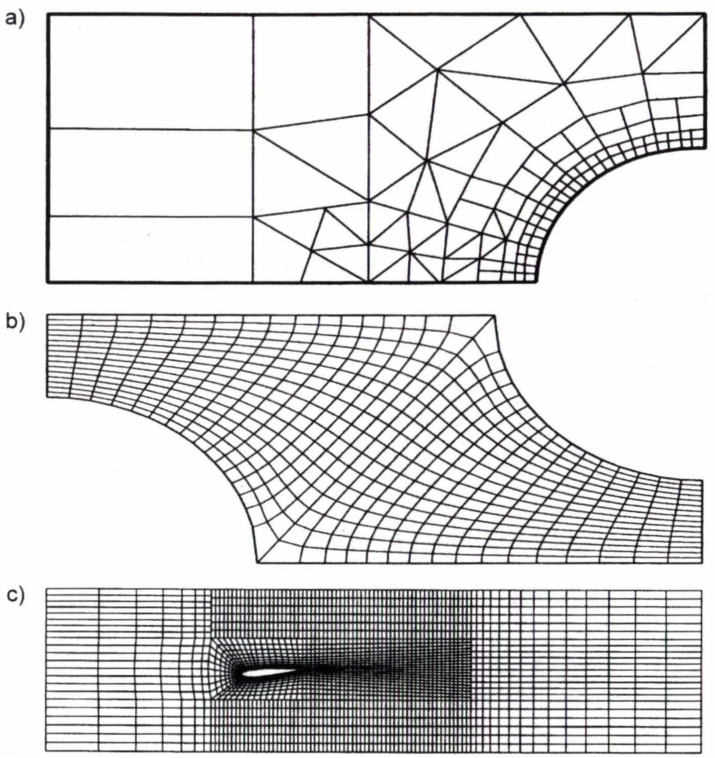

Figure 3.1: *An unstructured grid* (a), *a non-orthogonal grid* (b), *and a block-structured grid* (c) *(from Ferziger and Perić, 1996).*

approximate boundaries well. Also, because the resolution is uniform, the grid may offer higher resolution than necessary in some places, and not enough in others (Ferziger and Perić, 1996).

Since researchers often wish to simulate complex flows, improvements over structured grids are necessary, improvements provided by block structuring and non-orthogonal grids, examples of which are shown in Figure 3.1. Also possible are adaptive girds in which the grid changes with time (Oran and Boris, 1987). Progressing to unstructured grids removes the simplicity of the method, but this is a necessary cost if the method is to be used to solve complex problems. Generating grids is the topic of considerable volumes of literature, and it is by no means a trivial task.

In the finite difference method spatial derivatives in the governing equations are approximated by difference equations, an example being the two point centred difference equation for the derivative with respect x of ψ at the grid point x_i (this example is written for a structured grid)

$$\frac{\partial \psi(x_i)}{\partial x} \approx \frac{\psi(x_{i+1}) - \psi(x_{i-1})}{2h} \tag{3.5}$$

where $h = (x_{i+1} - x_i)$ is the distance between two adjacent points. In the limit as $h \to 0$ Equation 3.5 becomes exact but for $h > 0$ ***discretisation errors*** are introduced. Equation 3.5 is just a single possible example of a difference equation, many others being available, using as many of the neighbouring points as desired. Schemes using four or five points are common.

In comparison to meshfree methods, finite difference methods have the subtle merit that the points involved in a calculation are always known: if an N-point difference at a node is desired, then the N points required just need to be counted off. By contrast particle-based Lagrangian methods need to spend ***significant*** amounts of time finding exactly what other particles are required in any calculation.

The Finite Volume Method

A variation of the finite difference method is the finite volume method, where the governing differential equations, are integrated over each grid cell. Using quadrature rules to approximate the resulting integrals (Ferziger and Perić, 1996), the governing equations are written in terms of the values of variables at grid vertices. Ferziger and Perić (1996) suggests that this is the most simple of the methods to program and understand, and notes that boundaries can be well approximated, but that higher order (more accurate) solutions are difficult to develop.

The Finite Element Methods

This family of methods are an extension to the finite volume method, but have the merit of being easily able to handle unstructured grids, such as that shown in Figure 3.1, a merit that means they can be used to simulate complex domains on arbitrary geometries, with higher resolutions where required (Ferziger and Perić, 1996). In this method the solution is represented by a simple function (such as a polynomial) that approximates the solution over an element. This function is evaluated from the as-yet unknown values of variables at the nodes (Ferziger and Perić, 1996). The problem then is to determine the values at the nodes from the governing equations, a difficult task to achieve efficiently (Ferziger and Perić, 1996).

Meshfree Methods

Not surprisingly many of the meshfree methods are extensions of the families of methods discussed in the preceding section. These approaches require specification of points within the computational domain but there are no requirements as to the distribution, which is the case for finite element and finite difference methods. While some meshfree methods have been applied to fluids, others have been used for simulating complex solids such as those experiencing large deformation, cracking and fragmentation where some methods fail.

The generalised finite difference method (Tworzydlo, 1987, 1989) involves choosing the number of points to be involved in a computation (say sixteen, for example), dividing the neighbourhood of a particular point into sectors centred on that point (eight sectors, to continue the example) then finding the closest points (two, in the example) in each sector. The author is not aware that this method has been used with the Lagrangian formulation to simulate fluids.

An alternative to finite differences are the kernel methods, such as SPH and MPS. In these methods the value of a variable is determined by a kernel estimate. This has the effect of smoothing the value of a variable, and hence the name. These two meshless methods are closely related and have been applied to many problems involving fluids.

There are a number of other meshfree methods including the reproducing kernel particle method (Liu *et al.*, 1995) (another kernel method with similarities to SPH), which is an improvement over SPH, but while it is more accurate, it takes longer to run. Also the method of least squares (Duarte, 1995) has been applied to arbitrary distributions of points and hence used to discretise partial differential equations. Notable are finite element methods for arbitrary distributions of points: the h-p clouds method (Duarte, 1995) and the partition of unity (Babuška and Guo, 1996) being examples. The element-free Galerkin method is also available to modellers (Belytschko *et al.*, 1994).

There is a meshfree equivalent of BEM called the local boundary integral equation (LBIE) approach (Fries and Matthies, 2003). Where BEM uses a *global* boundary integral equation (an integral performed over the entire domain), the local version uses an LBIE in which the integral is performed over a sub-domain. The LBIE approach is summarised by Fries and Matthies (2003).

It is evident that meshfree methods have points of similarity, and in grouping these methods by their similarities, different researchers have arrived at different conclusions as to what families

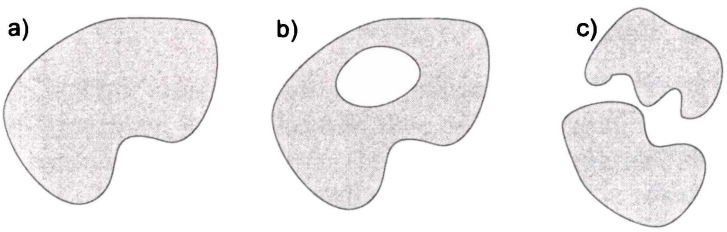

Figure 3.2: *Examples of a simply connected domain* (a), *a multiply connected domain* (b), *and an unconnected domain* (c).

exist, and to which methods belong to which families (for example Duarte, 1995; Liu *et al.*, 1995; Belytschko *et al.*, 1996; Fries and Matthies, 2003).

Potential Flow Theory

At the heart of potential flow methods, such as the boundary element method (BEM) (Grilli *et al.*, 1989), is the definition of the velocity potential, a potential that satisfies the Laplace equation everywhere within the domain. Using any of a number of methods, the governing equations are evaluated only at the surface of the domain (to generate, using BEM as an example, the boundary integral equations), greatly reducing the time required to find solutions. However requirements for the existence of the potential include the conditions of irrotationality and simple connectedness of the domain (Guignard *et al.*, 2001).

The term *simply connected* requires definition. Firstly a definition of a connected domain is given by (Weisstein, 1999)

> A space D is connected if any two points in D can be connected by a curve lying wholly within D.

A definition for a simply connected domain is provided by (Churchill and Brown, 1990)

> A simply connected domain D is a domain such that every simple closed contour within it contains only points within D. If a domain is connected, but not simply connected then it is multiply connected.

Examples of these are shown in Figure 3.2. In potential flow methods, discretisation occurs when the boundary integral equations are evaluated by dividing the boundary into elements.

Time-Stepping Methods

In fluid mechanics solutions are often desired for governing equations that are expressed as sets of first order differential equations

$$\frac{dy_{(j)}}{dt} = f_{(j)}(y_{(1)}, ..., y_{(N)}, t) \tag{3.6}$$

for $j = 1 : N$, where N is the total number of equations, $f_{(j)}$ are functions in terms of the variables $y_{(j)}$. Because solutions are often not available analytically, numerical methods are required, methods that can be collectively called, ***time-stepping methods*** (but only when time is the dependent variable however), but more generally as methods for ordinary differential equations.

Time-stepping represents discretisation in time, but is distinct from discretisation in space in that nothing is known about future time-steps. Time-stepping methods are introduced here very briefly. Examples of time-stepping methods include the explicit Euler and the families of

Runge-Kutta, predictor-corrector and multi-step methods (Lambert, 1973). Hybrid methods that combine features of the Runge-Kutta and predictor-corrector methods are also available (Lambert, 1973).

Consider the initial value problem for a single (for simplicity) first-order differential equation (Lambert, 1973)

$$\frac{dy}{dt} = f(y,t), \qquad y(t_0) = y_0. \tag{3.7}$$

in which the subscript 0 indicates the initial state of the system, notation that is used elsewhere within this section. By integrating with respect to time from t_i to t_{i-1}, this becomes

$$\int_{t_{i-1}}^{t_i} \frac{dy}{dt} dt = y_i - y_{i-1} = \int_{t_{i-1}}^{t_i} f(y,t) dt \tag{3.8}$$

in which the notation that $y_{i-1} = y(t_{i-1})$ has been used, a convention that is used for the function f_i also. So the problem has reduced to integrating $f(y,t)$. From here it is easy to see the generation of the explicit Euler method (the simplest of all time-stepping methods), where a left-hand rule to approximate the integral (Ferziger and Perić, 1996)

$$y_i = y_{i-1} + \tau f_i \tag{3.9}$$

in which $\tau = t_i - t_{i-1}$ is the time-step. This approach is easy to program and is fast but inaccurate, being the least accurate of all methods with a local error $O(\tau^2)$ (Ferziger and Perić, 1996). To obtain an approximation for y_f at the final time-step (t_f), $(t_f - t_0)/\tau$ steps are required and the global error in y_f is $O(\tau)$, which is why the explicit Euler method is referred to as being ***first order***. However, worse than the large errors, is that the explicit Euler is not ***A-stable*** (Lambert, 1973), which in some systems (a simple pendulum for example) causes an unphysical growth in energy.

To solve the problem of low accuracy a corrector step can be employed, where the value of y_i estimated using the Explicit Euler is used with a mid-point rule to re-evaluate the integral of Equation 3.8

$$y_i^* = y_{i-1} + \tau f_{i-1} \tag{3.10}$$

$$y_i = y_{i-1} + \frac{1}{2}\tau(f_{i-1} + f(y^*(t_i), t_i)) \tag{3.11}$$

where the first line is the predictor stage and the second line is the corrector stage, describing a 2-stage predictor corrector method. The superscript $*$ indicates the value is a intermediary value. Monaghan (1994) employs this method with the approximation

$$f_{i-1} \approx [f(y^*(t_i), t_i)]_{\text{previous step}}. \tag{3.12}$$

This modification reduces the number of computations required but introduces a local error $O(\tau^3)$, the same order as for the unmodified 2-stage predictor-corrector method. By using additional corrector steps higher accuracy can be achieved (Lambert, 1973) (provided τ is small enough), but at some stage improvements are better made by reducing the time-step.

Alternatively more points can be used in the evaluation of the integral, that is employing a multi-step method. This might use information for the current time-step ***and*** the ***preceding*** time-step to predict information for the ***next*** time-step. Multi-step methods require special care when starting, since at the first time-step only the initial condition exists. Multi-step methods may be ***started*** by using a one-step method. Actually, predictor-corrector methods can be created that use multi-step methods for each stage, that is, there is no constraint requiring that the predictor step be the explicit Euler or that the mid-point rule be used for the corrector (Lambert, 1973).

Runge-Kutta methods are a family of one step methods, so they require information for only one time-step in order to predict the next, but with a bit of mathematical trickery (and many more computations) they are able to do so more accurately than a method like the explicit Euler (Lambert, 1973). Significantly these methods can be made A-stable (Lambert, 1973).

Once a time-stepping method has been chosen, higher accuracy can be attained by using more points (for a multi-step method), using more repetitions (for predictor-corrector methods) or by using a higher-order method (for the Runge-Kutta family). Inevitably though, higher accuracy is achieved at the cost of longer run-times.

3.2 Smoothed Particle Hydrodynamics

This thesis required the generation of a Lagrangian particle-based model for simulating breaking waves in shallow water. To this end a choice was required regarding the discretisation and time-stepping methods for the model. SPH, the method used by Monaghan (1994) and by Dalrymple and Knio (2001), was also chosen here, a choice that needs to be justified. The intricacies of how the method works are discussed here.

3.2.1 A Justification for Choosing SPH

Given the need (in this work) to create a Lagrangian particle-based model, any of the meshless methods could have been utilised with the Lagrangian formulation for the conservation equations. The choice here is based on what other researchers had done in the field, the author's abilities as a programmer, and in particular the nature of this thesis. This thesis is an examination of the use of such a Lagrangian particle-based model in the study of coastal oceanography, so it requires a conclusion regarding their suitability for the task. The aim is not to make improvements or test variations for numerical techniques in fluids: this is not a mathematics thesis. For this reason techniques were avoided for which applicability to free surface fluid flows were not verified.

SPH was chosen since it best satisfied these constraints. Following the lead of Monaghan (1994) and Dalrymple *et al.* (2001) the 2-stage predictor-corrector was chosen as the time-stepping method, and was implemented with the time-saving procedure described by Monaghan (1994), in which only one function evaluation is required per time-step. So in principle the models used by Monaghan (1994) and by Dalrymple *et al.* (2001) should generate the same results as the model created here, but in practice slight differences occur.

3.2.2 Division of the Model Domain

For a Lagrangian particle-based model the computational domain must be divided into control masses, often referred to as particles, for which mass is constant in time. SPH determines the density of the fluid from the positions of particles, and as such, requires that the entire fluid body be accounted for in the division of the domain.

But it must be noted that these particles are macroscopic and bear no relation to microscopic particles such as molecules and atoms. For each particle the mass can be considered as being distributed about its centre of mass according to the kernel (Benz, 1990). Kernels are discussed in more depth on Page 51.

In standard SPH for each particle a, for a from 1 to N the position $\boldsymbol{r}_a$, velocity $\boldsymbol{v}_a$, mass m_a, density ρ_a and pressure P_a are required. Density is required because in the standard application of SPH to weakly compressible flows (further discussed on Page 58) the density does change. In MPS (an SPH variant) the density is not required since the fluid is incompressible. Particle mass might conceivably vary between particles, but here they are held constant.

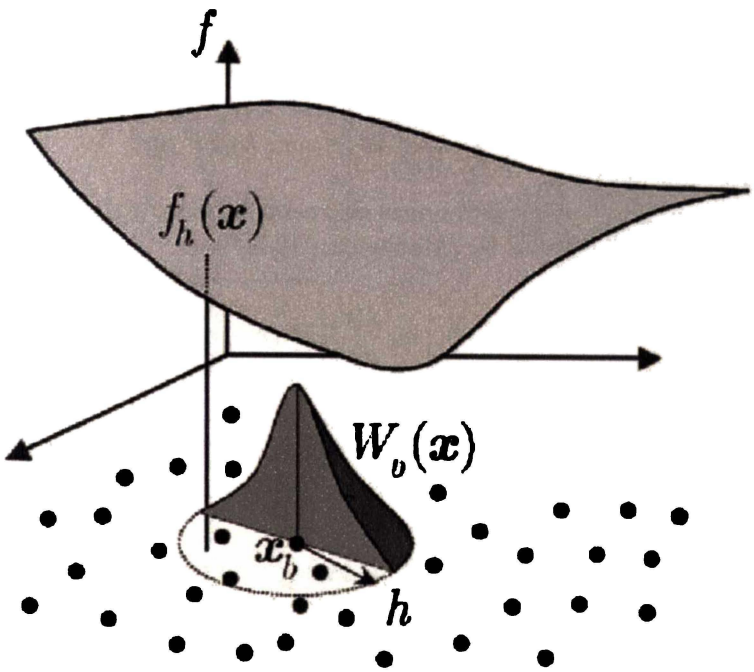

Figure 3.3: *How the summation interpolation works (after Bonet* et al., *2004). Here the summation interpolant is denoted by $f_h(x)$ rather than $f_S(x)$, which is used in this thesis, and at position x the kernel centred at x_b has value $W_b(x)$. At any point x all of the particles within the support contribute, here one of these particles is labeled b.*

The Integral Interpolant

The integral interpolant is often the first step in the introduction of SPH. For the arbitrary function $f(\boldsymbol{r})$ on the domain D the integral interpolant is given by (Monaghan, 1992)

$$f_I(\boldsymbol{r}) = \int_D f(\boldsymbol{r}')W(\boldsymbol{r} - \boldsymbol{r}', h)d\boldsymbol{r}'. \tag{3.13}$$

Here W is a kernel and is a function of the distance between $\boldsymbol{r}$ and $\boldsymbol{r}'$, and h, the smoothing length which determines the width of the kernel. Kernels are discussed further on Page 51.

In the literature Equation 3.13 is referred to by different names, including a kernel estimation for $f(\boldsymbol{r})$, the integral interpolant for $f(\boldsymbol{r})$ and the smoothed value of $f(\boldsymbol{r})$. Kernels are chosen so that they are normalised, that is

$$\int_D W(\boldsymbol{r}', h)d\boldsymbol{r}' = 1 \tag{3.14}$$

and also so that

$$\lim_{h\to 0} f_I(\boldsymbol{r}) = f(\boldsymbol{r}). \tag{3.15}$$

But in SPH this limit is never attained, instead $h > 0$ is used, an approximation which is the source of a significant discretisation error (this is further in Page 51). In practice the functional form of $f(r)$ is seldom known, so integral interpolation cannot be used as it stands, instead Equation 3.13 is approximated by a summation interpolant. Figure 3.3 shows how a function is approximated by kernel summation.

The Summation Interpolant

To derive the summation interpolant the integral interpolant for the general function $f(\boldsymbol{r})$ is rewritten as

$$f_I(\boldsymbol{r}) = \int_D \frac{f(\boldsymbol{r}')}{\rho(\boldsymbol{r}')} W(\boldsymbol{r} - \boldsymbol{r}', h)\rho(\boldsymbol{r}')d\boldsymbol{r}'. \tag{3.16}$$

The domain is divided into N discrete volumes $d\boldsymbol{r}_a$ centred at $\boldsymbol{r}_b$ with mass $m_b = \rho(\boldsymbol{r}_b)d\boldsymbol{r}_b$, then Equation 3.16 can be approximated by (Monaghan, 1988)

$$f_S(\boldsymbol{r}) = \sum_{b=1}^{N} m_b \frac{f(\boldsymbol{r}_b)}{\rho(\boldsymbol{r}_b)} W(\boldsymbol{r} - \boldsymbol{r}_b, h). \tag{3.17}$$

The summation need only be performed over the neighbours of point $\boldsymbol{r}$, the points that lie within the support of the kernel centred at $\boldsymbol{r}$. In an alternative derivation Equation 3.16 can be approximated using a quasi-Monte Carlo estimate

$$f_S(\boldsymbol{r}) = \sum_{b=1}^{N} \frac{f(\boldsymbol{r}_b)}{n(\boldsymbol{r}_b)} W(\boldsymbol{r} - \boldsymbol{r}_b, h) \tag{3.18}$$

where $n(\boldsymbol{r}_b)$ is the number density of particles at $\boldsymbol{r}_b$ and with the definition

$$\begin{aligned} \rho(\boldsymbol{r}_b) &\equiv m_b n(\boldsymbol{r}_b) \\ &= m \sum_{a=1}^{N} W(\boldsymbol{r}_b - \boldsymbol{r}_a, h) \end{aligned} \tag{3.19}$$

from which Equation 3.17 follows.

The Error in the Summation Interpolant

For kernels with compact support the number of neighbours (M) is less than the total number of points (N), and only contributions from the neighbours are nonzero. So Equation 3.16 can be rewritten

$$f_I(\boldsymbol{r}) = \int_{\text{support}} \frac{f(\boldsymbol{r}')}{\rho(\boldsymbol{r}')} W(\boldsymbol{r} - \boldsymbol{r}', h)\rho(\boldsymbol{r}')d\boldsymbol{r}'. \tag{3.20}$$

where the integral limit "support" indicates the integral need only be performed over the support of the kernel (that part of the domain in which the kernel is nonzero), rather than the entire domain. The quasi-Monte Carlo estimate of this integral is

$$f_S(\boldsymbol{r}) = \sum_{b=1}^{M} \frac{f(\boldsymbol{r}_b)}{n(\boldsymbol{r}_b)} W(\boldsymbol{r} - \boldsymbol{r}_b, h) \tag{3.21}$$

in which the summation is performed over neighbours. If the points are random then the error is (Niederreiter, 1992)

$$O\left(\frac{1}{\sqrt{M}}\right). \tag{3.22}$$

Monaghan (1982) asserts that this rate of convergence for the error seems a little slow, and notes that if the points are considered to be disordered and equi-distributed then the error (in d dimensions) in this is (Niederreiter, 1992)

$$O\left(\frac{(\ln M)^{d-1}}{M}\right). \tag{3.23}$$

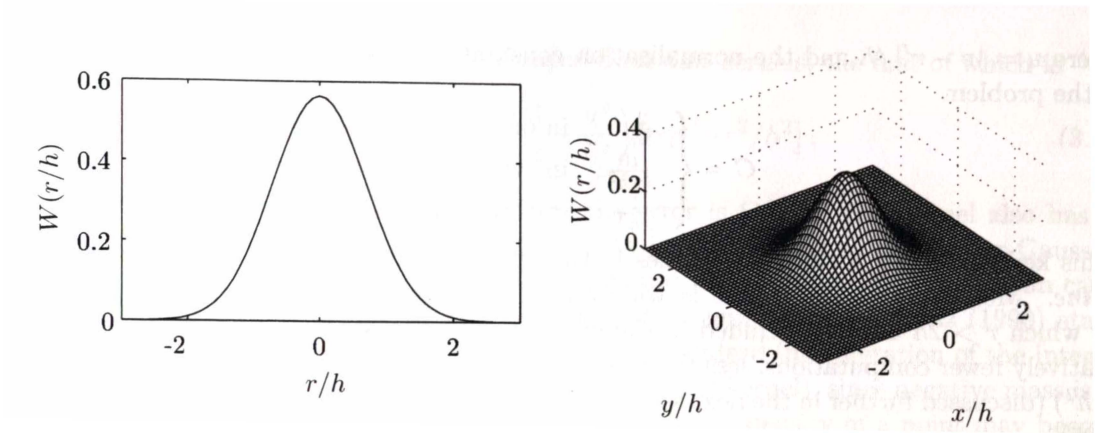

Figure 3.4: *The Gaussian kernel in one dimension on the left, and two on the right. Although very small for large r, the kernel is non-zero for all r.*

This assertion rests on the assumption (Monaghan (1982) notes the value a proof would provide) that the points are disordered but equi-distributed, but there is no proof that points in an SPH simulation necessarily conform to such a distribution. The number of neighbours can be approximated by $M \propto Nh^d$ so the error becomes

$$O\left(\frac{1}{\sqrt{Nh^d}}\right) \tag{3.24}$$

or

$$O\left(\frac{(\ln Nh^d)^{d-1}}{Nh^d}\right). \tag{3.25}$$

It is most likely that the error in and SPH estimate lies somewhere between $\mathrm{O}\left(1/\sqrt{Nh^d}\right)$ and $\mathrm{O}\left((\ln Nh^d)^{d-1}/Nh^d\right)$, but without a proof that the later is true, we should take a conservative approach and use the former.

Kernels

Fundamental to kernel methods is the kernel itself, the nature of which is a significant factor in determining the accuracy of the method. Monaghan (1982) gives an example of a kernel:

$$W(\boldsymbol{r}, \boldsymbol{r}', h) = C \exp\left[-(\boldsymbol{r} - \boldsymbol{r}')^2/h^2\right], \tag{3.26}$$

in which C is a normalisation constant chosen so that Equation 3.14 is satisfied. For this kernel $C = \frac{1}{h\sqrt{\pi}}$ for a one dimensional problem. This kernel is plotted in Figure 3.4. The problem with this kernel is that the support is infinite, that is, every point in the entire domain make non-zero contributions, albeit that contributions further from the centre get small rapidly. Consequently it is computationally expensive, but this cost can be reduced by truncating the support to those points which make significant contributions.

An alternative is the cubic spline suggested by Monaghan and Lattanzio (1985)

$$W(u) = C \begin{cases} 1 - \frac{3}{2}u^2 + \frac{3}{4}u^3; & 0 \le u \le 1 \\ \frac{1}{4}(2-u)^3; & 1 < u \le 2 \\ 0; & 2 < u < \infty \end{cases} \tag{3.27}$$

where $u = |\boldsymbol{r} - \boldsymbol{r}'|/h$ and the normalisation constant C depends on the number of dimensions in the problem

$$C = \begin{cases} \frac{2}{3h} & ; \text{ in one dimension} \\ \frac{10}{7\pi h^2} & ; \text{ in two dimensions} \\ \frac{1}{\pi h^3} & ; \text{ in three dimensions} \end{cases} \tag{3.28}$$

This kernel is given a variety of names in the literature and here it is called *Monaghan's cubic spline*. Monaghan's cubic spline is widely used because it has a compact support (particles for which $r > 2h$ are not included in the calculation) and consisting of polynomials it requires relatively fewer computations, less than an exponent, for example. The accuracy of this kernel is $O(h^2)$ (discussed further in the next section), and it is positive definite (Monaghan and Lattanzio, 1985).

Kernel Accuracy

The accuracy of the kernel is determined by how well the integral interpolant (Equation 3.13) approximates $f(\boldsymbol{r})$. If the kernel were perfect (an unachievable ideal) it would be exact, that is, $f_I(\boldsymbol{r}) \equiv f(\boldsymbol{r})$.

The accuracy of a kernel can be understood by noting that a Taylor expansion can be performed on $f(\boldsymbol{r}')$ about $\boldsymbol{r}$ in Equation 3.13 (Monaghan, 1982). While the generalisation to two or three dimensions is not difficult, it is simpler to examine the one dimensional case, for which the Taylor expansion of $f(r')$ about r is (Monaghan, 1982)

$$\begin{aligned} f_I(r) &= \int_{-\infty}^{\infty} W(r-r',h)\left[f(r) + f'(r)(r'-r) + \frac{f''(r)}{2!}(r'-r)^2\right. \\ &+ \frac{f^{(3)}(r)}{3!}(r'-r)^3 + \frac{f^{(4)}(r)}{4!}(r'-r)^4 + \ldots + \frac{f^{(N)}(r)}{N!}(r'-r)^N \\ &+ \left.\frac{f^{N+1}(c)}{(N+1)!}(r-r')^{N+1}\right] dr', \end{aligned} \tag{3.29}$$

the last term being the remainder term and in which c lies between r and r'.

By making the substitution $u = r - r'$ and after fiddling with integral limits, Equation 3.29 becomes

$$\begin{aligned} f_I(r) &= f(r)\int_{-\infty}^{\infty} W(u,h)du - f'(r)\int_{-\infty}^{\infty} uW(u,h)du + \frac{f''(r)}{2!}\int_{-\infty}^{\infty} u^2W(u,h)du \\ &- \frac{f^{(3)}(r)}{3!}\int_{-\infty}^{\infty} u^3W(u,h)du + \frac{f^{(4)}(r)}{4!}\int_{-\infty}^{\infty} u^4W(u,h)du + \ldots \\ &+ \frac{f^{(N)}(r)}{N!}\int_{-\infty}^{\infty}(-u)^NW(u,h)du + \frac{f^{(N+1)}(c)}{(N+1)!}\int_{-\infty}^{\infty}(-u)^{N+1}W(u,h)du. \end{aligned} \tag{3.30}$$

The integrals are known as the moments of the kernel. At this point Monaghan (1982) noted that the normalisation condition usually demanded of kernels means the zeroth moment is 1, and if the kernel is radially symmetric all odd moments are zero. The error in the integral interpolant is determined by the lowest non-zero even moment (usually the second). Using Lagrange's form for the remainder one obtains

$$f_I(r) = f(r) + \frac{f''(c)h^2}{2!}\int_{-\infty}^{\infty} u^2W(u)du = f(r) + O(h^2) \tag{3.31}$$

in which c is a constant lying within the support for the kernel. For Monaghan's cubic spline (Equation 3.27) the second moment is 1/3, and for the Gaussian it is 1/2. However Equation 3.31 fails to account for the lack of consistency, a point that is explained in Section 3.3.

Monaghan (1982) notes that the family of ***super-Gaussian*** kernels, the first of which is

$$W(r,h) = \frac{1}{(\pi h^2)^{3/2}}\left(\frac{3}{2} - \frac{u^2}{h^2}\right)\exp\left[-r^2/h^2\right], \tag{3.32}$$

achieve higher accuracy, and for the example here the error is $O(h^4)$. This kernel also has an infinite support and so, like the Gaussian, it is computationally expensive. The super-Gaussian kernels change sign for some values of r and Monaghan and Kos (1999) note that this can cause problems for simulations involving, for example, high Mach number flows. Benz (1990) states that using kernels that are not positive definite voids the standard interpretation of the integral interpolant (mass distributed about the centre according to the kernel), since negative mass is an unphysical idea. Also problematic with such kernels is that the density of a point may become negative (Benz, 1990).

Properties of the Integral Interpolant

It can be seen that

$$\left(\frac{\partial f(\boldsymbol{r})}{\partial t}\right)_I = \frac{\partial f_I(\boldsymbol{r})}{\partial t} \tag{3.33}$$

if the domain is constant in time. If instead $D(t)$ an additional term is introduced, shown in the derivation of the SPH equation for conservation of mass (see Section 3.2.3). It can also been seen that

$$\nabla_r \left(f(\boldsymbol{r})\right)_I = \int_D f(\boldsymbol{r}')\nabla_r W(\boldsymbol{r}-\boldsymbol{r}',h)d\boldsymbol{r}' \tag{3.34}$$

where ∇_r indicates the gradient is taken with respect to the global coordinates $\boldsymbol{r}$, and NOT the dummy coordinates $\boldsymbol{r}'$. Integration by parts shows that

$$\begin{aligned}\left(\nabla_r f(\boldsymbol{r})\right)_I &= \int_D \left(\nabla_{r'} f'\right) W(\boldsymbol{r}-\boldsymbol{r}',h)d\boldsymbol{r}' \\ &= \int_D \nabla_{r'}\left(f' W(\boldsymbol{r}-\boldsymbol{r}',h)\right)d\boldsymbol{r}' - \int_D f'\left(\nabla_{r'} W(\boldsymbol{r}-\boldsymbol{r}',h)\right)d\boldsymbol{r}'.\end{aligned} \tag{3.35}$$

Because $\nabla_{r'} W(\boldsymbol{r}-\boldsymbol{r}',h) = -\nabla_r W(\boldsymbol{r}-\boldsymbol{r}',h)$, and after an application of the divergence theorem, this becomes

$$\begin{aligned}\left(\nabla f\right)_I &= \int_D f(\boldsymbol{r}')\left(\nabla_r W(\boldsymbol{r}-\boldsymbol{r}',h)\right)d\boldsymbol{r}' - \int_{\partial D} f(\boldsymbol{r}')W(\boldsymbol{r}-\boldsymbol{r}',h)\boldsymbol{n}dS \\ &= \nabla_r\left(f(\boldsymbol{r})\right)_I - \int_{\partial D} f(\boldsymbol{r}')W(\boldsymbol{r}-\boldsymbol{r}',h)\boldsymbol{n}dS\end{aligned} \tag{3.36}$$

Here ∂D is the boundary of D, being one or more surfaces if D is three dimensional, and one or more closed curves if D is two dimensional. In general, researchers have assumed the boundary integral in 3.36 is negligible (Monaghan, 1988; Benz, 1990), but this is not necessarily a good assumption for fluid mechanics where the fluid is constrained by a boundary (Benz, 1990) and may have a free surface. When a kernel with a compact support is used the errors deriving from neglecting the surface terms occur only near the boundaries, but these errors will propagate into the fluid with time. These missing boundary integrals are a topic that deserves more research.

The kernel estimate of the material derivative can be given by (Monaghan, 1982; Benz, 1990), using a Taylor expansion for the convective term

$$\begin{aligned}\left(\frac{df(\boldsymbol{r})}{dt}\right)_I &= \frac{\partial f_I(\boldsymbol{r})}{\partial t} + \int_D \boldsymbol{v}(\boldsymbol{r}').\left(\nabla_{r'} f(\boldsymbol{r}')\right) W(\boldsymbol{r}-\boldsymbol{r}',h)d\boldsymbol{r}' \\ &= \frac{\partial f_I(\boldsymbol{r})}{\partial t} + \boldsymbol{v}(\boldsymbol{r}).\left(\nabla_r f(\boldsymbol{r})\right)_I + O(h^2)\end{aligned} \tag{3.37}$$

and by replacing the exact values of the velocity and function gradient this becomes

$$\begin{aligned}\left(\frac{df(\boldsymbol{r})}{dt}\right)_I &= \frac{\partial f_I(\boldsymbol{r})}{\partial t} + \left[\boldsymbol{v}_I(\boldsymbol{r}) + O(h^2)\right] . \left[(\nabla_r f(\boldsymbol{r}))_I + O(h^2)\right] + O(h^2) \\ &= \frac{\partial f_I(\boldsymbol{r})}{\partial t} + \boldsymbol{v}_I(\boldsymbol{r}) . (\nabla_r f(\boldsymbol{r}))_I + O(h^2) \end{aligned} \tag{3.38}$$

so finally

$$\left(\frac{df(\boldsymbol{r})}{dt}\right)_I = \frac{df_I(\boldsymbol{r})}{dt} + O(h^2). \tag{3.39}$$

Using similar techniques it can be shown that

$$\left(\frac{f(\boldsymbol{r})}{g(\boldsymbol{r})}\right)_I = \frac{f_I(\boldsymbol{r})}{g_I(\boldsymbol{r})} + O(h^2). \tag{3.40}$$

Properties of the Summation Interpolant

From Equation 3.36 with the assumption that the surface integrals are negligible, it can be shown that

$$\nabla_r f_S(\boldsymbol{r}) = \sum_{b=1}^{N} m_b \frac{f(\boldsymbol{r}_b)}{\rho_b} \nabla_r W(\boldsymbol{r} - \boldsymbol{r}_b, h). \tag{3.41}$$

This means that to evaluate the gradient in a property, the gradient in the kernel needs to be evaluated, a relatively simple task that is possible provided the kernel is differentiable, which it always is.

Gradients from Summation Interpolants

At this point it is interesting to note a departure from SPH used by MPS. Rather than take advantage of Equation 3.41, MPS employs (Koshizuka *et al.*, 1995, for example)

$$\nabla f_S(\boldsymbol{r}) = \sum_{b=1}^{N} \nabla \left(m_b \frac{f(\boldsymbol{r}_b)}{\rho_b} \right) W(\boldsymbol{r} - \boldsymbol{r}_b, h), \tag{3.42}$$

in which $\nabla(m_b \frac{f(\boldsymbol{r}_b)}{\rho_b})$ is approximated by what is essentially a two point difference scheme where the differences are forward differences in the variable r. So in MPS the gradient in a property is the smoothed value of approximations to the local gradient, but the two approaches (Equations 3.41 and 3.42) could be used interchangeably, in either SPH or MPS. This expression can be thought of as the smoothed value of two-point finite difference approximations. It has the notable merit that it does not cause artificial clustering (see section 3.2.2) when flat-topped kernels are used. Also (a point that will make more sense once consistency has been discussed, later in this chapter)

Koshizuka *et al.* (1995) do not state the error in Equation 3.42, but do provide a comparison of performances by their expression and by Equation 3.41 in evaluating the derivative of a quadratic at a number of random points. They conclude their expression is better, but in this author's opinion their conclusion cannot be justified from the data they present. Notable is that Equation 3.42 performs poorly near the limits of the domain, which should be expected of a forward difference scheme near boundaries. Poor performance near boundaries by Equation 3.41 should also be expected due to the neglected boundary integrals, but this is not reflected in the comparison, since it is poor everywhere.

It is significant that in MPS the summation interpolant is modified to ensure normalisation, that is, they have used *Shepard interpolation*, a method that is discussed along with the idea

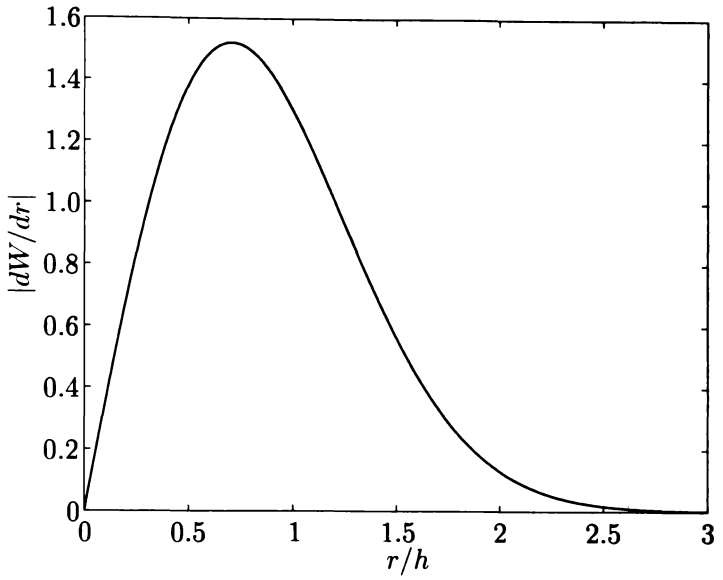

Figure 3.5: *The magnitude of the gradient of the Gaussian kernel, for which a maxima occurs at a finite distance from the origin. The existence of the maxima is the cause of artificial clustering.*

of consistency on Page 63. This approach improves the accuracy of the kernel estimation by ensuring that zeroth order consistency is achieved. Details of the expression used by Koshizuka *et al.* (1995) to generate their SPH data are not given, and one suspects that it was not modified to ensure zeroth order consistency.

The Formation of Artificial Clusters

The formation of artificial clusters is an undesirable effect that occurs when flat-topped kernels are used (a description that only makes sense in one or two dimensions). Artificial clustering was first noted by Schussler and Schmitt (1981) who coined the term used here, but it has been referred to by other names elsewhere. This effect is discussed here because of the relation it bears to choice of kernels and their accuracy.

The Gaussian kernel (Equation 3.26, plotted in Figure 3.4) is an example of a flat-topped kernel. The magnitude of the kernel gradient is plotted in Figure 3.5, and the zero gradient at the centre is clear, as is the maxima that causes artificial clustering.

The total pressure force at a particle is dependent on the gradient in the pressure, and if Equation 3.41 is used, the pressure gradient is dependent on the gradient in the kernel at each of its neighbours. Provided the pressures of the particle and its neighbour are not zero, the particle always repels its neighbours. This means that if two particles are pushed together the force with which they repel each other will increase to a certain point and then decrease. When they pile on top of one another the repulsive force is zero and an artificial cluster has been generated.

To avoid artificial clustering the method of Koshizuka *et al.* (1995) (Equation 3.42) could be used or a kernel with a monotonically decreasing gradient could be employed. A kernel cannot have a monotonically decreasing gradient and be flat-topped. The merit in flat topped kernels is quite simply that the gradient in the kernel is defined everywhere, so in the unlikely event that one particle sits on top of another, it does not matter. This is not true for the monotonically decreasing kernel however. It appears to be a trade off, computationally secure but cluster-generating kernels, or cluster-proof but high-risk kernels.

An example of a kernel with a monotonically decreasing gradient is the inverse function used

by Koshizuka and Oka (1995) in their MPS model

$$W(r,h) = \begin{cases} \frac{2h}{r} - 1; & 0 \leq \frac{r}{h} \leq 2 \\ 0; & 2 < \frac{r}{h} < \infty \end{cases} \tag{3.43}$$

which has the dubious distinction of being infinite for $r = 0$ (and where, consequently the gradient is also undefined), has a discontinuity in the gradient at $r = 2h$, as well as being non-integrable. Not being integrable means the kernel cannot be normalised , which means the kernel cannot be used in SPH. Another example from the literature also has a discontinuous gradient at the outer edge of its support (Schussler and Schmitt, 1981)

$$W(r,h) = C \begin{cases} 1 - \left(\frac{r}{2h}\right)^{1/2}; & 0 \leq \frac{r}{h} \leq 2 \\ 0; & 2 < \frac{r}{h} < \infty \end{cases} \tag{3.44}$$

In this kernels C is a constant determined by requiring normalisation.

Morris *et al.* (1997) note that SPH stability is strongly dependent on the second derivative of the kernel, so the discontinuities in the gradients of the kernels in Equations 3.43 and 3.44 will adversely affect simulations. If second derivatives are required to be continuous then Monaghan's cubic spline cannot be used either.

Alternatively, to avoid artificial clustering, Thomas and Couchman (1992) use Monaghan's cubic but modify its gradient so that it is non-zero at the origin

$$\frac{dW(r)}{dr} = C \begin{cases} 4; & 0 \leq r \leq 2/3 \\ 3r(4-3r); & 2/3 \leq r \leq 1 \\ 3(2-r)^2; & 1 < r \leq 2 \\ 0; & 2 < r < \infty \end{cases} \tag{3.45}$$

in which C is the normalisation constant and $W(r)$ is left unmodified. The value four in the interval $0 \leq r \leq 2/3$ is chosen so that the kernel gradient is continuous. Steinmetz (1996) also uses this modified form for the kernel gradient and notes that while it is only marginally stable (an assertion they do not further justify), it has the advantage of enabling larger step-sizes to be used.

Errors in the Initial State

When an SPH simulation is created lists of particle positions and other properties need to be specified, and these will probably be generated by simple methods like placing the points at the vertices of a regular grid. The danger in this approach is that particles may be given incorrect properties, so spurious energy will have been added to the system, and if it is not removed it may dominate the behaviour of the system (Monaghan, 1994).

For a weakly compressible fluid with a constant gravity constrained by boundaries there are three sources of spurious energy: spurious sound waves, incorrect elevation of the water body and excess boundary potential energy.

Spurious sound waves are generated when particles are assigned incorrect initial densities. If particles are given incorrect positions the system can have spaces where densities are too low to generate the pressure gradients required to oppose gravity. Quite simply, the water body is initially too high or it has holes in, so the system has spurious gravitational potential energy. Boundary potential energies can be too great if particles are inadvertently placed too near the boundaries.

These spurious energies are transformed to kinetic energy or to sound waves as the simulation progresses and are either dissipated by artificial viscosity or are present as noise. To avoid simulations with unphysical behaviour caused by errors in the initial state Monaghan (1994)

employed a smoothing algorithm to remove spurious energy. In Monaghan's algorithm the system was allowed to evolve with a very large viscosity, till a significant portion of the spurious energy had been damped. From this state the simulation could be started and run without significant errors. The problem with this method of removing spurious energies is that a significant number of time-steps may be required to achieve a sufficiently smooth initial state.

3.2.3 Derivation of the SPH Equations

Several steps are required in the generation of the set of governing equations for an SPH model, a derivation that is examined here. The starting point of this derivation is writing the conservation equations, from which point the process of discretisation proceeds.

The Balance Equations

Fundamental to computational fluid mechanics are the conservation equations, equations which are derived from the conservation principles (Ferziger and Perić, 1996). There are four principles of interest here, the principles of conservation of angular momentum, linear momentum, energy and mass.

The first two of these derive from Newton's second and third laws. The principle of conservation of energy is the first law of thermodynamics and conservation of mass is also known as the principle of continuity. With these principles in hand the corresponding conservation equations can be derived in differential form, and the process of discretisation can begin.

The equation of conservation of mass is given by

$$\frac{d\rho}{dt} = -\rho \nabla . \boldsymbol{v} \tag{3.46}$$

where it has been assumed that there are no sources or sinks, that is, no mass is generated or destroyed. If radioactivity is to be accounted for (for example) source and sink terms would have to be added. If more than one type of fluid is present (a system with oil and water, for example) then a continuity equation is needed for each, but such systems are not considered here. Details are important so it should be noted that relativistic and quantum effects are not accounted for.

If it is assumed that the fluid is isothermal then energy in the system is stored as gravitational potential energy, kinetic energy, work and dissipated energy. The first two energies are easily understood and can be calculated readily once positions and velocities are found. The last of these is the energy dissipated by viscosity, constructed to increase stability of the method, energy that must be accounted for using the equation of energy. Energy is stored as work (internal energy) because the system contains a compressible fluid. It is the work required to compress or expand the fluid. The equation of conservation of energy is given by

$$\frac{du}{dt} = -\frac{P}{\rho} \nabla . \boldsymbol{v} \tag{3.47}$$

in which u is the internal energy.

The principle of conservation of angular momentum is of little importance in SPH, since most implementations of the method do not account for the angular momentum of a single particle. In order to account for the rotation of a single particle it is this principle that needs to be invoked. This principle is used however, in some correction techniques, namely Corrected SPH (Bonet and Lok, 1999) (CSPH).

From the principle of conservation of momentum the equation of conservation of momentum can be derived, given by

$$\frac{d\boldsymbol{v}}{dt} = -\frac{\nabla P}{\rho} + \boldsymbol{g} \tag{3.48}$$

where $\boldsymbol{g}$ is a body force per unit of mass, usually just the acceleration due to gravity.

Also important, though not deriving from a conservation principle but from the definition for the velocity is

$$\frac{\partial \boldsymbol{r}}{\partial t} = \boldsymbol{v}. \tag{3.49}$$

The only remaining requirement is an equation of state.

The Equation of State

In traditional SPH the density is related to the pressure by an equation of state, so the fluid is assumed compressible and its compressibility selected as a model input (Monaghan, 1994). This is quite a departure from standard hydrodynamic modelling where the pressure is determined from a Poisson equation for the pressure, meaning that the fluid has been assumed incompressible. In this section the subscript 0 indicates a property evaluated for the system in the reference state.

An equation of state (these are usually empirical or semi-empirical in origin) relates state variables in a thermodynamic system, the state variables being parameters that describe the temperature, composition, density and pressure. For an isothermal system with constant composition the equation of state just relates density and pressure. Monaghan (1994) rewrites the equation of state given by Batchelor (1967) so that pressures are gauge pressures in Pascal,

$$P = B\left(\left(\frac{\rho}{\rho_{ref}}\right)^{\gamma} - 1\right) \tag{3.50}$$

where ρ_{ref} is a reference density, $\gamma = 7$ and B is a coefficient that determines the speed of sound. A number of researchers (Monaghan, 1994; Bonet and Lok, 1999; Monaghan and Kos, 1999; Dalrymple and Knio, 2001) have used this equation in their SPH models for simulating free-surface flows. A reference state for the system can be arbitrarily defined, though it should be defined with simplicity in mind, as having a density ρ_{ref} ($1000\,\mathrm{kg/m^3}$), a gauge pressure of 0 Pa at which the velocity of sound is c_{ref}. At these conditions the speed of sound (c_{ref}) is measured to be around 1500 m/s.

In the reference state the coefficient B is given by

$$B = \frac{c_{ref}^2 \rho_{ref}}{\gamma}. \tag{3.51}$$

The maximum Mach number (M_{max}) is given by (Monaghan and Kos, 1999)

$$M_{max} = \frac{v_{max}}{v_{ref}} \tag{3.52}$$

where v_{max} is the maximum velocity likely in a simulation, so the coefficient B becomes (Monaghan and Kos, 1999) (Monaghan (1994) is erroneous)

$$B = \frac{v_{max}^2 \rho_{ref}}{M_{max}^2 \gamma}. \tag{3.53}$$

The speed of sound is

$$c^2 = \frac{dP}{d\rho} = \frac{\gamma P}{\rho} + \frac{\rho_{ref} v_{max}^2}{\rho M_{max}^2} \tag{3.54}$$

In practice much lower values for the speed of sound are used than those measured experimentally, because the speed of sound is a significant factor affecting the Courant condition which limits step-size. For a high speed of sound a short time-step is needed, greatly increasing model

run-times. However the lower speed of sound introduces variations in the maximum density ($\delta\rho/\rho$) given by (Monaghan, 1994)

$$\frac{\delta\rho}{\rho} = O\left(\frac{v_{max}^2}{c_{ref}^2}\right) = O\left(M_{max}^2\right) \tag{3.55}$$

So choosing a Mach number of 0.1 means maximum relative density variations are ~ 1 .

The Density from a Summation Interpolant

Using Equation 3.17 the density can be estimated by

$$\rho_S(\boldsymbol{r}) = \sum_{b=1}^{N} m_b W(\boldsymbol{r} - \boldsymbol{r}_b, h) \tag{3.56}$$

which ensures conservation of mass provided the total number and the masses of particles remain constant (Monaghan and Lattanzio, 1985).

Benz (1990) interprets Equation 3.56 as meaning every particle has its mass smoothed in space and the kernel represents the distribution of its mass, and $\rho_S(\boldsymbol{r})$ is the sum of all contributions at $\boldsymbol{r}$.

Particularly in astrophysical applications, Equation 3.56 is used in the place of the continuity equation, but for fluids in which the density falls discontinuously to zero at boundaries, it causes problems and flows will feature spurious variations at the boundaries (Monaghan, 1994).

The Continuity Equation

Rather than endure the poor performance of Equation 3.56 Monaghan (1994) used an expression derived from the continuity equation, the derivation of which is given here. The continuity equation (Equation 3.46) can be written in the form

$$\frac{\partial\rho(\boldsymbol{r},t)}{\partial t} = -\nabla_r.\left(\rho(\boldsymbol{r},t)\boldsymbol{v}(\boldsymbol{r},t)\right) \tag{3.57}$$

The operator ∇_r is defined by (acting on the unspecified property ψ)

$$\nabla_r\psi = \frac{\partial\psi}{\partial r_x}\boldsymbol{i} + \frac{\partial\psi}{\partial r_y}\boldsymbol{j} + \frac{\partial\psi}{\partial r_z}\boldsymbol{k} \tag{3.58}$$

in which each partial derivative is taken while holding time constant. The continuity equation is satisfied at all points within the domain of the simulation, so it can also be evaluated a point $\boldsymbol{r}'$. It is helpful to define another operator (acting on the unspecified property ψ)

$$\nabla_{r'}\psi = \frac{\partial\psi}{\partial r'_x}\boldsymbol{i} + \frac{\partial\psi}{\partial r'_y}\boldsymbol{j} + \frac{\partial\psi}{\partial r'_z}\boldsymbol{k}. \tag{3.59}$$

To proceed, Equation 3.57 is evaluated at $\boldsymbol{r}'$, multiplied by the kernel $W = W(\boldsymbol{r}, \boldsymbol{r}', h)$, and integrated over the entire domain with respect to the variable $\boldsymbol{r}'$:

$$\int_{V(t)} \frac{\partial\rho'}{\partial t} W d\boldsymbol{r}' = -\int_{V(t)} \nabla_{r'}.\left(\rho'\boldsymbol{v}'\right) W d\boldsymbol{r}' \tag{3.60}$$

in which $\rho' = \rho(\boldsymbol{r}', t)$ and $\boldsymbol{v}' = \boldsymbol{v}(\boldsymbol{r}', t)$. By applying integration by parts this becomes

$$\frac{\partial\rho_I}{\partial t} = -\int_{V(t)} \nabla_{r'}\left(\rho'\boldsymbol{v}'W\right) d\boldsymbol{r}' + \int_{V(t)} \rho'\boldsymbol{v}'.\nabla_{r'}(W) d\boldsymbol{r}' - \lim_{\Delta t\to 0}\int_{\Delta V(t)} \rho' W d\boldsymbol{r}' \tag{3.61}$$

in which the final integral on the right hand side results from recognising that in free surface fluid flows the volume of the domain is not constant. The volume ΔV is defined by $\Delta V(t) = V(t + \Delta t) - V(t)$ (p. 439 Adams, 1999). The divergence theorem can be used to transform the first integral on the right hand side, and if $\boldsymbol{v}.\nabla_r \rho_I$ is added to both sides, equation 3.61 becomes

$$\frac{\partial \rho_I}{\partial t} + \boldsymbol{v}.\nabla_r \rho_I = \int_{V(t)} \rho' \boldsymbol{v}.\nabla_r W \boldsymbol{dr}' + \int_{V(t)} \rho' \boldsymbol{v}'.\nabla_{r'} W \boldsymbol{dr}' - \int_{S(t)} \left(\rho' \boldsymbol{v}' W\right) .\boldsymbol{n} dS - \lim_{\Delta t \to 0} \int_{\Delta V(t)} \rho' W \boldsymbol{dr}' \tag{3.62}$$

in which n is the outward pointing normal unit vector to the surface S.

If (and only if) the kernel gradient is antisymmetric, then $\nabla_r W = -\nabla_{r'} W$ and this becomes

$$\frac{d\rho_I}{dt} = \int_{V(t)} \rho' \left(\boldsymbol{v} - \boldsymbol{v}'\right) .\nabla_r W \boldsymbol{dr}' - \int_{S(t)} \left(\rho' \boldsymbol{v}' W\right) .\boldsymbol{n} dS - \lim_{\Delta t \to 0} \int_{\Delta V(t)} \rho' W \boldsymbol{dr}'. \tag{3.63}$$

The final two integrals are normally neglected (discussed in section 3.2.2), and by approximating the remaining integrals by Monte Carlo estimates the discretised continuity equation for a particle at position $\boldsymbol{r}_a$ is obtained

$$\frac{d\rho_a}{dt} = \sum_b m_b (\boldsymbol{v}_a - \boldsymbol{v}_b) \nabla W(\boldsymbol{r}_a, \boldsymbol{r}_b, h). \tag{3.64}$$

But Equation 3.57 is just one starting point for this derivation, others being found by the use of various vector identities. An alternate and unfavoured form is

$$\frac{d\rho_a}{dt} = -\sum_b m_b \boldsymbol{v}_b \nabla W(\boldsymbol{r}_a, \boldsymbol{r}_b, h). \tag{3.65}$$

If two particles have the same velocity this alternate form still predicts a change in density, but Equation 3.64 is only better because it gives a non-zero density change if the velocities of the two particles are different. This is possibly the justification for Monaghan's (1992) *second golden rule of SPH*, which is the assertion that equations should be rewritten so that the density only occurs inside operators.

Benz (1990) states that the continuity equation given here means that energy and momentum are not conserved exactly, because the number density for a particle b is no longer given by

$$n_b = \frac{\rho_b}{m_b}. \tag{3.66}$$

Monaghan (1992) states that using the continuity equation rather than the summation interpolant for the density means mass is no longer conserved exactly, but doesn't support this statement with any reasoning. To the contrary, Monaghan (1982) asserts that using the continuity equation results in conservation of mass both globally (so net mass is conserved) and locally.

When the summation interpolant for the density is integrated over the entire domain the result is $M = M$, a statement that the net mass M is conserved. Equation 3.63 can be integrated over the entire domain to obtain $\partial M/\partial t = 0$, which is also a statement that net mass is conserved. But only the continuity equation enforces conservation of mass at the local scale. It is the opinion of teh author that the summation interpolant for the density should never be used as a replacement for the continuity equation.

The Momentum Equation

The derivation for the equation of conservation of momentum follows similar lines to that for the continuity equation. Monaghan (1992) gives one form as being

$$\frac{d\boldsymbol{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Pi_{a,b} \right) \nabla_a W(\boldsymbol{r}_{ab}, h) + \boldsymbol{g} \tag{3.67}$$

in which $\Pi_{a,b}$ is the artificial viscosity introduced to improve numerical stability and to mimic physical viscosity. The subscript a, b indicates that the viscosity term has a functional dependence on the properties of both particle a and b and ∇_a indicates the gradient is taken with respect to the coordinates $\boldsymbol{r}_a$. In this expression the notation $\psi_{ab} = \psi_a - \psi_b$ and $\overline{\psi_{ab}} = (\psi_a + \psi_b)/2$ for the general property ψ have been used. A second form is given by Bonet and Lok (1999) as

$$\frac{d\boldsymbol{v}_a}{dt} = -\sum_b m_b \left(\frac{P_b + P_a}{\rho_a \rho_b} + \Pi_{a,b} \right) \nabla_a W(\boldsymbol{r}_{ab}, h) + \boldsymbol{g}. \tag{3.68}$$

These two equations each have their own advantages and disadvantages. Equation 3.67 for particles with the same pressure does not calculate a zero pressure gradient (Monaghan, 1992). For an isolated pair of particles with different pressures, Equation 3.68 predicts they will ***bootstrap*** themselves to infinity (Monaghan, 1992).

The form of the viscosity term is somewhat arbitrary and many different schemes have been proposed. The scheme used by Monaghan (1994) in his paper describing simulations of breaking waves is,

$$\Pi_{a,b} = \frac{-\alpha \overline{c_{ab}} \mu_{a,b}}{\overline{\rho_{ab}}} \begin{cases} 1 & \boldsymbol{v}_{ab} \cdot \boldsymbol{r}_{ab} < 0 \\ 0 & \boldsymbol{v}_{ab} \cdot \boldsymbol{r}_{ab} > 0 \end{cases} \tag{3.69}$$

where, as for $\Pi_{a,b}$, the subscript of $\mu_{a,b}$ indicates it is dependent on the properties of particles a and b:

$$\mu_{a,b} = \frac{\boldsymbol{v}_{ab} \cdot \boldsymbol{r}_{ab}}{\boldsymbol{r}_{ab}^2 + \eta h^2} \tag{3.70}$$

The term ηh^2 is added artificially to avoid singularities that would occur whenever $\boldsymbol{r}_{ab}^2 = 0$, and generally $\eta = 0.01$. One might choose $\alpha = 0.01$, or thereabouts, depending on the application (Monaghan, 1994).

In one dimension, in the limit as the number of particles becomes infinite and the smoothing length decreases to zero, Lattanzio *et al.* (1986); Monaghan and Kos (1999) note that this viscosity has the same form as the Navier-Stokes viscosity. By this comparison Monaghan and Kos (1999) determined that the kinematic viscosity can be written as

$$\nu = \frac{1}{8} \alpha h c. \tag{3.71}$$

The kinematic viscosity (ν) is measured to be around 1×10^{-6} m^2/s for water at 20°C (Batchelor, 1967).

While the physical viscosity accounts for the dissipation of kinetic energy to heat, using a value for α derived from the measured kinematic viscosity may not provide sufficient damping to remove noise from particle velocities. Consequently the kinematic viscosity is generally chosen 100 to 1000 times larger than measured values (Monaghan and Kos, 1999).

The XSPH Variant

XSPH is a variation of SPH developed to reduce penetration in colliding clouds of particles (Monaghan, 1989). It involves a modification to the SPH equation for particle position, given

by (Monaghan, 1989)

$$\frac{\partial \boldsymbol{r}_a}{\partial t} = \boldsymbol{v}_a + \epsilon \sum_b \frac{m_b \boldsymbol{v}_{ab} W_{ab}}{\overline{\rho_{ab}}}. \tag{3.72}$$

Here the second term is the contribution of the XSPH variant and $0 < \epsilon < 1$ is an arbitrary unitless constant (unknown and hence the "X" in XSPH). When first introduced, Monaghan (1989) used $\epsilon = 1$ (for the colliding gas cloud problem), while Monaghan (1994) used used $\epsilon = 0.5$ in simulations of free-surface, weakly-compressible flows. Benz (1990) reports reasons why $\epsilon = 0.1$ is a good choice, and notes that this term acts to reduce noise in particle velocities. Benz (1990) also notes that the kernel used to evaluate this term need not be the same as that used for other kernel estimates.

3.3 Consistency

The term consistency has several meanings in the literature, and these are discussed here, with an emphasis on their implications for SPH. Consistency is often used as meaning that the error in a solution goes to zero with h (Fries and Matthies, 2003). SPH is consistent if this definition is used, provided certain conditions are met, notably the constraint on how the number of points used in approximating integrals changes with h.

Consistency is also used to refer to the highest order polynomial that can be represented exactly (Fries and Matthies, 2003). So a method that can represent a quadratic exactly is second order consistent. Since most commonly used kernels (which are $O(h^2)$) can interpolate linear functions exactly, integral interpolants are first order consistent. But in determining the accuracy to be $O(h^2)$ it was assumed the integral of the kernel over its support is equal to zero, which it is, but not near boundaries. In fact, near boundaries, integral interpolants do not even achieve zeroth order consistency.

Further problems occur when integral interpolants are approximated by summation interpolants, indeed in using this approach determining the errors becomes difficult. Correction methods have been developed that ensure consistency requirements are met and these methods are discussed in the following sections. An SPH algorithm that has been corrected to ensure some order of consistency is achieved is a called Corrected SPH (CSPH), following the terminology of Bonet and Lok (1999).

3.3.1 Consistency and Error

Here the error in a summation interpolant is examined further. This error is obtained by approximating the integrals in the error for integral interpolant (Equation 3.30) using Monte Carlo estimates. Although these integrals can be found exactly, when Monte Carlo estimates are used they are not , and Equation 3.30 becomes (using the more conservative $1/sqrtNh^d$ for the error)

$$f_I(r) = f(r)\left(1 + \frac{C_1}{\sqrt{Nh^d}}\right) + C_2 f'(r)\frac{h}{\sqrt{Nh^d}} + \frac{f''(r)}{2!}\left(C_3 h^2 + \frac{C_4 h^2}{\sqrt{Nh^d}}\right)$$

in which the C_i are constants. If the kernel is chosen so that some order of consistency is enforced, then some of the constants are zero, and the non-zero constants determine the nature of the error. If first order consistency is enforced, then the constants $C_1 = C_2 = 0$ and the error is $O(h^2/\sqrt{Nh^d})$. If zeroth order consistency is achieved then only $C_1 = 0$ and the error is $O(h/\sqrt{Nh^d})$. Finally if no consistency is achieved then all of the constants are non-zero and the error is $O(1/\sqrt{Nh^d})$. In this discussion only the most significant contributions to the errors are kept.

The nature of the most significant error term determines how N should be scaled as h is changed. If first order consistency is achieved then ensuring that $N \propto h^{-d}$ will ensure the error

in the summation interpolant reduces like $O(h^2)$. If no consistency is achieved then ensuring that $N \propto h^{-4-d}$ means the error is $O(h^2)$. This underscores the importance of consistency to SPH. With no consistency SPH can still converge at a good rate, but it requires stupendous increases in particle numbers. The work (and hence the runtime) increases as the number of points used increases, and quickly become too large. Without consistency SPH is limited to providing entry level, low resolution simulations.

3.3.2 Consistency and Gradients

The implications of consistency are not so well covered in the literature for SPH models that employ the continuity equation and use the kernel gradient approach to find the gradients of properties. The same issue applies in this case though, since the kernel will not be normalised near boundaries.

However in some applications the gradient in the property is negligible near the boundary, and so the integral interpolant is negligible near the boundary. So the lack of consistency means that near the boundaries errors are small in spite of their low rate of convergence. This is similar to the reasoning used to justify neglecting integrals over boundaries (Benz, 1990). If Equations 3.64 and 3.68 are selected for the governing equations, and if the $\psi_a - \psi_b$ terms are zero near boundaries, the lack of consistency is not an issue.

In breaking waves significant gradients in the velocity are present, particularly at the free surface and possibly near the beach. For SPH models used to simulate breaking waves the lack of consistency is problematic.

3.3.3 Corrections for SPH

A number of improvements to SPH have been described in the literature (Bonet and Lok, 1999; Bonet and Kulasegaram, 2000). Kernels can be corrected to ensure consistency (kernel correction), and their gradients can be corrected (kernel gradient correction) in order that angular momentum conservation be satisfied (Bonet and Lok, 1999).

In addition, integral correction can be used to ensure the method passes the *patch test* (Bonet and Kulasegaram, 2000), and a method has been developed for avoiding *spurious modes* in SPH solutions (Bonet and Kulasegaram, 2000). In the following sections these techniques are discussed briefly.

3.3.4 Kernel Correction

In discretising Equation 3.30 (the Taylor expansion of an arbitrary function in terms of the moments) by approximating each of the moments by Monte-Carlo estimates, errors are generated in each term, errors that are $O\left(\frac{(\ln Nh^d)^{d-1}}{Nh^d}\right)$. A kernel may be fourth order consistent (except near the boundaries), but the summation estimates using this kernel are, in general, not even zeroth order consistent. This means that there is no advantage in using higher order (higher than second order) kernels, since the lack of consistency negates any increase in accuracy.

So there are two causes for the failure to achieve low order consistency in summation interpolants, the first cause is boundaries lying within the support of integral interpolants, and the second is errors introduced by using Monte-Carlo approximations.

To ensure consistency in SPH, Bonet and Lok (1999) developed kernel correction which involves modifying the kernel according to

$$\tilde{W}(\boldsymbol{r}, \boldsymbol{r}_a) = W(\boldsymbol{r}, \boldsymbol{r}_a)\beta(\boldsymbol{r})\left[1 + \boldsymbol{\gamma}(\boldsymbol{r}).(\boldsymbol{r} - \boldsymbol{r}_a)\right] \tag{3.73}$$

The correction parameters $\beta(\boldsymbol{r})$ and $\gamma(\boldsymbol{r})$ are determined by requiring that

$$\sum_{a=1}^{N} \frac{m_a}{\rho_a} \tilde{W}(\boldsymbol{r}, \boldsymbol{r}_a) = 1 \tag{3.74}$$

(essentially this is a normalisation condition) and that

$$\sum_{a=1}^{N} \frac{m_a}{\rho_a} (\boldsymbol{r} - \boldsymbol{r}_a) \tilde{W}(\boldsymbol{r}, \boldsymbol{r}_a) = 0 \tag{3.75}$$

(which is similar to the requirement that the first moment be zero). A kernel corrected by this approach is able to interpolate linear functions exactly, and so a method employing it is first order consistent.

Bonet and Kulasegaram (2000) suggest a simplification which involves setting $\gamma = \mathbf{0}$, in which case the kernel correction becomes

$$\tilde{W}(\boldsymbol{r}, \boldsymbol{r}_a) = \frac{W(\boldsymbol{r}, \boldsymbol{r}_a)}{\sum_{b=1}^{N} \frac{m_b}{\rho_b} W(\boldsymbol{r}_b, \boldsymbol{r}_a)} \tag{3.76}$$

and so the method is reduced to zeroth order consistent, however it is computationally quicker. This is also known as Shepard interpolation.

The problem with these methods for achieving consistency is that the kernel gradient is no longer antisymmetric (see Section 3.2.3). This means that the SPH equations no longer apply, since they require this property of kernels. The consequence of using these correction techniques with the standard SPH equations is that additional errors are introduced, and conservation of mass, energy and momentum is no longer enforced. It may be possible to show that the errors introduced by this approach converge at a sufficient rate that they can be neglected, but this has not yet been done. Alternatively the SPH equations for use with corrected estimates should be derived.

3.3.5 Kernel Gradient Correction

Kernel gradient correction is a method for improving accuracy in calculations of property gradients when Equation 3.41 is used. It is achieved by introducing a correction matrix, $\boldsymbol{L}_a$, that modifies the kernel gradient according to (Bonet and Lok, 1999):

$$\tilde{\nabla} W(\boldsymbol{r}_a, \boldsymbol{r}_b) = \boldsymbol{L}_a \nabla_a W(\boldsymbol{r}_a, \boldsymbol{r}_b). \tag{3.77}$$

The correction matrix is found by requiring that conservation of angular momentum be satisfied and ensures that the gradient of a linear function is evaluated exactly.

(Bonet and Lok, 1999) include discussion of *mixed* kernel and gradient correction where the kernel gradient correction is applied after the kernel has been corrected for consistency. The consequence of employing kernel gradient correction on the neglected surface integrals (see Equation 3.36) are not discussed in the literature.

3.3.6 Integral Correction

The patch test is sufficient to prove a method converges to the exact solution as $N \to \infty$ (Bonet and Kulasegaram, 2000). The patch test for a Poisson equation is the requirement that a potential which is linear with position is reproduced exactly, and reduces to (Bonet and Kulasegaram, 2000)

$$\sum_{b=1}^{N} \frac{m_b}{\rho_b} \nabla \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_b, h) = \sum_{B=1}^{N_B} A_B \boldsymbol{n}_B \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_B, h). \tag{3.78}$$

The sum on the right is for B, an index for boundary points going from 1 to N_B (the total number of boundary points lying within the support of the point at $\boldsymbol{r}_a$). Associated with boundary point B is the area A_B and the normal vector $\boldsymbol{n}_B$. Bonet and Kulasegaram (2000) note that this expression is the discretised form of Green's theorem

$$\int_D \nabla \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}', h) d\boldsymbol{r}' = \int_{\partial D} \boldsymbol{n} \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}', h) d\boldsymbol{r}'. \tag{3.79}$$

While requiring that the patch test be passed is equivalent to requiring that the discretised form of Green's theorem be met for a Poisson equation, it has not been shown that this is always the case: it has not been shown that the patch test for the conservation equations used here reduces to a discretised form of Green's theorem. Bonet and Kulasegaram (2000) use the Green's theorem condition in their study of several metal-forming process which are not governed by Poisson's equation. The issue is not whether Green's theorem should apply (it should) but whether it is always equivalent to a patch test (which has not been proven).

To satisfy Equation 3.78, Bonet and Kulasegaram (2000) introduce integral correction by

$$\tilde{\nabla} \phi_a = \nabla \phi_a + \gamma_a \left(\phi_a - \sum_{b=1}^{N} \frac{m_b}{\rho_b} \phi_b \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_b, h) \right) \tag{3.80}$$

in which γ_a is the integral correction vector at $\boldsymbol{r}_a$, and when this expression is substituted into Equation 3.78 it becomes

$$\gamma_a - \sum_{b=1}^{N} \frac{m_b}{\rho_b} \gamma_b \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_b, h) = \sum_{B=1}^{N_B} A_B \boldsymbol{n}_B \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_B, h) - \sum_{b=1}^{N} \frac{m_b}{\rho_b} \nabla \tilde{W}(\boldsymbol{r}_a - \boldsymbol{r}_b, h) \tag{3.81}$$

which is a set of N equations in N variables. This can be solved directly or by an iterative method (Bonet and Kulasegaram, 2000).

3.3.7 Spurious Modes

Bonet and Kulasegaram (2000) note that spurious modes can be generated in SPH, saw tooth-like features that are a numerical artefact, that cannot be damped by use of artificial viscosities. Bonet and Kulasegaram (2000) describe a method for removing these modes. These modes are not present in all SPH simulations (Bonet and Kulasegaram, 2000).

3.3.8 Boundaries in SPH

The application of SPH to weakly compressible flows with free surfaces requires a means of defining boundaries to contain the fluid, and several techniques have been reported in the literature. The methods have various merits and demerits and are outlined here.

Ghost Point Boundaries in SPH

The earliest method reported in the literature (Monaghan, 1994) employs ghost points that may or may not be stationary, depending on the nature of the boundary (Monaghan created waves by using a piston-type wave maker constructed from moving ghost particles). These ghost points exert a mass-averaged force derived from a Lennard-Jones type potential (though any potential could be used) on free particles, with the total force acting on the free particle being the sum of all contributions from all ghost points.

The form of the potential (Φ) is usually chosen so that it is zero at a finite distance, reducing computational requirements. So the force $\boldsymbol{f}_{bound}$ acting on a free particle due to one ghost point a distance r away is

$$\boldsymbol{f}_{bound} = \nabla\Phi \tag{3.82}$$

where Monaghan (1994) used

$$\Phi = D\left[\left(\frac{r_0}{r}\right)^{p1} - \left(\frac{r_0}{r}\right)^{p2}\right] \tag{3.83}$$

D being a magnitude for the force, r_0 being the distance beyond which the force is zero and $p1$ and $p2$ being arbitrary exponents, subject to the condition $p1 > p2$. Monaghan (1994) used D between gH and $10gH$ for simulations of flows, where H is the depth. The aim is to provide a force large enough that free particles are constrained, but not so large that the boundary is too hard. Monaghan (1994) used r_0 equal to the initial particle spacing and choose $p1$ and $p2$ to be 4 and 2 respectively, noting that other choices did not seem to have significant effects, though they should be integers since reals would be more expensive to compute.

Providing a repulsive force is the sole task of the ghost points, and they do not interact with the free particles in any other way. The exception to this is when no-slip conditions are required, when boundary terms will appear in the momentum equation (Monaghan, 1994),

A notable problem with potential boundaries like this is the generation of corrugations (Monaghan and Kos, 1999). For the example of a linear boundary, the curve of equi-potential is not a straight line running parallel to the boundary, instead it has dips halfway between each pair of adjacent boundary points. This corrugation could be reduced by increasing the number of ghost particles. The merit in these boundaries is that the boundary potential energy is readily computed, and that boundaries can be created with any shape very easily.

Number Density Boundaries

In the SPH model of Dalrymple and Knio (2001) (and also in the MPS model of Koshizuka *et al.* (1995)) boundaries are created from stationary fluid particles. They have non-zero mass and are accounted for when using the summation interpolant for the density (Equation 3.56). This means the fluid density does not decrease as the number densities of free particles decrease near solid boundaries. To ensure this effect is avoided Koshizuka *et al.* (1995) use two or more rows of boundary particles, depending on the smoothing length they have chosen.

In comparison to potential boundaries this approach will require a few more computations because there are more boundary points. In SPH with this type of boundary the densities of the boundary particles will need to be updated also, or else non-conservation of energy will occur. On the other hand no special treatment is required of boundary points in source code, other than setting velocities to zero. Because there are no potentials involved, the boundary potential energy does not make a contribution to the total energy.

However this type of boundary may still experience corrugations, but rather than a corrugated potential, curves of equal density are corrugated.

Normal/Tangential Boundaries

To avoid the corrugated boundaries, Monaghan and Kos (1999) developed boundaries that, in this thesis, are referred to as normal/tangential boundaries. This approach is a variation on the ghost point boundaries but has the demerit that the boundary potential is not defined (at least it is not in Monaghan's implementation).

Normal/tangential boundaries are created by specifying, at each boundary point, an inward pointing normal vector ($\boldsymbol{n}$). Equations 3.84 to 3.87 are from Monaghan (1994). The force

provided by the boundary particle is given by

$$\boldsymbol{f}_{bound} = \boldsymbol{n}R(y)P(x) \tag{3.84}$$

where x and y are the distance from the boundary point to the free particle in the tangential and normal directions respectively. Δp is the particle spacing and the function $R(y)$ and $P(x)$ are given by

$$R(y) = \begin{cases} A\left(\left(\frac{y}{2\Delta p}\right)^{-\frac{1}{2}} - \left(\frac{y}{2\Delta p}\right)^{\frac{1}{2}}\right); & y \leq 2\Delta p \\ 0 \qquad\qquad\qquad\qquad\quad ; & y > 2\Delta p \end{cases} \tag{3.85}$$

and

$$P(x) = \begin{cases} \frac{1}{2}\left(1 + \cos\left(2\pi\frac{x}{2\Delta p}\right)\right); & x \leq \Delta p \\ 0 \qquad\qquad\qquad\quad ; & x > \Delta p \end{cases} \tag{3.86}$$

respectively. Finally the parameter A is given by

$$A = \frac{1}{h}(0.01c^2 + \beta c\boldsymbol{v}_{ab}.\boldsymbol{n}_b) \tag{3.87}$$

in which

$$\beta = \begin{cases} 1; & \text{if the particle is approaching the boundary} \\ 0; & \text{if the particle is moving away from the boundary} \end{cases} \tag{3.88}$$

The term with β in Equation 3.87 helps to damp out motion of free particles normal to the boundary.

While Monaghan (2003) asserts that the boundary potential energy is not large and can be ignored, the author considers it a useful tool in debugging simulations, since it can be used to identify free particles that are too close to the boundary. Determining the boundary potential energy requires finding a potential (Φ) such $\boldsymbol{f}_{bound} = \nabla\Phi$. The expression for A means this potential does not exist.

3.4 Additional Features of SPH Models

As more research is being conducted into SPH, the list of physical effects and numerical features that have been accounted for is growing, some of which are outlined here.

3.4.1 Turbulence

A laminar flow is characterised by regular, steady streamlines (Oran and Boris, 1987). In a turbulent flow stream lines are time dependent, featuring eddies that may grow or dissipate in time (Oran and Boris, 1987). These eddies are 3 dimensional features (Ferziger and Perić, 1996) and a numerical simulation must be three dimensional to accurately account for this effect.

In SPH the smallest eddy that can be simulated consists of two particles rotating about a centre of mass. The dissipation of turbulent kinetic energy for this smallest eddy is accounted for by the inclusion of viscosity. But at this level the rates at which turbulent kinetic energy is dissipated will exceed the rates measured in real systems because in general the viscosity is chosen much larger than measured values to ensure noise is smoothed. So too much importance should not be attributed to turbulence in SPH simulations at this scale.

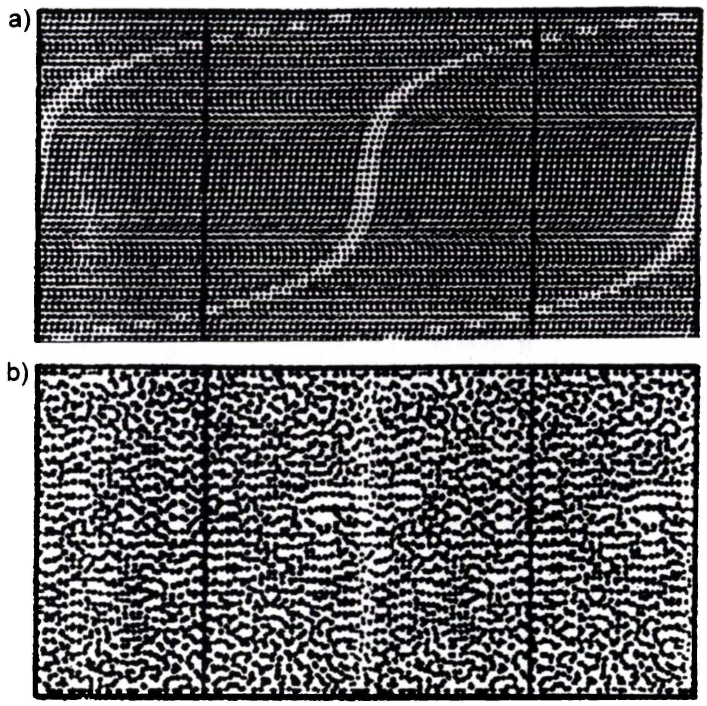

Figure 3.6: *SPH simulations of Couette flow from Cleary and Monaghan (1993). The vertical black lines represent the periodic boundaries. The lighter* marker *particles were initially a column of at the centre of the plot, and demonstrate the distortion of the fluid column in both snapshots. Both snapshots are at time 1.6 s,* (a) *showing a laminar flow and* (b) *showing turbulent flow.*

Transition to Turbulence

Cleary and Monaghan (1993) performed simulations of Couette and Poiseuille flows using an SPH model, simple flows that are used extensively in the study of turbulence. The Poiseuille flow simulations were two dimensional and consisted of fluid (specified with an initial velocity) between two parallel stationary walls. Periodic boundary conditions were employed to achieve the infinite wall lengths required. The initial velocity in part determines the Reynolds number for the flow, along with the fluid density and viscosity (Cleary and Monaghan (1993) used the viscosity term described in Section 3.2.3). A Couette flow is similar to a Poiseuille flow but the walls are moved with equal speeds in opposite directions and the fluid is initially at rest. Examples from the Couette simulation are shown in Figure 3.6.

Cleary and Monaghan (1993) observed very different flows for high Reynolds numbers than for low Reynolds numbers, and described the different flows as being either laminar or turbulent. Cleary and Monaghan (1993) assert that these results mean SPH can simulate the transition of a flow from the laminar to the turbulent regime. They also note that while turbulence is a three dimensional phenomenon, the transition to turbulence is two dimensional.

Turbulence and Chaos

Turbulence is a chaotic phenomenon (McComb, 1990), meaning that the system is in a significantly different state if the initial conditions are changed slightly. But it should be kept in mind that a "significant difference" may not be important in a particular simulation. For example, in simulating a propagating wave (reasonably large) in shallow water a turbulent boundary layer may be present at the sea floor (Isobe, 1988), and while this boundary layer is chaotic, it is not

important for the propagation of the wave.

A single simulation of a chaotic system provides just one possible path that the state of the system may take and this may not be useful. It is most likely interest lies in average or typical conditions, and to this end, one of two approaches can be employed when dealing with chaotic systems. The first is to time average quantities over timescales longer than the turbulence, and so the quantity has turbulent and turbulence-averaged contributions, a process called Reynolds averaging (McComb, 1990). This is widely used in Eulerian models, but is not applicable to Lagrangian models.

The second approach is to perform an ensemble average. This involves creating an *ensemble* of *many* similar simulations, differing in their initial conditions only by small amounts, and then averaging over the ensemble (Reif, 1985). The ensemble average is the only option for obtaining average or typical conditions in particle-based Lagrangian systems and it is a computationally expensive one.

Tensile Instability

When a fluid in an SPH simulation experiences tensile forces which occur when attractive forces like surface tension or molecular attraction are included, instabilities occur (Swegle *et al.*, 1995; Morris, 2000). This instability acts to cause the unphysical clumping of particles. For fluids without attractive forces this is not an issue.

Bonet and Kulasegaram (2001) and Rabczuk and Xiao (2004) note that most SPH models employ *Eulerian* kernels, as opposed to *Lagrangian* kernels. Models using Lagrangian kernel do not experience the tensile instability, but are limited in the distortions that can be simulated (Rabczuk and Xiao, 2004). Rabczuk and Xiao (2004) state that Lagrangian kernels are better for solids and Eulerian kernels are better for liquids, because of the distortions typical to each state.

3.4.2 Multiple Phases and Surface Tension in SPH

SPH has been used to simulate systems containing more than one type of fluid, exemplified by Monaghan *et al.* (1999) who simulated a gravity current in a stratified flow using fluids of three different densities. The emphasis here is breaking waves in which the presence of air is of secondary importance, but it is clear that one could, if one wished, account for it in an SPH model.

In SPH free surfaces are not defined as sharp interfaces, rather as being somewhere within a range of places. However to account for surface tension, the position of the free surface must be known. There are however, steps that can be taken to address this weakness. In an SPH model the surface can be tracked by creating marker particles that are convected with the fluid (Morris, 2000). These particles can be placed either throughout the entire domain, or just on the free surface.

Accounting for surface tension has been achieved relatively recently in the work of Morris (2000), for example. Using the marker particles described in the previous paragraph, the surface curvature can be calculated, from whence the surface tension acting on free particles near the surface can be found. The inclusion of surface tension in an SPH model will allow for improved simulation of fragmentation and coalescence in multi-phase systems.

3.4.3 Variable Smoothing Length

It is sometimes necessary to simulate one or more regions with greater accuracy, either because the solution is desired with a higher resolution (but it might be too expensive to conduct the entire simulation with that resolution), or because critical regions exist in which errors would be too

large without the higher resolution (Benz, 1990). In Eulerian models, locally high resolutions can be achieved by nested sub-grids, while in SPH this can be achieved by allowing variable smoothing lengths Benz (1990). The SPH equations need to be redefined if this course is chosen, in particular, the meaning of a kernel estimate requires clarification when smoothing lengths vary between particles.

Benz (1990) notes that depending on the choice of kernel, and particularly for those with compact supports like Monaghan's cubic spline, if the smoothing length is too small then only the particle itself contributes to the summation interpolants for its properties. Also, the error in approximating the integral interpolant by a summation interpolant (Equation 3.25), will increase if N is not decreased faster than h^d. To this end Benz (1990) suggests relating the smoothing length to the density (he uses a summation interpolant for the density in his simulations which are mostly astrophysical).

3.4.4 The Courant-Friedrichs-Lewy Condition

In the previous sections the derivation of discretised equations from the governing equations has been discussed, and if everything goes according to plan, the solution to the discretised equation should *converge* to the exact solution obtained from the continuous governing equations. Here converge means the solutions are the same (for the continuous and discretised equations) as the smoothing length and time-step reduce to zero. However for some choices of smoothing length and time-step the discretised equations *diverge*, are unstable and the solution does not converge.

It turns out that the time-step is not independent of the smoothing length or velocities (such as fluid velocities and the speed of sound), rather they are related by the Courant-Friedrichs-Lewy (CFL) condition (Ferziger and Perić, 1996). For SPH the CFL condition is (Lattanzio *et al.*, 1986)

$$\tau^{cfl} = 0.3\min(\tau_1^{cfl}, \tau_2^{cfl}) \tag{3.89}$$

(although other references use values slightly larger or smaller than 0.3). Here

$$\tau_1^{cfl} = \min_b \left(\frac{h_b}{\boldsymbol{F}_b} \right)^{\frac{1}{2}} \tag{3.90}$$

and (Monaghan, 1992)

$$\tau_2^{cfl} = \min_b \left(\frac{h_b}{c_b\,(1 + 0.6\alpha)} \right) \tag{3.91}$$

in which viscosity is included. In these equations particle b has smoothing length h_b, mass averaged body forces $\boldsymbol{F}_b$ and velocity of sound c_b. The notation $\min_b$ means the minimum value of the set containing the argument evaluated for each b. Different versions of Equation 3.90 occur in the literature but this expression is the only one with consistent units.

3.4.5 Incompressible SPH

All fluids are compressible to some degree, but because it often simplifies a problem, these compressible fluids are often approximated by incompressible fluids when numerical or analytic solutions are sought. As mentioned earlier, SPH approximates the fluid by one that is more compressible, but an alternative option is to use the incompressible approximation. The are several SPH variants that do this: PSPH (Cummins and Rudman, 1999) (Projection method for SPH), and MPS (Koshizuka *et al.*, 1995, for example).

In this work the term MPS has been used rather loosely to describe two related but distinct approaches for requiring incompressibility. The first approach is termed *a particle method* (Koshizuka *et al.*, 1995) and employs a penalty method in which the pressure field is modified

when the number density of particles differs from a reference level. By contrast MPS (Koshizuka *et al.*, 1998) solves a Poisson equation for the pressure. PSPH also solves a Poisson equation for the pressure (Cummins and Rudman, 1999) but employs a different method to do so (it uses a projection operator).

Meritorious in PSPH and the two MPS methods is that the speed of sound does not restrict the CFL condition for the time-step as it does in SPH. On the negative side many more calculations are required every time-step. Cummins and Rudman (1999) note that the accuracy of the MPS methods in comparison to SPH has not been established, and that while PSPH shows the potential to simulate incompressible free-surface flows more accurately than SPH, more research is required to achieve this potential.

Both approaches (SPH and the incompressible versions) are approximations to reality, the real question being which is more accurate for the same amount of computational effort, a question that as yet, remains unanswered in the literature. The rates of convergence to the exact solution for the different methods need to be compared also.

3.4.6 A Summary of Errors and Simulation run-times

In this section a summary of the errors occurring in, and run-times required for, SPH simulations are presented. The emphasis is on discretisation errors, since modelling errors are very much case specific. Irrespective of how errors are generated, they will propagate through the fluid via smoothing, the very means by which the method works. So in order to avoid the accumulation of errors to the point where a simulation is valueless, errors must be identified and reduced. All steps that can be taken to improve accuracy result in simulations requiring longer to run to completion. These steps are discussed along with other factors influencing run-times.

Time-Stepping Errors

The accuracy of time-stepping is a very much dependent on the order of the method employed. Monaghan (1994) employed a two-stage predictor corrector method for which in every step the predictor stage was completed using the function value from the previous step. Both the two-stage predictor corrector and the time saving modification have an error $O(\tau^3)$ (Monaghan, 1994), so as $\tau \to 0$ the error reduces proportionally to τ^3. By repeating the corrector step until a desired tolerance is achieved, the time-stepping error is a tolerance error (Lambert, 1973). Such a step clearly comes at a much greater computational cost, meaning that simulation run-times are longer.

Errors from the Kernel and Summation Interpolant

Most of the commonly employed kernels are $O(h^2)$, but methods exist (Monaghan and Lattanzio, 1985) for generating kernels of $O(h^4)$ or higher, (for example see the super-Gaussian kernels of Monaghan (1982)). Higher order kernels are negative in some parts of their domains and this can cause problems in some simulations.

The error in approximating the integral interpolant by the summation interpolant is given by the ratio $O\left(\left(\ln(Nh^d)\right)^{d-1}/Nh^d\right)$. Requiring that this error is the same order as the kernel estimate places constraints on how the smoothing length is changed when the number of particles is changed. Any increase in the number of particles means an increase in the number of computations required. Additionally, the run-times for some algorithms increase with the number of neighbours.

Steps can be taken to ensure higher order consistency is achieved; if they are not taken, the summation interpolant is not even zeroth order consistent. Typically one might modify kernel approximations to achieve first order consistency. Achieving second order consistency is

computationally expensive. The literature does not include estimates for the order of the error due to the lack of consistency, but it is quite likely to dominate over the errors discussed in the previous two paragraphs.

Errors in Property Gradients

In evaluating the gradient of a property in SPH, surface integrals are generated that are neglected, hence introducing an error. For kernels with compact support these errors are exactly zero for particles which have no boundaries within their supports, and non-zero otherwise. The order of these errors is not clear. In MPS models gradients are estimated from smoothed difference approximations. Estimates for these errors are likely to be similar to the SPH approach within the body of the fluid, but are significant for particles that have boundaries within their supports. So irrespective of the method used to find gradients, errors are generated near boundaries.

Density Variations and the Speed of Sound

The use of an equation of state to determine the pressure means the density can change. The amount by which it is allowed to change is determined by the speed of sound, which is typically chosen so that the variations are less than 1%. The choice of the speed of sound directly affects the maximum allowable step-size, via the Courant condition, but also increases errors in density variations.

Artificial Clustering

In using the SPH method for finding the gradients of properties, artificial clustering may be generated if flat-topped kernels are employed. The magnitude of this error is not known but it can be significant, and steps can be taken to avoid the effect.

Tensile Instability

Fluids with attractive forces can experience instability due to the presence of tensile forces, an instability that appears very similar to artificial clustering. Steps can be taken to avoid these effects (Morris, 2000) but this is still a current topic for research.

Errors in the Initial State

Incorrect particle properties can lead to significant amounts of spurious energy being present in the initial state of the system, energies that can be removed by simple smoothing algorithms. This smoothing can be time consuming, especially if a particularly smooth initial state is desired.

Rounding Errors

An unavoidable consequence of discretisation is that solutions are subjected to rounding errors which accumulate with time. Rounding errors can be avoided by reducing any or all of the total number of operations, time-steps or particles. Another option would be to use higher precision numbers, that is double or quad precision numbers, but this greatly increases run-times. The problem is that steps taken to reduce other sources of errors, inevitably involving higher numbers of computations, result in increased rounding errors. So optimal accuracies exist that can only be surpassed by increasing precision.

Choice of Domain

The choice of domain, both temporal and spatial, for a simulation affects run-times, and can possibly cause errors also. For a constant resolution, larger domains (both temporal and spatial) cause longer run-times. So choosing as small a domain as possible is desirable, but there comes a point where steps taken to reduce the size of the domain impact on how well the simulation reflects reality.

Bugs

Programming errors, called bugs, also generate avoidable errors. The only method to avoid these is care and attention to detail when the source is written. But almost every code will have bugs and these need to be fixed. Extensive checking of output is required to determine their presence.

Algorithms

Improving algorithms is a significant means by which error generation and run-times can be reduced. The algorithms employed to do a particular task can generate avoidable errors such as rounding and subtractive cancellation errors. The use of alternative algorithms can sidestep the generation of avoidable errors.

There are often many algorithms available for solving a particular problem but while output from these may appear similar, there may be significant variations in the error generation and in the time required to complete the job. Finding better algorithms is always a path for programmers wishing to improve their programmes but the deficiencies of an algorithm can be very subtle.

In SPH (and some of the other Lagrangian methods) finding the neighbours of a particle is a significant task not required by Eulerian grids in which neighbours are always known. The simplest algorithm involves finding the distance from the particle to every other particle, then those within a certain distance are the neighbours. If there are N particles then finding neighbours for all particles using this algorithm requires $O(N^2)$ calculations, a load that can easily dominate total run-times.

Hardware

An obvious step for decreasing run-times is to use parallel computers, either multiple CPU machines or Beowulf clusters. It should be noted that using parallel computers does not reduce the total number of computations required to complete a simulation, this stays the same, rather the job is completed faster because many hands make light work. In fact the additional book-keeping and time spent moving information between CPUs or computers increases the calculations required. Exemplary are fast Fourier transforms, which can take longer to complete on four CPUs than on one.

Nonetheless using parallel computers is a necessity if high resolutions are to be attained in SPH simulations, and consequently code must be parallelised. This is the process whereby source code is rewritten so it can be compiled and run successfully on more than one CPU. Such codes are common in astrophysics but less so in hydrodynamics.

Manners and the Maximum Runtime

In spite of the reducing cost and increasing power of computers, most computing facilities are shared and this has the consequence that modellers need to behave in a considerate manner. Massive simulations that gobble system resources and take a long time to complete may be out of the question. SPH modellers are no different from other users and need to be considerate, so

their simulations need to be constructed with this in mind. Also significant is that long run-times require the computer to be running for a long time, something that may not be possible given maintenance requirements of the computer and stability of the operating system.

Consequently for most users there is a ceiling (somewhat grey in colour) on the maximum simulation runtime. This determines maximum achievable resolutions and in turn the magnitude of errors. This point is the main limiting factor in the usefulness of SPH for modelling hydrodynamics (a factor that affects VOF modellers too).

3.5 Conclusions

SPH was selected for the basis of the model to be created here from the two particle-based Lagrangian methods that have already been applied to fluid flow. Because SPH has been widely applied to nearly incompressible fluids it has a significant body of literature associated with it. The algorithm the method employs is simple in comparison to other methods meaning an SPH code is easier to create. While SPH is simple to utilise and is well supported by literature, it has significant demerits associated with it. These demerits have been examined, and where they exist, techniques of avoiding them have been described.

Nomenclature

c	The velocity of sound	$\mathrm{m{\cdot}s^{-1}}$
C	A constant that normalises kernels	variable
d	The dimensionality of the domain, being 1, 2 or 3	
f	A general function	
$\boldsymbol{f}_{bound}$	The force applied by a boundary point	$\mathrm{m{\cdot}s^{-2}}$
g	A general function	
g, $\boldsymbol{g}$	The acceleration due to gravity	$\mathrm{m{\cdot}s^{-2}}$
h	A smoothing length	m
m	Mass	kg
n	The number density	points $\cdot\mathrm{m}^{-d}$
$\boldsymbol{n}$	Inward-pointing unit-vector normal to the boundary	m
N	Total number of particles, equations or terms	
P	The gauge pressure	Pa
r	The separation of two points	m
$\boldsymbol{r}$	The position vector	m
t	Time	s
u	A dummy variable	
$\boldsymbol{v}$	The velocity vector	$\mathrm{m{\cdot}s^{-1}}$
W	A non-specific kernel function	m^{-d}

Greek Symbols

α	A coefficient in the viscosity term	
ψ	A general property	
τ	The length of the time-step	s
ρ	The density	$\mathrm{kg{\cdot}m^{-3}}$

Operators and Relations

$\equiv$	Definitively equal to
$\nabla\psi$	The gradient operator
$\nabla_r\psi$	The gradient operator taken with respect to coordinates r

Superscripts

$\boldsymbol{r}'$	A dummy variable in integration
f', f'' and $f^{(N)}$	The first, second and Nth derivatives of a general function
$\overline{c_{ab}}$, $\overline{\rho_{ab}}$	The arithmetic averages of c and ρ respectively
$\boldsymbol{r}_{ab}$, $\boldsymbol{v}_{ab}$	The difference $\boldsymbol{r}_a - \boldsymbol{r}_b$ (similar for $\boldsymbol{v}_{ab}$)

Subscripts and Integral Limits

ψ_a, ψ_b	The values of property ψ for particles a and b respectively
ψ_I	The integral interpolant for ψ
ψ_S	The summation interpolant for ψ
$f_{I\nabla}$	The gradient integral interpolant for ψ
$\int_D$	An integral performed over the entire d-dimensional domain

Chapter 4

Model Verification

Recollect that the aim of this work is to create a particle-based Lagrangian model of a breaking wave and establishing its suitability for the task. In the preceding chapter the selection of Smoothed Particle Hydrodynamics as a suitable method was assessed, and here the details of the implementation of the method are described. The model created here has been named MARIAN.

In creating the model a number of options were available that determine the behaviour of MARIAN. For the most part MARIAN was created to be as similar as possible to the models of Monaghan (1994) and of Dalrymple and Knio (2001), so that results from a common verification exercise could be compared and any differences should be due to programming errors.

Two tests are performed here: the evolution of an elliptic drop and a bursting dam. These tests were first performed by an SPH model by Monaghan (1994) and subsequently by Bonet and Lok (1999). The elliptic drop tests compares an initially circular distribution of particles (velocity specifications are described in the following section) with the analytic solution. The bursting dam simulations are a comparison to experimental results which were published by Martin and Moyce (1952). In addition the effect of particle stacking is demonstrated. Finally the behaviour of MARIAN is examined in a number of simulations of a seiching basin.

4.1 Model Details

In this section the details of MARIAN are presented. The reasons for choices made in its construction will also be discussed. These discussions are made with reference to the literature reviews contained in the previous chapters. The model was written using FORTRAN 95, and compiled using Lahey-Fujitsu compilers on both Windows and Linux platforms. Output from the model was written to binary data files which are post-processed using purpose written MATLAB programs.

Governing Equations

In MARIAN, the density is generated by using the continuity equation (Equation 3.64), rather than from the summation interpolant for the mass (Equation 3.56). When this work was started this choice was made in order that results from this model would compare well to those from the models of Monaghan (1994) and Dalrymple and Knio (2001). The same reasons Monaghan (1994) used for making this choice also applied here (the summation interpolant offering poor performance due to the lack of consistency, especially near boundaries).

Equation of State

The equation of state used here (Equation 3.50) is the same as that used by other modellers Monaghan (1994); Bonet and Lok (1999); Dalrymple and Knio (2001), in their models for nearly incompressible fluids. Improving the equation of state is unlikely to improve the accuracy of results since these are dominated by other sources of error.

Time-Stepping Method

The same time-stepping technique used by Monaghan (1994) and by Dalrymple and Knio (2001) was selected here. Again, this was to ensure that results would be as similar as possible. The method used is the two-stage predictor corrector with the time saving variation discussed by Monaghan (1994), whereby the functional evaluation in the predictor stage is replace by the functional evaluation for the corrector stage from the previous step. While this method is not very accurate (in fact errors due to this time-stepping technique may dominate a simulation), it is fast.

Program Structure

The main program unit of MARIAN contains a *while* loop, which controls the stepping through time. Within the time-stepping loop is a *do* loop that counts through the particles, identifying neighbours and calculating the values of the governing equations. This *do* loop cannot be removed by vectorisation without changing the neighbour-finding algorithm. Following the *do* loop there is a little book-keeping, involving updating particle properties, writing data to disk and ensuring the time-step is short enough, so that stability conditions are met.

Finding Neighbours

In MARIAN neighbours are found by a simple vectorised approach. For a given particle the square of the distance to every other particle is found (it would be unnecessarily time consuming to calculate the distance), then a vector is created with a place for every other particle. If the other particle is a neighbour then the vector holds the unique identifying number for that particle, otherwise it holds a zero. For example for particle 5 this vector may look like $[0\,2\,3\,0\,5\,6\,0\,8\,9\,10\,0\ldots]$. The *pack* command is used to extract non-zero elements to a smaller vector.

The method for generating the list of neighbours that is used here is fast because it is highly vectorised, involving only one DO loop. Further vectorisation is not possible with this algorithm (at least it is not in FORTRAN) because there is no two-dimensional version of the *pack* command. It can be seen that with this algorithm for each pair of particles in the system (if there are N particles there are N^2 pairs), the square of the distance is calculated twice in every stage. That is, for example, the square of the distance is calculated once for particle a (for which one neighbour is b), and once for particle b (for which one neighbour is a).

Several authors (Monaghan, 1994, for example) employ linked-lists to improve the speed at which the neighbours for particles are found. This involves creating a grid with cells $2h$ wide (if the kernel employs Monaghan's convention where the support is a circle $2h$ in diameter) and then all potential neighbours of a given particle lie within the same cell or in the eight cells surrounding it. Consequently linked-lists reduce the time required to find neighbours since the distances between the majority of pairs are not evaluated as they are not needed. However linked-lists require more book-keeping than the simple algorithm used here.

A linked-list was implemented in MARIAN, but for particle numbers up to 10 000 run-times were *longer*. This is most likely to be because the cost introduced by the requisite book-keeping is greater than the reduction in cost due to the linked-list. Quite possible is that the algorithm employed in implementing the linked-list was less than ideal. It is also possible that the high cost

of the linked-list will reduce for higher numbers of particles, but this was not investigated here. Because the performance of MARIAN was not improved by the use of linked-lists, this feature was removed.

Energies in MARIAN

There are several forms of energy in systems MARIAN can simulate, and in this section, these are introduced and discussed. Issues regarding their definition are discussed and the decisions made here described.

Each free particle has a mass (m_i) and a velocity (v_i) and the kinetic energy of each particle is given simply by

$$Ek_i = \frac{1}{2} m_i v_i^2. \tag{4.1}$$

The net kinetic energy is the sum of the kinetic energies for each particle. The result is a vector with the net kinetic energy for each time-step.

Since MARIAN has a constant acceleration due to gravity (g), and each particle has a height (h_i), the gravitational potential energy is set equal to

$$Eg_i = m_i g h_i \tag{4.2}$$

from which the net gravitational potential energy for the system can be found from the sum of all values.The reference level is arbitrary may be chosen so as to simplify the problem. While running, MARIAN takes this reference value to be zero (all heights are with respect to the origin), but this means that when the output is processed, gravitational potential energies are often very large and any detail in the energies is drowned out by the large bias. The solution used here, applied during post-processing, is to subtract the minimum value of the net gravitational potential from all values. The resulting minimum value is zero, the bias is removed, and the detail in plots of energy is visible.

Since each particle has a variable density, work must be done to change the volume (and hence the pressure and density). This work is called compressive energy in this thesis. It can be evaluated in two ways, the first being to use the result from thermodynamics (Carrington, 1994)

$$W_{12} = -\int_{V_1}^{V_2} P dV \tag{4.3}$$

where W_{12} is the work required to change the volume of the particle from V_1 to V_2. The initial volume V_1 can be chosen arbitrarily to be a reference volume V_{ref}, corresponding to a reference density. By this approach, given the density of a particle and the equation of state, the work to compress or expand a particle from a given reference state can be found. This approach requires an integrable equation of state. The second method for calculating the energy comes from discretising the conservation of energy equation using the standard SPH techniques.

If viscosity is employed energy is *lost* from the system. In reality it would become thermal energy, heating the fluid. But temperature is not accounted for here, so to ensure energy conservation, viscous dissipation must be accounted for. The discretised conservation of energy (which accounts for both viscosity and compressive energies) is given by (Monaghan, 1994)

$$\frac{du_a}{dt} = \frac{1}{2} \sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \Pi_{ab} \right) v_{ab} \nabla_a W_{ab} \tag{4.4}$$

in which the notation is the same as in the previous chapter.

The final form of energy is boundary potential energy. If ghost point boundaries are used with a Lennard-Jones type potential, then the boundary potential energy for a particle is just

the value of the potential (Φ) less a reference value. It is suitable to take the reference value to be zero, so a particle far from the boundary has zero boundary potential energy. If the boundary force is non-integrable then the potential does not exist, and the boundary potential energy is not defined. This is the case for the normal/tangential boundary method.

The sum of all energies for all particles in a single time-step is referred to here as the net total energy. If a system is closed, that is no energy, mass or momentum crosses the boundaries, then the net total energy should remain constant, by the law of conservation of energy. The use of a wave maker means the net total energy of the system will increase.

Removing Spurious Energy

Spurious energy may be present in the initial state of a simulation. This will occur if particles are inadvertently given wrong heights, incorrect densities, or are placed too close to boundaries. This energy can be removed by allowing the system to evolve, during which spurious energy is transformed to kinetic energy which can be damped by a number of means. This approach does not work in all cases. The bursting dam simulation presented later in this chapter is an example.

MARIAN contains a subroutine called VMOSS, that dampens spurious energy. The name is an acronym for Vertical Movement Only Settling Subroutine, which gives hints as to how it works. In order to allow it to be used for cases like the bursting dam, the system is allowed to progress but only vertical velocities are modified, so horizontal velocities are set equal to zero every time-step. In addition, every N time-steps the vertical velocities are set equal to zero, hence removing energy from the system.

Typically, VMOSS takes as long, or longer, to create equilibrium states from which simulations can be started, as the simulation itself. While this settling subroutine can be used to remove spurious energy from systems where the water body is not meant to be in equilibrium, it cannot be used where the initial state has velocities specified, since these will get damped. The operation of VMOSS needs to be monitored carefully because there is a danger that the system will progress to an equilibrium state, a crystallised state (introduced on Page 99), meaning VMOSS is allowing spurious transport of energy (Lombardi *et al.*, 1999).

The systems state is output to file during VMOSS operation if the user desires. These VMOSS output files are the same as normal output files and are post-processed using the same MATLAB programs.

Control of Step-Size

At setup time the user can specify the preferred step-size MARIAN will use, τ_{pref}. But it may happen that the system will require a shorter time-step in order to ensure that the Courant condition be satisfied for stability. To this end, the Courant time-step (τ_{cfl}) is evaluated every time-step, and if τ_{pref} is larger (than τ_{cfl}), it is split into small steps by finding the smallest positive integer N such that $\tau_{pref}/2^N < \tau_{cfl}$ and then the time-step is $\tau_{=}\tau_{pref}/2^N$. Normally two half-sized steps are sufficient, and the system will progress till the τ_{pref} is less than τ_{cfl} again.

Model Input

Each simulation with MARIAN requires a .dat file as input. This is simply a file containing ASCII text, in which specific positions in the file contain information regarding the parameters that control operation of the model.

The initial state of the system is specified by one of two methods, the first being to specify the names of text files which contain lists of initial positions, velocities and pressures for particles and boundaries. The second option is to specify the name of a binary output file, a file created by MARIAN listing variables for the entire system. It is possible to start from almost any output

file, and from almost any of the time-steps listed in the output file. This means MARIAN can perform a *hot start*. This is very useful meaning loss of power or computer crashes do not mean the end of a simulation. It also means a single file (perhaps a very smooth VMOSS output file) can be used to start a large number of simulations.

Analysis of Output

Because short time-steps must be used to ensure stability, it is not always possible for data to be written to disk every time-step. To attempt to do so could mean output files too big to be processed. Consequently MARIAN was created so that the user could specify how frequently data is written. For example, one might choose to have data from every 100th step written to file, but this may change, depending on the simulation.

In each time-step the unique label, position, density, velocity pressure, kinetic energy boundary and gravitational potential energies, viscous dissipation and compressive energy are written to file for each particle. The list of output variables can be changed but at this stage such changes require modifications to source code and subsequent recompilation of the program.

4.2 Model Verification: The Evolution of an Elliptic Drop

The evolution of an elliptic drop is a simple test used by both Monaghan (1994) and by Bonet and Lok (1999) to verify the accuracy of the SPH models they created. The solution to the problem can be written as a pair of first order ordinary differential equations and while solutions to these equations would be difficult to obtain analytically they can be readily obtained using numerical techniques, such as the explicit Euler method or the family of Runge-Kutta methods. Solvers based on these methods are widely available and can find solutions to high accuracy.

The merit in testing SPH models against this problem is that a numerical solution can be readily obtained, and also that other researchers have examined the problem. So the performance of the model created here can be compared against two standards.

The initial set-up of the simulation is described and results are compared with the known solution and also with results from both Monaghan (1994) and Bonet and Lok (1999). The effects of the XSPH and CSPH variants on this simulation are also examined.

4.2.1 Background for the Elliptic Drop Simulations

In this section the theoretical solution of the evolution of an elliptic drop is examined. This simulation involves the generation of an initially elliptic drop that remains elliptic as time passes, a drop consisting of an incompressible fluid. To proceed, the question is posed: ***what initial velocity field is required?***, the answer to which comes from noting that the parametric equations of an ellipse are

$$\begin{aligned} r_x(\eta,\theta,t) &= a(t)\eta cos(\theta) \\ r_y(\eta,\theta,t) &= b(t)\eta sin(\theta) \end{aligned} \tag{4.5}$$

where r_x and r_y are the coordinates of all points within the ellipse and are written in terms of the parameters η and θ which satisfy $0 \leq \eta \leq 1$ and $0 \leq \theta \leq 2\pi$. The pair of variables $a(t)$ and $b(t)$ comprise the semi-major and semi-minor axes for the ellipse. The velocities within the ellipse can be written

$$\begin{aligned} v_x(\eta,\theta,t) &= \frac{\partial r_x}{\partial t} = \frac{r_x}{a}\frac{da}{dt} \\ v_y(\eta,\theta,t) &= \frac{\partial r_y}{\partial t} = \frac{r_y}{b}\frac{db}{dt} \end{aligned} \tag{4.6}$$

in which v_x and v_y are the components of the velocity vector. At any point $(x, y) = (r_x, r_y)$ within the ellipse Equations 4.6 can be rewritten in terms of the independent variables x, y and t as

$$\begin{aligned} v_x(x, y, t) &= \frac{x}{a}\frac{da}{dt} \\ v_y(x, y, t) &= \frac{y}{b}\frac{db}{dt} \end{aligned} \tag{4.7}$$

which are a pair of equations that must be satisfied if the drop is to remain elliptic, so answering the question of earlier.

A new variable can be defined

$$B = \frac{1}{b}\frac{db}{dt} \tag{4.8}$$

and because the fluid must obey the continuity equation (which, for an incompressible fluid, reduces to $\nabla.\boldsymbol{v} = 0$), this becomes

$$B \equiv \frac{1}{b}\frac{db}{dt} = -\frac{1}{a}\frac{da}{dt}. \tag{4.9}$$

Using this definition the velocity vector becomes

$$\boldsymbol{v} = (-x, y)B(t) \tag{4.10}$$

and by the product rule Equation 4.9 implies that

$$\frac{d}{dt}(a(t)b(t)) = 0 \tag{4.11}$$

so the condition that $a(t)b(t)$ is constant with respect to time is equivalent to the condition of conservation of mass. This is not surprising since the area of an ellipse is πab and conservation of mass is equivalent to conservation of volume for an incompressible fluid.

The law of conservation of momentum leads to

$$\frac{d\boldsymbol{v}}{dt} + (\boldsymbol{v}.\nabla)\,\boldsymbol{v} = -\frac{\nabla P(x, y, t)}{\rho} \tag{4.12}$$

which can be rewritten

$$\begin{aligned} \frac{d}{dt}\nabla\phi(x, y, t) + \nabla\left(\frac{1}{2}\boldsymbol{v}.\boldsymbol{v}\right) + \frac{\nabla P(x, y, t)}{\rho} &= 0 \\ \nabla\left[\frac{d\phi(x, y, t)}{dt} + \frac{1}{2}\boldsymbol{v}.\boldsymbol{v} + \frac{P(x, y, t)}{\rho}\right] &= 0 \end{aligned} \tag{4.13}$$

where it has been assumed that the flow is irrotational so a potential $\phi(x, y, t)$ exists such that $\boldsymbol{v} = \nabla\phi(x, y, t)$. From Equation 4.10 it can be concluded that

$$\phi(x, y, t) = \frac{1}{2}\left(y^2 - x^2\right)B(t) \tag{4.14}$$

Equation 4.13 simplifies to

$$\frac{d\phi(x, y, t)}{dt} + \frac{1}{2}\boldsymbol{v}.\boldsymbol{v} + \frac{P(x, y, t)}{\rho} = \frac{f(t)}{\rho} \tag{4.15}$$

where $f(t)$ is a function that is constant with respect to both x and y. This can be simplified by referring to Equations 4.10 and 4.14

$$\frac{dB(t)}{dt}\left(y^2 - x^2\right) + \left(y^2 + x^2\right)B^2 + \frac{P(x, y, t)}{\rho} = \frac{f(t)}{\rho} \tag{4.16}$$

from where it can be seen that $f(t) = P(0,0,t)$, the pressure at the origin.

Equation 4.16 can be written at the points $(x,y) = (0,b)$ and $(x,y) = (a,0)$ to get the two equations

$$b^2\frac{dB(t)}{dt} + b^2B^2 + \frac{P(0,b,t)}{\rho} = \frac{f(t)}{\rho} \tag{4.17}$$

$$-a^2\frac{dB(t)}{dt} + a^2B^2 + \frac{P(a,0,t)}{\rho} = \frac{f(t)}{\rho}. \tag{4.18}$$

By subtracting Equation 4.18 from 4.17, and by requiring that the pressure be constant on the surface of the ellipse it can be written that

$$\frac{dB(t)}{dt}\left(b^2 + a^2\right) + \left(b^2 - a^2\right)B^2 = 0. \tag{4.19}$$

This can be simplified to get

$$\frac{dB(t)}{dt} = B^2\frac{\left(\omega^2 - b^4\right)}{\left(\omega^2 + b^4\right)} \tag{4.20}$$

where $\omega = ab$. Together with Equation 4.9, the definition of B, this constitutes a system of first order ordinary differential equations that can be solved by any of the standard methods for such problems. Both Monaghan (1994) and Bonet and Lok (1999) record this expression incorrectly, having ω^4 rather than ω^2, but this error is trivial and has no effect since $\omega = 1$ in both references.

4.2.2 The Solution to the Evolution of the Elliptic Drop

Here the elliptic drop initially has $a = b = 1$. Also $v_x = -100x$ and $v_y = 100y$ which is consistent with Equation 4.7, the condition that must be met if the drop is to remain elliptic. The solution to the system of equations

$$\frac{dB(t)}{dt} = B(t)^2\frac{\left(1 - b(t)^4\right)}{\left(1 + b(t)^4\right)} \tag{4.21}$$

$$\frac{db(t)}{dt} = b(t)B(t) \tag{4.22}$$

is desired, subject to the initial conditions $b(0) = 1$ and $B(0) = 100$. This was solved here by using the *ode45* solver provided with MATLAB, which employs a Runge-Kutta method. A plot of $b(t)$ against time would be nice, but it is not really very interesting since it appears to be a straight line to the naked eye.

4.2.3 The Initial State

The elliptic drop simulations were generated using 1884 free points regularly distributed on a rectangular grid within a circle of radius 1 m with velocity initially $v_x = -100x$ and $v_y = 100y$. This initial state is plotted in Figure 4.1. The simulations used here are designed to be as similar as possible to that of Monaghan (1994), using the same number of particles initially distributed in the same pattern.

The speed of sound was chosen to be $1400\,\mathrm{ms}^{-1}$ (close to the measured value for water) and the density $1000\,\mathrm{kgm}^3$. Four simulations were created that examined the effect of using the XSPH (Page 61) and CSPH (Page 62) variants and also a combination of the two.

Monaghan (1994) does not specify whether viscosity was included, so it has been assumed that it was not used, and consequently viscosity has not been used here either. The simulation

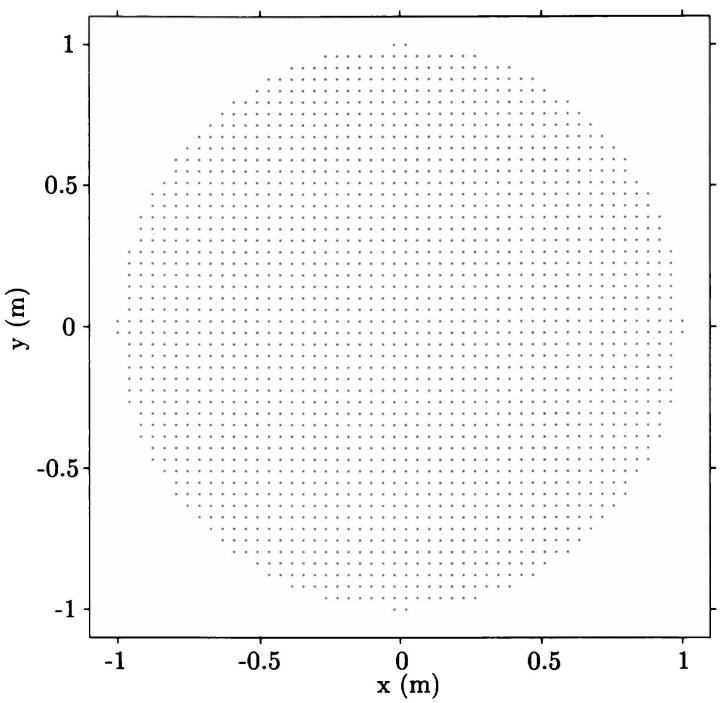

Figure 4.1: *Particle positions in the initial state of the elliptic drop simulations. This is designed to be as close as possible to the simulation used by Monaghan (1994).*

used by Bonet and Lok (1999) differs in the number of particles, in the initial distribution, and in the inclusion of viscosity. Monaghan (1994) does not state whether the XSPH variant was used in his simulation, but there is evidence that it was, as is discussed in Section 4.2.5.

4.2.4 Results of the Elliptic Drop Simulations

Because different initial states were used by Monaghan (1994) in comparison with Bonet and Lok (1999), and because viscosity was not accounted for, the results of the two cannot be expected to agree exactly. The simulations performed here are modelled closely to those of Monaghan (1994) and so should compare well with those. Comparisons with Bonet and Lok (1999) may not be as good.

The distinguishing features of the four simulations performed here are summarised in Table 4.1. In the Sections 4.2.5 and 4.2.6 the simulations are compared against the plots and tabulated data extracted from Monaghan (1994) and Bonet and Lok (1999). Differences in performance are noted and reasons for these stated. In Sections 4.2.8 to 4.2.9 the conservation of energy and of linear and angular momentum in the elliptic drop simulations are examined.

4.2.5 A Comparison With Monaghan's Results

Table 4.2 shows a comparison of results from the elliptic drop simulation performed here with those of Monaghan (1994). The semi major axis was extracted from results quite simply by taking the y coordinate of the highest point in each of the selected time-steps.

The use of XSPH causes the values of the semi-major axis to be consistently (but only slightly) smaller than the values found when XSPH is not used. This is because the effect of using XSPH is to retard the movement of particles, as they are moved according to the averaged velocities of the neighbouring particles.

Simulation #	Variant
1	SPH
2	CSPH
3	XSPH
4	CXSPH

Table 4.1: *The use of the* XSPH *and* CSPH *variants in the four simulations of the evolution of an elliptic drop. Variants are identified by the use of* C *or* X *preceding the term* SPH, *and* CXSPH *means both variants were used.*

Time (ms)	Theory	Monaghan (1994)	SPH	CSPH	XSPH	CXSPH
0.8	1.0831	1.086	1.0819	1.0820	1.0813	1.0810
3.8	1.4392	1.44	1.4369	1.4357	1.4337	1.4271
7.6	1.9445	1.91	1.9371	1.9367	1.9313	1.9191

Table 4.2: *Values of the semi-major axis for the elliptic drop simulations and the effect of the* CSPH *and* XSPH *variants. The term* CXSPH *means both variants were used. Results from Monaghan (1994) are included.*

Monaghan's values for the semi-major axis are within 2 % (not very accurate since Monaghan has used few digits) of the theoretical values and results here are 0.1 % - 0.8 % lower than the theoretical values. The reason for the slight difference is not clear. It may be that Monaghan's time-stepping approach of calculating the velocities first and then the densities (in MARIAN the opposite is done) explains the difference. Not stated in Monaghan (1994) is the step-size used and it may be that a shorter time-step was used here, resulting in more accurate simulations. The existence of programming errors in source code cannot be discounted. Irrespective of the reason, these differences are slight and do not represent pressing concerns.

Monaghan (1994) presents plots of particle positions at two times, the first of which (at time 0.8 ms) is not very interesting because it varies little between results here and Monaghan's. The second plot is notably different to some of the plots here and is reproduced in Figure 4.2. The particle positions plotted in Figure 4.3 vary in the particle distributions. The first simulation, using neither of the two variants, has a distinct distortion near the free surface, a distortion caused by clustering.

Distortion due to clustering does not appear in Figure 4.2, perhaps in part due to the lower resolution of copied image, but more likely explained by the use of XSPH. Monaghan (1994) did not use CSPH but is very likely to have used XSPH, an assertion strengthened by the similarity of the XSPH plot of Figure 4.3 and Figure 4.2.

4.2.6 A Comparison With Bonet and Lok's Results

Table 4.3 is taken from Bonet and Lok (1999) while the data in Table 4.4 are from the simulations performed here. Significant in these tables are that only slight differences appear between the simulations performed here. However the values for the semi-major axis in Table 4.4 are always less than 0.9 % from the theoretical values.

Notable differences in the model setup between MARIAN and the model of Bonet and Lok (1999) are the time-stepping technique used, the viscosity formulation, and the initial state of the simulation. Bonet and Lok (1999) did not disclose the step-size they used and this is most likely to be a point of difference. Bonet and Lok (1999) use a leap-frog scheme while MARIAN uses the same two-stage predictor-corrector method used by Monaghan (1994) (except in MARIAN

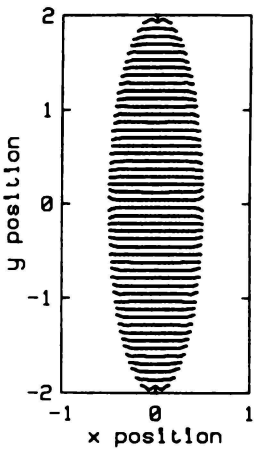

Figure 4.2: *Particle positions for Monaghan's elliptic drop at 8.2ms, after Monaghan (1994).*

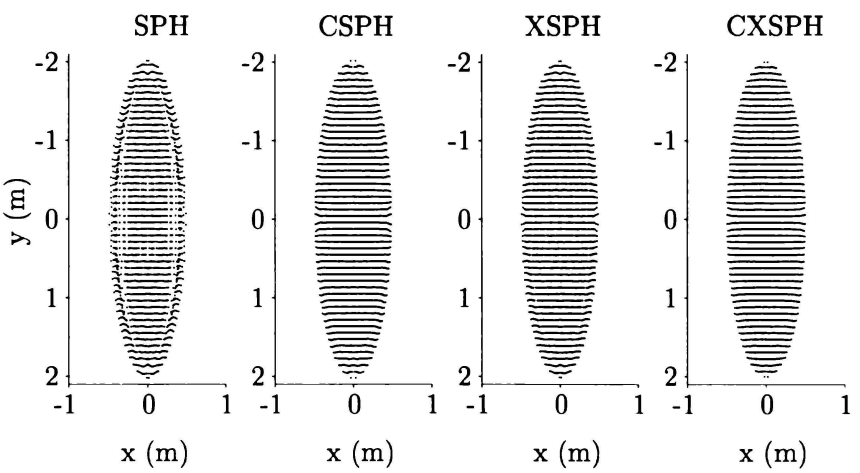

Figure 4.3: *The effects of the XSPH and CSPH variants on particle positions for the evolution of an elliptic drop.*

Time (ms)	Theory	SPH	CSPH
2	1.22	1.22	1.22
5	1.60	1.61	1.60
8	2.00	2.02	2.00

Table 4.3: *Values of the semi-major axis for the evolution of an elliptic drop from Bonet and Lok (1999).*

Time (ms)	Theory	SPH	CSPH	XSPH	CXSPH
2	1.2190	1.2168	1.2168	1.2153	1.2124
5	1.5948	1.5913	1.5899	1.5872	1.5784
8	1.9993	1.9900	1.9913	1.9848	1.9726

Table 4.4: *Values of the semi-major axis for the elliptic drop simulations in comparison to theoretical results.*

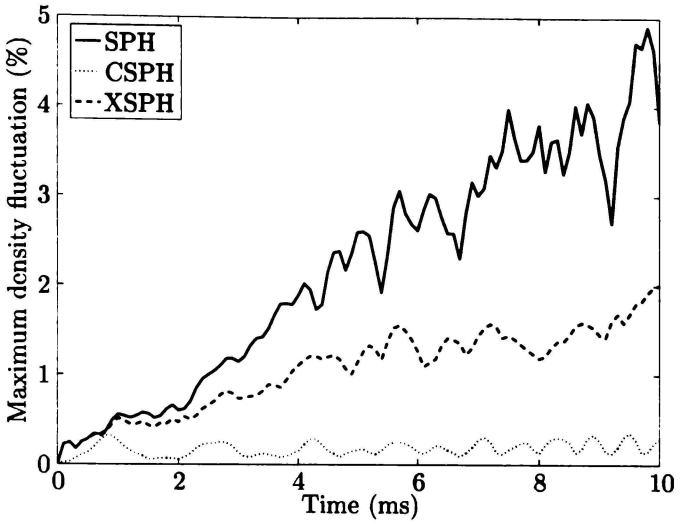

Figure 4.4: *The maximum deviation from the standard density for the elliptic drop simulations. Curves for* CSPH *and* CXSPH *are indistinguishable.*

the density is calculated first, rather than the velocities), but both methods have truncation error that scales with the cube of step-size. The effect of viscosity is not accounted for in the simulations here, while it is in Bonet and Lok (1999).

The initial states used to create simulations here differ from those used by Bonet and Lok (1999) since they had 1184 randomly distributed particles within a circle of particles as their initial state, whereas here 1884 particles were arranged on a rectangular grid. Also Bonet and Lok have placed a point at the semi-major axis for which $x = 0$, while here and in Monaghan (1994) points were placed on either side of the x-axis. So both the results of Monaghan and the results here contain an error due to the lack of a particle for which $x = 0$, and it is estimated that the results are consistently under-predicted by 2.0×10^{-4} to 2.5×10^{-3}.

But while the causes of the differences in results are not obvious, the magnitudes of these errors are small and can be disregarded. Changes to the model and to the initial simulation state could establish these causes. Changes to the model could include using identical initial states, viscosity formulations and time-stepping techniques, but these will not be followed up here.

4.2.7 Density Variation in the Elliptic Drop Simulations

Monaghan (1994) notes that the change in the variation in the density $\delta\rho$ in an SPH simulation can be predicted from

$$\frac{\delta\rho}{\rho} \approx \frac{v_{max}^2}{c_{ref}^2} \tag{4.23}$$

which is written in terms of typical velocity v and the speed of sound c. For example, if the speed of sound is doubled the variations in density should reduce to a quarter of the original value. For the elliptic drop simulations $v = 100\,\mathrm{m/s}$ (the maximum initial speed) is chosen arbitrarily, and since $c = 1400\,\mathrm{m/s}$, density fluctuations are less than 0.5 %.

The maximum deviation from the standard density ($1000\,\mathrm{kg.m^{-3}}$) is plotted for the four simulations in Figure 4.4. The best performance with regard density fluctuation occurs for the CSPH and CXSPH simulations for which are the same in plot 4.4. The growth in the variation of densities for the SPH simulation can be attributed in part to the occurrence of clustering.

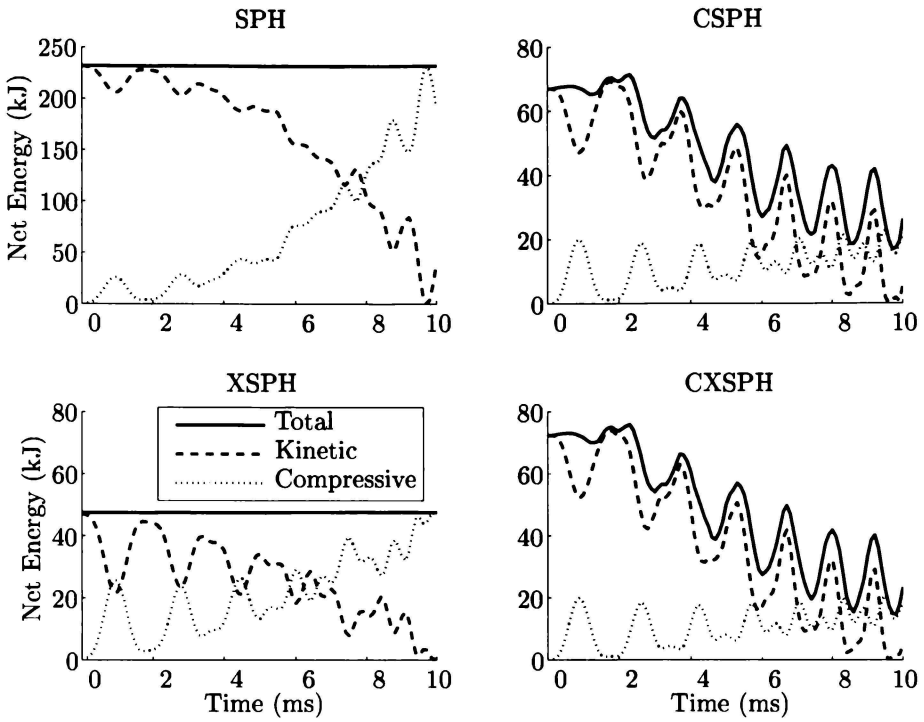

Figure 4.5: *The net total, compressive and kinetic energies against time for each of the elliptic drop simulations. Note the different vertical scale of the SPH plot.*

4.2.8 Conservation of Energy in the Elliptic Drop Simulations

The principle of conservation of energy is fundamental to any CFD model and a significant test of the performance of any model is whether this principle is satisfied. Figure 4.5 contains plots of the net total, compressive potential and kinetic energies against time. The vertical scale on the SPH plot is different to those for the other plots.

All of the plots in Figure 4.5 feature the periodic exchange of energy between kinetic and compressive energy modes. This exchange occurs with a frequency of approximately 700 Hz, varying slightly between simulations. Quite simply, a sound wave is generated that propagates through the fluid. Changing the velocity of sound will change the frequency of this feature.

All of the plots show a net increase in the kinetic energy and a decrease in the compressive potential energy. These trends occur because the energy stored in the sound wave (discussed in the previous paragraph) is being dispersed through the liquid. This effect is similar in all of the simulations.

The trends of decreasing compressive potential and increasing kinetic energies are most evident in the SPH simulation, indeed the magnitude of the effect is three to four times larger than any of the other simulations. The magnitude of the trend in the SPH simulation is due to clustering, a numerical effect that causes unphysical variations in density, generating larger variations in density in the SPH simulation than in any other.

In the SPH and XSPH simulations energy is conserved because the lines for the net total energies are flat. In fact the net total energy lines in the SPH and XSPH plots have small, positive slopes that are too low to see on these graphs. This growth in energy is generated by the accumulation of errors.

But most notable in Figure 4.5 is the failure to conserve energy in the simulations using the CSPH and CXSPH variants, evident in that the net total energy lines are not flat. The use of kernel gradient correction may well enforce conservation of angular momentum (Bonet and Lok,

1999) and cause more even particle distributions (evident in Figure 4.3), but it does so only at the cost of energy conservation.

4.2.9 Conservation of Linear and Angular Momentum in the Elliptic Drop Simulations

Bonet and Lok (1999) contains results for the evolution of the elliptic drop (using SPH without correction) in which linear momentum is conserved to within rounding errors, but angular momentum is not. This is not reflected in the work here, where exact conservation of both linear and angular momentum was achieved. The difference may have been due to the different choice for the SPH equations, since here the continuity equation has been used while Bonet and Lok (1999) used the summation interpolant.

4.2.10 Summary of the Elliptic Drop Simulations

MARIAN was tested by its ability to simulate the evolution of an elliptic drop and results were extracted that showed good agreement with those from the literature for the same problem. The effect of the XSPH and CSPH variants on the simulation were examined and it was demonstrated that the use of XSPH, CSPH or CXSPH generated more even particle distributions than if SPH were used. It was noted that energy conservation was poor for the CSPH and CXSPH simulations while being good for the SPH and XSPH simulations. The justification for using CSPH (enforcing angular momentum conservation) did not apply here because this quantity was already conserved in SPH and XSPH simulations of the elliptic drop. The CSPH variant does produce results with fewer distortions near the edges of the ellipse, than do the XSPH and SPH variants. The maximum density variations increase at a significant rate for SPH, at a lesser rate for XSPH while for CSPH and CXSPH they remain consistently small.

4.3 The Bursting Dam

Monaghan (1994) and Bonet and Lok (1999) created SPH simulations of bursting dam experiments performed by Martin and Moyce (1952) as tests for the veracity of their respective models. The same tests are performed here. In the following sections the experiment are described, the initial state of the simulation are outlined, and then the comparison with the experimental data of Martin and Moyce (1952) are conducted.

4.3.1 The Experimental Results

The bursting dam experiments were conducted in a channel constructed from Perspex, shown schematically in Figure 4.6. The cavity on the right was initially filled with water, and when a current was passed through the wires a horseshoe-shaped element was heated, melting the beeswax that fastened a sheet of waxed paper to the element. With the sheet of paper no longer attached the water column was free to collapse and its motion was filmed using high speed ciné cameras.

Martin and Moyce (1952) performed this experiment with water columns that were a variety of shapes, including a cubic column 57.15 mm on each side. Results from this experiment are the subject for comparison here. From films Martin and Moyce (1952) extracted data for the change in height and surge front with time. The same experiment was performed a number of times and results were tabulated, making this experiment ideal for such comparisons since a lot of detail is available.

There were some uncertainties when times were recorded, since exact synchronisation of the release of the waxed paper with the timer was difficult. To account for this Martin and

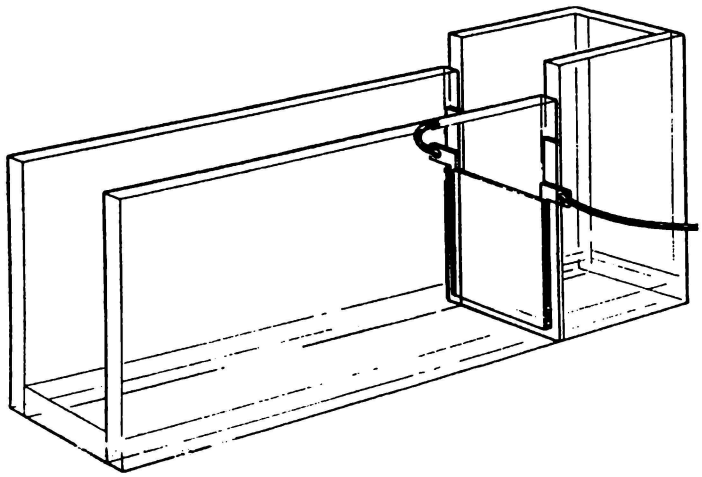

Figure 4.6: *The apparatus used by Martin and Moyce (1952) in their examination of collapsing dams.*

Moyce (1952) normalised data to have the same time (61.071 ms, which is a round number in the dimensionless units they used) when the surge front was a certain position (82.296 mm).

4.3.2 The Initial State

The initial state of the simulation is simply a square consisting of 2916 particles arranged on a rectangular grid, creating a water column with the same dimensions as that used by Martin and Moyce (1952). Normal/tangential boundaries were used because they avoid the problems of corrugations found with the ghost particle boundaries.

Monaghan (1994) estimates the maximum velocity to be

$$v_{max} = \sqrt{2gH}, \tag{4.24}$$

an estimate that was also used here. The Mach number is selected to be 0.1 meaning that the velocity of sound in the reference state (which has a density of $1000\,\mathrm{kg/m^3}$ and a pressure of 0 Pa) is $10.587\,\mathrm{ms^{-1}}$. The CSPH variant was not employed but XSPH was, using $\epsilon = 0.5$ (see Equation 3.72). The artificial viscosity discussed on Page 61 was used with $\alpha = 0.01$ and $\eta = 0.01$. VMOSS was used for 1000 time-steps with velocities set to zero every 50 th step. The smoothing length was 0.002 m which is just under twice the initial grid spacing, so a particle will have 40-50 neighbours (provided it is not near any boundaries). The step-size was chosen to be $20\,\mu$s, and the simulation ran for 20 000 steps with output written every 200 steps.

4.3.3 Results and Discussion

In this section the results of the bursting dam simulation are presented and discussed. First the positions of free particles at various times during the simulation are presented and coarse features in these plots examined. Comparisons to the experimental results for the position of the surge front and the height of the water column follow. The effect of the choice of method for constructing boundaries concludes this section.

Particle Positions

In Figures 4.7 and 4.8 the positions of particles at various time-steps during the bursting dam simulation are plotted. Frames up until the surge front has hit the far wall are included. These

frames show free surfaces that are reasonably smooth, but are not, perhaps, as smooth as might be generated by the use of CSPH.

A surprising feature of the simulation is the behaviour of the flow at the left boundary. Initially the water column is repulsed by the boundary force, but as the main water column drains away a layer of water is left behind. This layer stands by itself for a while and then slowly collapses onto the main water body, creating a disturbance in the free surface. This feature is explained by the phenomenon of ***stacking***, which occurs because particles are prone to stack when shear forces are small and is further explained in Section 4.4. The effect is not so apparent if ghost particles are used to create the boundary because damping of movement tangential to the boundary does not occur when ghost boundaries are employed. Notably, none of the simulations Bonet and Lok (1999) performed have this feature, but it does appear in Monaghan (1994).

Also present is the effect of clustering, which disturbs flow wherever it occurs. Figure 4.9 contains a plot showing the bottom left corner of the bursting dam simulation 16 ms after the start. The repulsive boundary forces from both vertical and horizontal boundaries have combined with the pressure gradient to cause several clusters to form.

The literature describes clustering and identifies the cause but does not offer a definition that can be used to identify when clustering has occurred. If Monaghan's cubic is used particles less than $2h/3$ apart could be defined as constituting a cluster since at this point the gradient in the kernel is at a maximum.

The Position of the Surge Front for the Busting Dam

Martin and Moyce (1952) published tabulated values for the position of the surge front for their bursting dam experiments. A selection of these are plotted in Figure 4.10 with results from the simulation performed here. Martin and Moyce (1952) determined that a significant bias existed in the times recorded for each experiment, and normalised their results to ensure agreement. This bias was generated by their inability to synchronizer the time-keeping and the initiation of the experiment.

The similarities between the two data sets in Figure 4.10 are obvious, both showing similar behaviour for the surge front. The simulation shows a greater acceleration than the experiment and the difference cannot be explained as experimental uncertainty, which is indicated by the spread of points in the red series.

Figure 4.10 is very similar to that produced by Bonet and Lok (1999) (reproduced in Figure 4.11) but their results for both SPH and CSPH show a surge front slightly accelerating faster than any of the simulations here. The surge front position was evaluated by finding the position of the rightmost particle. Evident is that the surge front accelerates faster in the simulation than in the experiment. Bonet and Lok (1999) attribute this to the effect of friction that is not accounted for in the simulation. Also significant is surface tension and the method of generating boundaries. All data series in Figure 4.10 were normalised to the same point as the experimental data, and where necessary MATLAB's *interp1* function was used to determine the necessary constants.

In the experiment the effect of friction is to generate a boundary layer, and because of the small scale of the experiment this boundary layer will quickly propagate in to the water body. The simulation fails to account for friction at all. A related issue is the means by which boundaries were created in the simulation. If ghost particles were used the boundaries would be corrugated and the motion of the surge front would be slowed as horizontal momentum is transformed to vertical momentum, which is evident in Figure 4.10. Bonet and Lok (1999) do not state how they created boundaries so this may affect their results. Surface tension is significant at the small scale of the experiment but is not accounted for in the simulation performed here. The effect of this phenomenon would be to retard the acceleration of the surge front.

The first three points from Monaghan (1994) agree very well with experiment and other

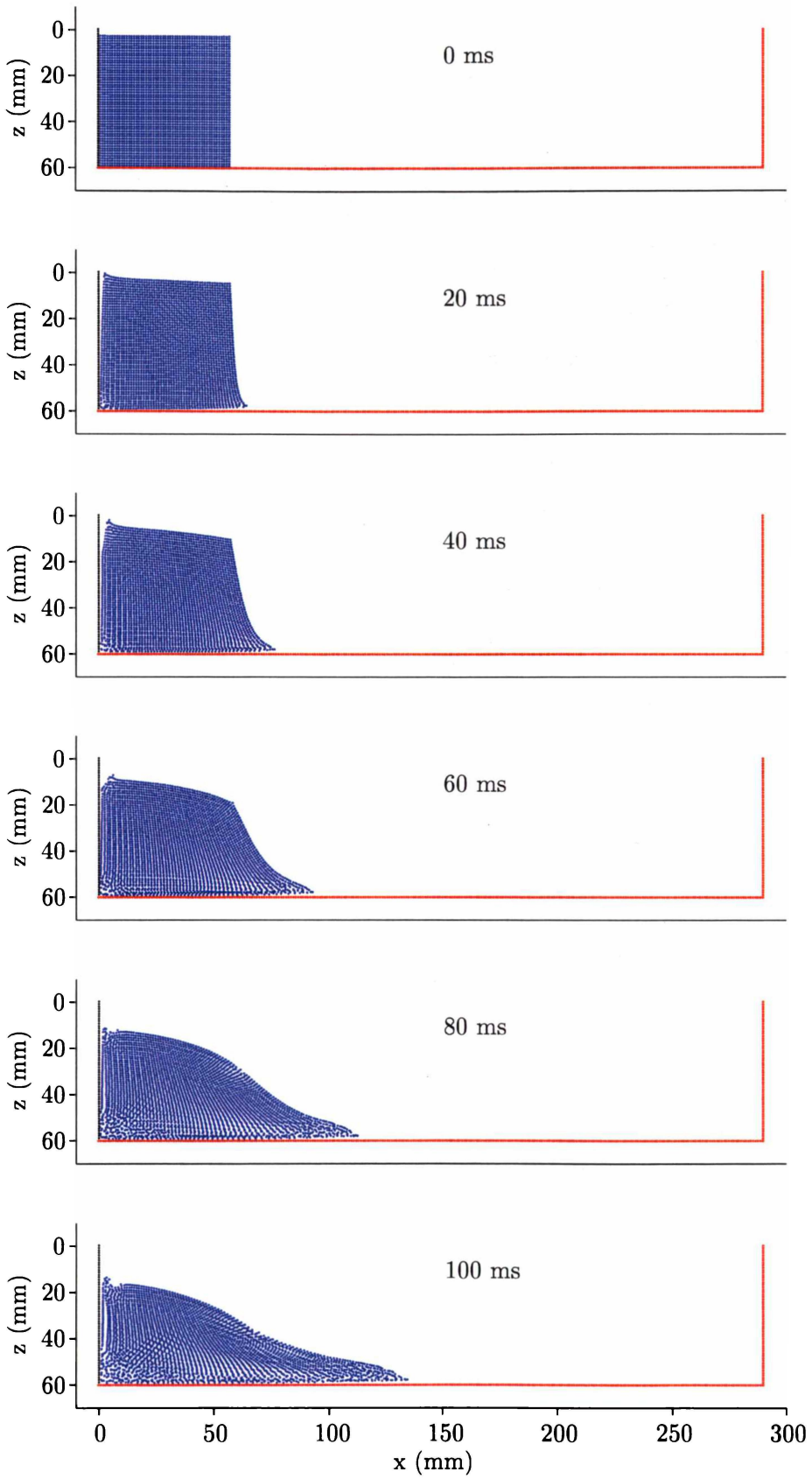

Figure 4.7: *Particle positions during the bursting dam simulation: 0 ms to 100 ms.*

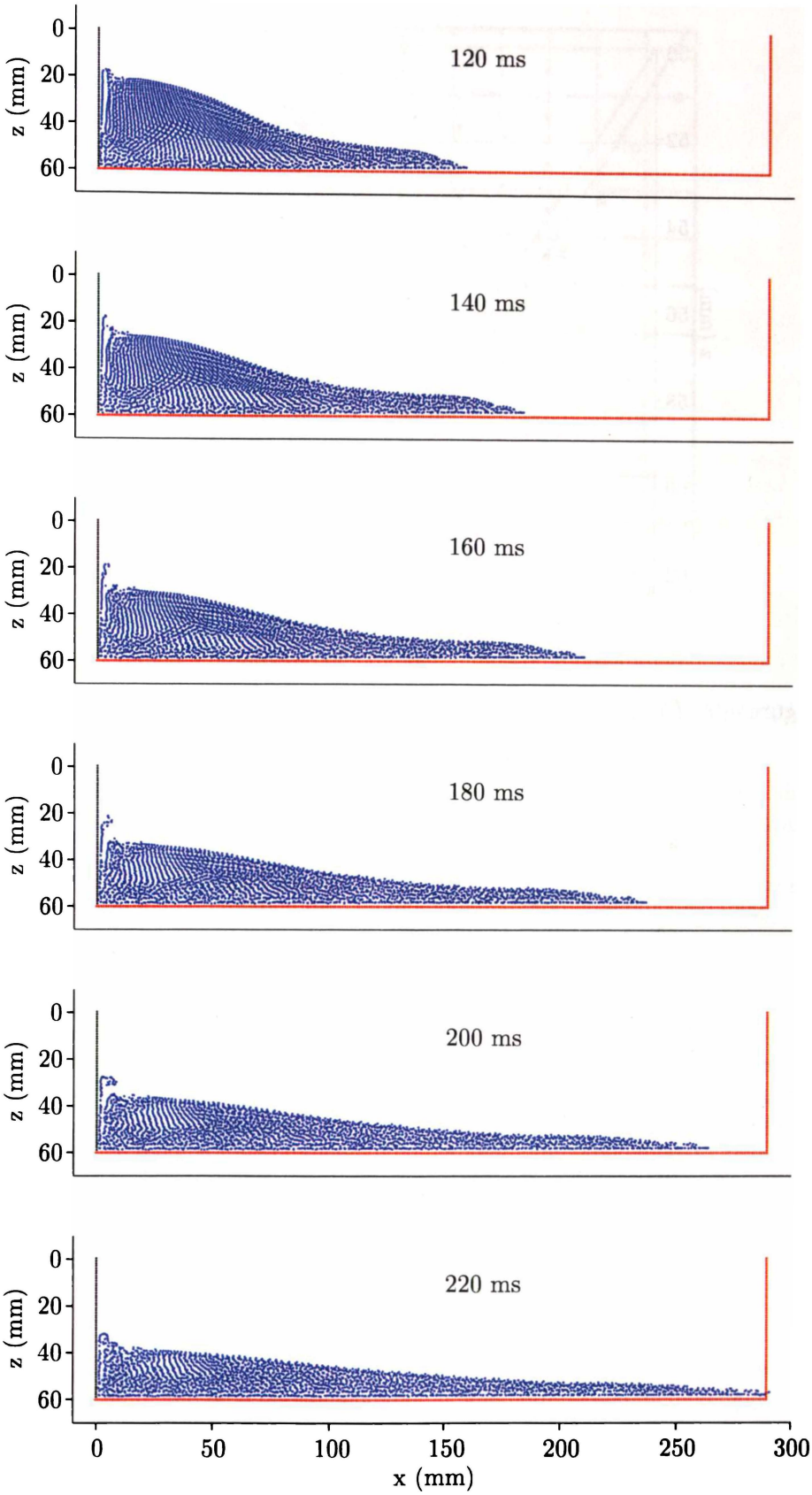

Figure 4.8: *Particle positions during the bursting dam simulation: 120 ms to 220 ms.*

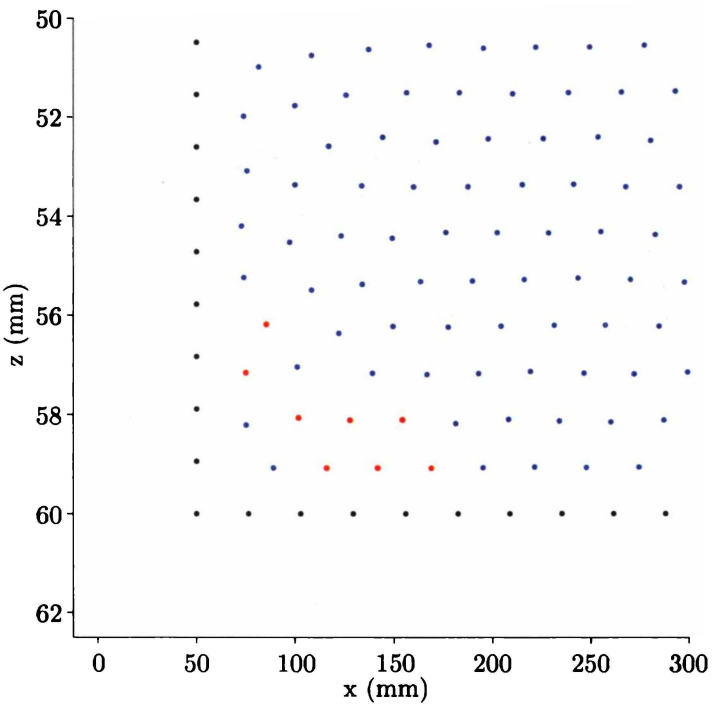

Figure 4.9: *The bursting dam at t=16 ms, showing clustered particles in red.*

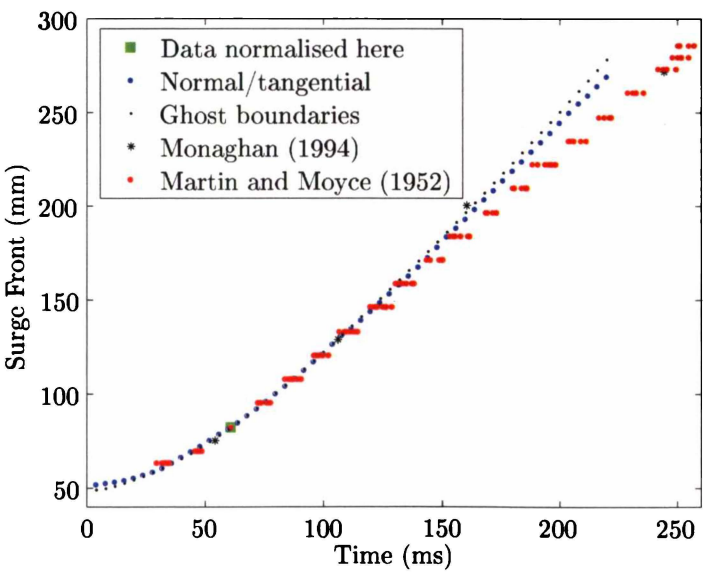

Figure 4.10: *The position of the surge front for the bursting dam. All of the experimental points from Martin and Moyce (1952) have been plotted to indicate the uncertainty. All data was normalised to ensure agreement at the single point shown. The position of the surge front is measured from $x = 0$, the rear of the water column. Two simulations using* MARIAN *were performed here, one with ghost boundaries and one with normal/tangential boundaries.*

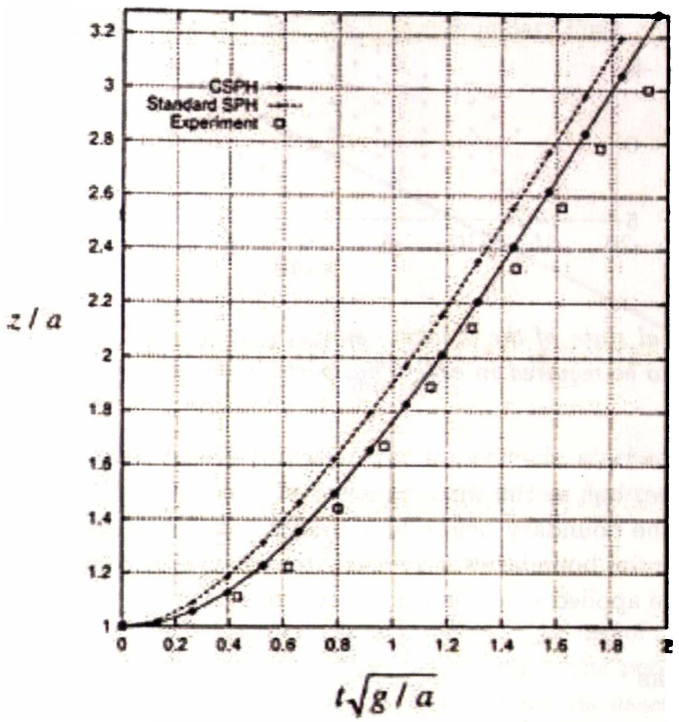

Figure 4.11: *The position of the surge front from Bonet and Lok (1999).*

simulations, but the excellent agreement for his fourth point is peculiar since it does not show the same trends as other computational results. No explanations for this are apparent to the author.

Martin and Moyce (1952) also tabulate the height of the water column, but comparisons with the simulation are difficult here. The effects of stacking at the left boundary make the column height ill-defined and so comparisons will not be made here.

Energies in the Bursting Dam

There is little point in presenting data on energies in the bursting dam simulation. Normal/tangential boundaries do not have a potential from which boundary potential energy can be evaluated, and consequently net total energy is not conserved. So it is impossible to determine the appropriate values for boundary force parameters that minimise spurious boundary potential energy. VMOSS can still be used to extract some spurious energies, but energy contributed by the vertical boundary will not be removed. This is a significant weakness and a boundary force of the normal/tangential type for which the potential is defined, would be of value.

Boundaries

The method used to create boundaries in MARIAN have a significant effect on the behaviour of the bursting dam simulation and in this case the normal/tangential method described by Monaghan and Kos (1999) was used. It can be seen in Figures 4.7 and 4.8 that particles near the top left boundary are repelled in a manner that creates an unphysical wedge-shaped gap, especially in the first few frames. It can also be seen that the surge front is elevated from the horizontal boundary by a small amount, a typical effect in SPH simulations (Monaghan, 1994). Clearly the method could be improved to reduce these effects.

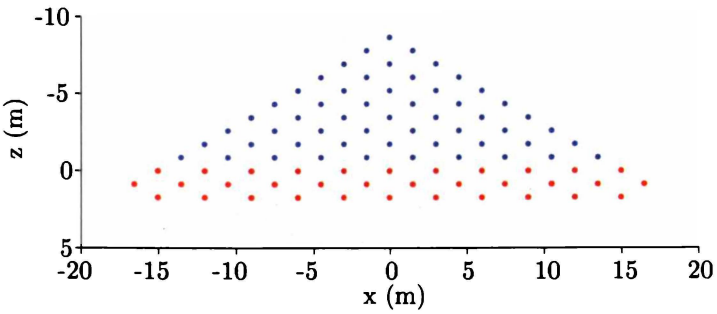

Figure 4.12: *The initial state of the stacking simulation. Three rows of boundary particles (the red ones) were found to be required in order that particles be constrained. The blue points are the free particles.*

It is difficult to see, but in the final frame of Figure 4.8 a particle at the leading edge of the surge penetrates the boundary before it is accelerated back inside the channel. This occurs because normal/tangential boundaries were used. Irrespective of where a particle is with respect the boundary the force applied is in the same direction.

Concluding Remarks

The most significant conclusion from the comparisons discussed here is that MARIAN simulated the bursting to an accuracy comparable with other SPH models. Comparisons with experimental results are good but not within experimental uncertainty, explained by the lack of surface tension and boundary friction in MARIAN. The effects of particle stacking and artificial clustering can be observed. The weaknesses of the normal/tangential and the ghost particle methods have been pointed out.

4.4 Particle Stacking

Particle stacking is the effect where particles *stack*, rather like oil drums. The effect is evident in some simulations where a particle is impeded by another particle. If the particle does not have enough energy it will not be able to get past, but will if it does. The larger the force acting on the particle the higher the chance it will get past.

In this section a simple simulation are presented that demonstrates clearly the effect of particle stacking. The initial state are described and the results discussed, in particular the implications of this effect on the modelling of fluid flow are examined.

4.4.1 The Initial State

In this simulation points are arranged on a triangular grid in the shape of a triangle approximately 8.5 m high, plotted in Figure 4.12. The simulation used 55 free particles (each with a mass of 2666 kg) and 33 ghost points for the boundary. In this simulation the reference density and speed of sound were 1000 kg/m^3 and 1500 m/s respectively. The smoothing length was 0.9 m and the step-size was 0.1 ms. Neither the XSPH nor the CSPH variants were used and artificial viscosity was turned off. Data was written to file every 200 th time-step. The VMOSS subroutine was used for 40 000 time-steps and the main simulation ran for 40 000 time-steps also.

The aim of this simulation is to demonstrate the effect of stacking. The ghost boundary method was selected for creating the boundary since the corrugated boundary force provides a stable base, discouraging movement in the lowest layer of free particles.

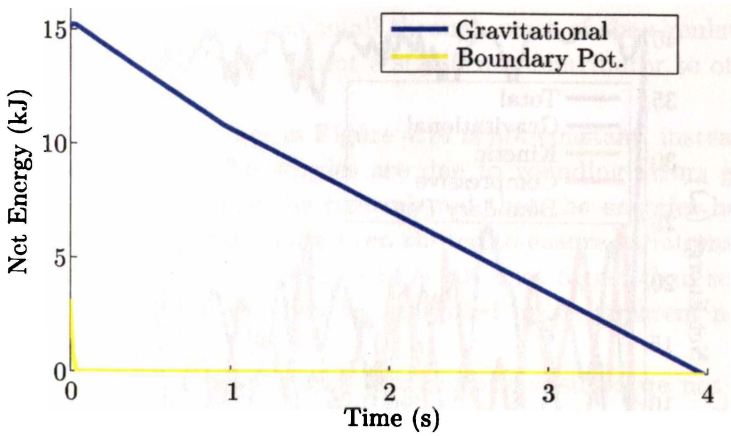

Figure 4.13: *Energies during VMOSS operation for the particle stacking simulation. Viscous dissipation, kinetic and compressive energies are not plotted in this figure since they are constant.*

4.4.2 Results and Discussion

In this section the stacking simulation are examined in detail. The use of the VMOSS subroutine are discussed with attention to the net energies of the system. The net energies and the final state of the main simulation are examined, and these results are discussed.

The Role of VMOSS

The VMOSS subroutine was used to ensure that initial state of the system was free from spurious perturbations that could provide some particles with enough energy to escape from the stack. The velocities were set equal to zero every 20 th time-step. The energies during the VMOSS operation are plotted in Figure 4.13.

The main features of Figure 4.13 are the initially high boundary and gravitational potential energies. So some particles were inadvertently set too close to the boundary and the entire water column was set too high. That the compressive energy is constant indicates the particles were given initial pressures very close to the correct values. The kinetic energy is constant at zero because in the same step that data is written to the file, the velocities were all set equal to zero.

The dominance of gravitational potential energy (over compressive and boundary potential energies) indicates the system still has spurious energy. But from this point stacking could still be demonstrated so there is little point in further smoothing the system. The particle positions at the end of VMOSS operation are not dissimilar to those in the initial state, differences cannot be observed with the eye, and consequently will not be presented here.

The Main Simulation

From the smoothed system provided by VMOSS, the main simulation ran for 40 000 time-steps, corresponding to a time of 4 s. The energies are plotted for this period in Figure 4.14. Once again, particle positions at the end of the simulation are not obviously different to the plot of the initial state and will not be presented here. There is an interesting feature on the side of the stack however, and this is plotted in Figure 4.15.

It is clear from Figure 4.14 that from the system state after VMOSS operation, the free particles fall (the kinetic energy increases) until increasing pressure gradients start to act (the compressive energy increases). Then they are decelerated, the velocities decrease and soon the centre of mass of the stack is increasing. Basically, the system is like a bunch of balls, bouncing in unison.

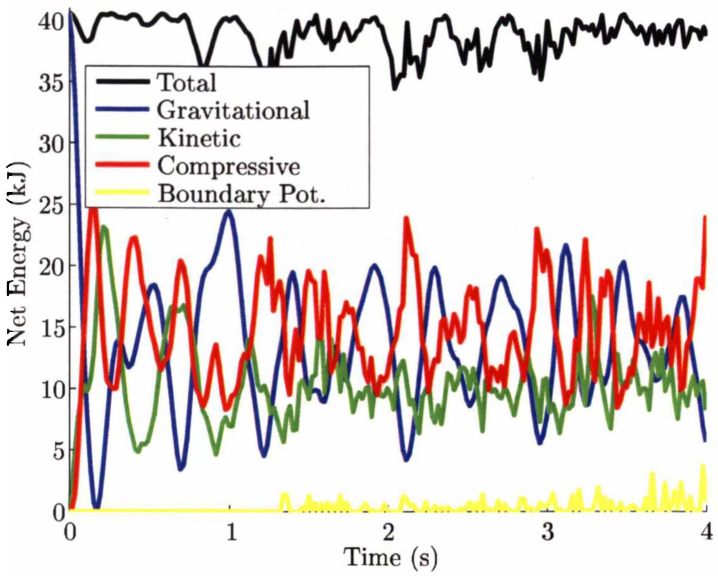

Figure 4.14: *Energies during the particle stacking simulation.*

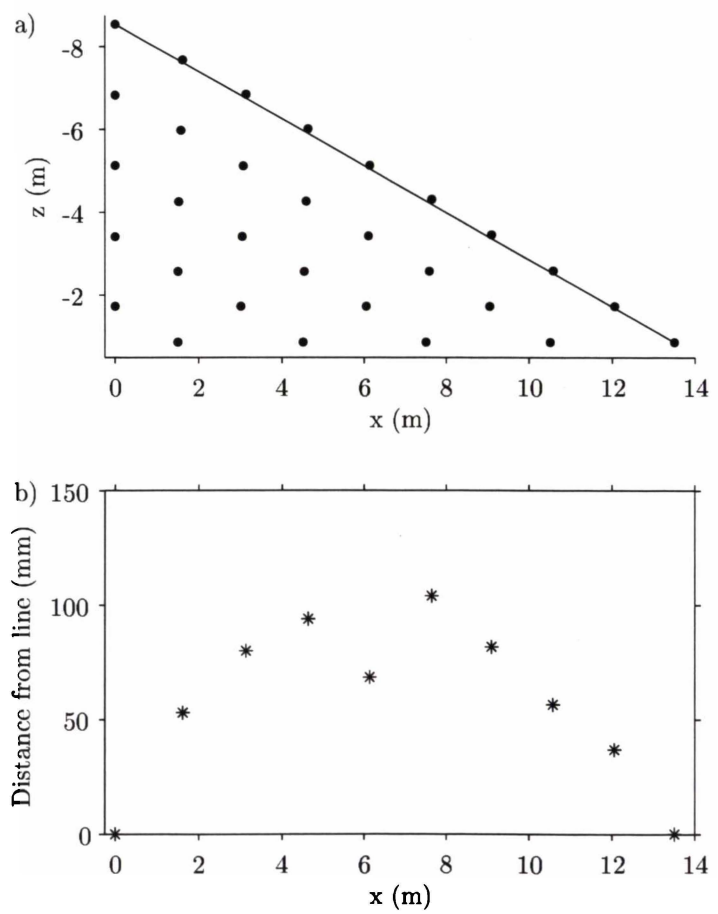

Figure 4.15: *The positions of particles on the side of the stack* (a)*, shown with a straight line passing through the top left and lower right points to indicate how pressure gradients have slightly distorted the stack. Plot* (b) *shows the orthogonal distance of particles along the side to the line.*

The boundary potential energy remains small through most of the simulation, and the only value in its inclusion is to indicate that it is not a significant contributor to other features, such as the nature of the net total energy.

It is notable that the net total energy in Figure 4.14 is not constant, instead it wiggles about an approximately constant value. The wiggles are due to rounding errors generated by both the model and post-processing. It must be remembered that the energies here are relative to arbitrary reference values, so all energies have been shifted to ensure variations are visible on the plot. The scale of the largest feature in Figure 4.14 is not a lot bigger than scale of the features due to rounding errors, so no importance can be attributed to the apparent non-conservation of energy.

Figure 4.15 shows that the particles along the sides of the stack are not in a straight line, instead they form a concave-down curve. This is what one would expect and shows that forces are being generated by pressure gradients that push the particles up and out from their original positions (which were on a straight line).

The consequence of particle stacking is not just that sometimes particles will stack, but that any flow with a low shear will not be simulated well. Instead of single rows of particles sliding across one another, layers that are several particles thick may form. This effect is not just a factor for SPH models, but will also probably occur for MPS.

Stacking means that when the continuity equation is used to find the density (rather than a summation interpolant), boundaries created by the number density method are corrugated. The author expects that this will also occur when the summation interpolant is used for the density, but has not demonstrated this assertion.

Stacking is very much dependent on the particle configuration: if this simulation were run with particles initially arranged on a rectangular grid it is much less likely stacking would occur. A significant problem is that it is not yet possible to look at a complex flow and be able to determine if stacking is occurring anywhere within the flow.

Particle Stacking in the Literature

At this point it is interesting to examine results from a simple SPH simulation performed by Lombardi *et al.* (1999). They used a three dimensional box with periodic boundaries containing an initially rectangular array 6859 particles with zero velocities. After running for a time they found their simulation had reached a new equilibrium state, one in which the particles had *crystallised* into a regular three dimensional triangular lattice. They termed this effect ***spurious transport***, and recognised that it provides a mechanism by which energy can be transported in a system.

Lombardi *et al.* (1999) note that an equilibrium state occurs whenever the average particle velocities are below a critical velocity. If too much noise exists in a system, the average velocity is above the critical velocity, and the system will not crystallise. They observe that the critical velocity reduces as the smoothing length increases. This is an important observation, and from this point a link can be drawn between spurious transport and the error in the quasi Monte-Carlo estimate (the means by which field variables are discretised in SPH). Both the error and the critical velocity decrease as the smoothing length is increased. So to avoid stacking: use more points and larger smoothing lengths.

It is no surprise then, that stacking should occur in the simulation presented here. Particles were placed close enough to equilibrium positions that they crystallised almost immediately (with a little help from the VMOSS subroutine). Particle stacking is subtly distinct from the spurious transport reported by Lombardi *et al.* (1999), because it represents the reluctance of a system to leave the crystallised state, whereas spurious transport is the process where particles move into the crystallised state. The full consequences of particle stacking and spurious transport for SPH

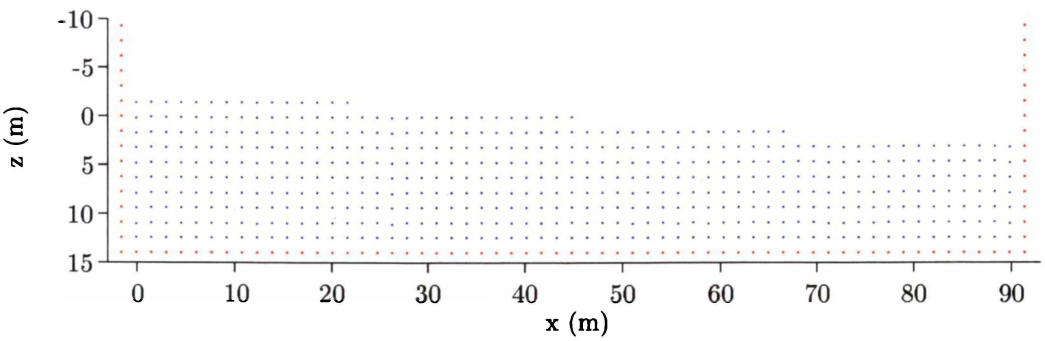

Figure 4.16: *Initial particle positions for the seiching basin simulations. This is the state of the system after VMOSS operation.*

models used to simulate nearly incompressible, free surface fluid flow have not been established, nor has the source of the effects been identified.

4.5 A Seiching Basin

This simulation involves a rectangular basin with an initially sinusoidal free surface. It is anticipated that the system will seiche, that is, a standing wave will exist. This set of simulations is presented here for a number of reasons. It offers an opportunity to examine the effect of various model parameters, namely the effect of changing the step-size, the total number of particles and the viscosity parameters. A simulation is shown that demonstrates the effect of stacking in low shear flows.

4.5.1 Initial State

The initial state for the majority of the seiching basin simulations (excluding the simulations with higher numbers of particles and the one which demonstrates stacking) are shown in Figure 4.16. The initial velocities are set to zero and VMOSS was used to ensure a smooth initial state.

There are 502 particles, each with mass 2465 kg, and the smoothing is typically 1 m. The standard density of water was chosen to be $1000\,\mathrm{kgm}^{-3}$ and the speed of sound was selected as 150 ms^{-1} which corresponds to a maximum Mach number of just under 0.1. Ghost point boundaries were used with the Lennard-Jones type potential of Monaghan (1994). This simulation would have been better performed with the normal/tangential boundaries but the author had not implemented that code when these simulations were created.

4.5.2 Typical results

The particle positions for the seiching basin for a typical simulation is shown in Figures 4.17 to 4.19 for a number of time-steps. Figure 4.20 contains a plot of the energies in a typical seiching basin simulation. The seiche is clear in both the energy and position plots.

It can be seen in Figure 4.20 that the net total energy increases unphysically with time. This is the effect of the accumulation of time-stepping errors and is discussed further in the following section. As the simulation progresses, the noise increases. This is due in part to the lack of artificial viscosity which acts to damp out these energy modes. Control of noise is discussed below.

Early in the simulation, as can be seen in Figure 4.20, the gravitational potential energy drops sharply. This is due to clustering, since several pairs of particles have formed clusters, leaving

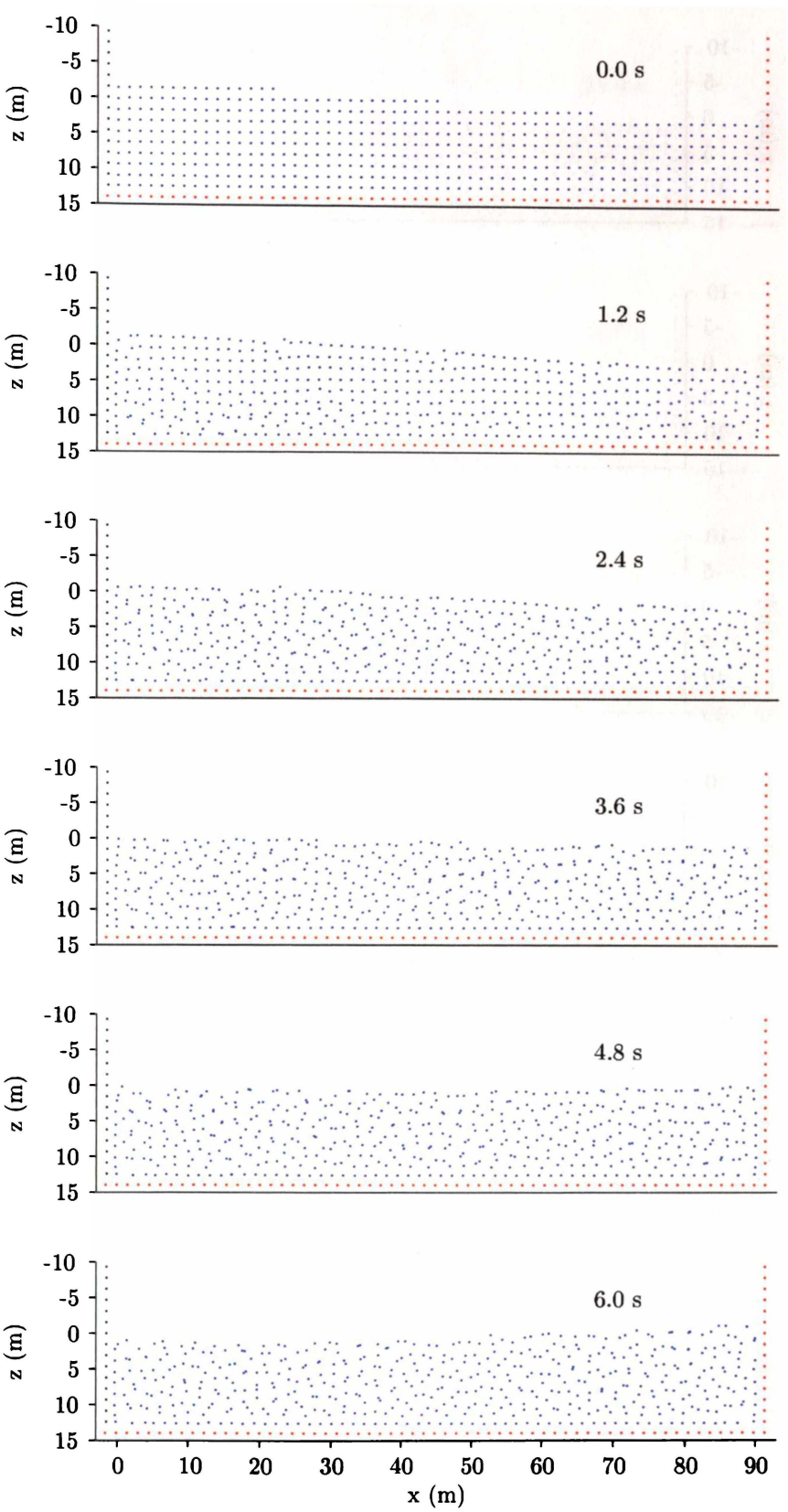

Figure 4.17: *Particle positions during a typical seiching basin simulation for time-steps from 0 s to 6 s. Note the reduction in the amplitude of the seiche. See the following two figures for later times.*

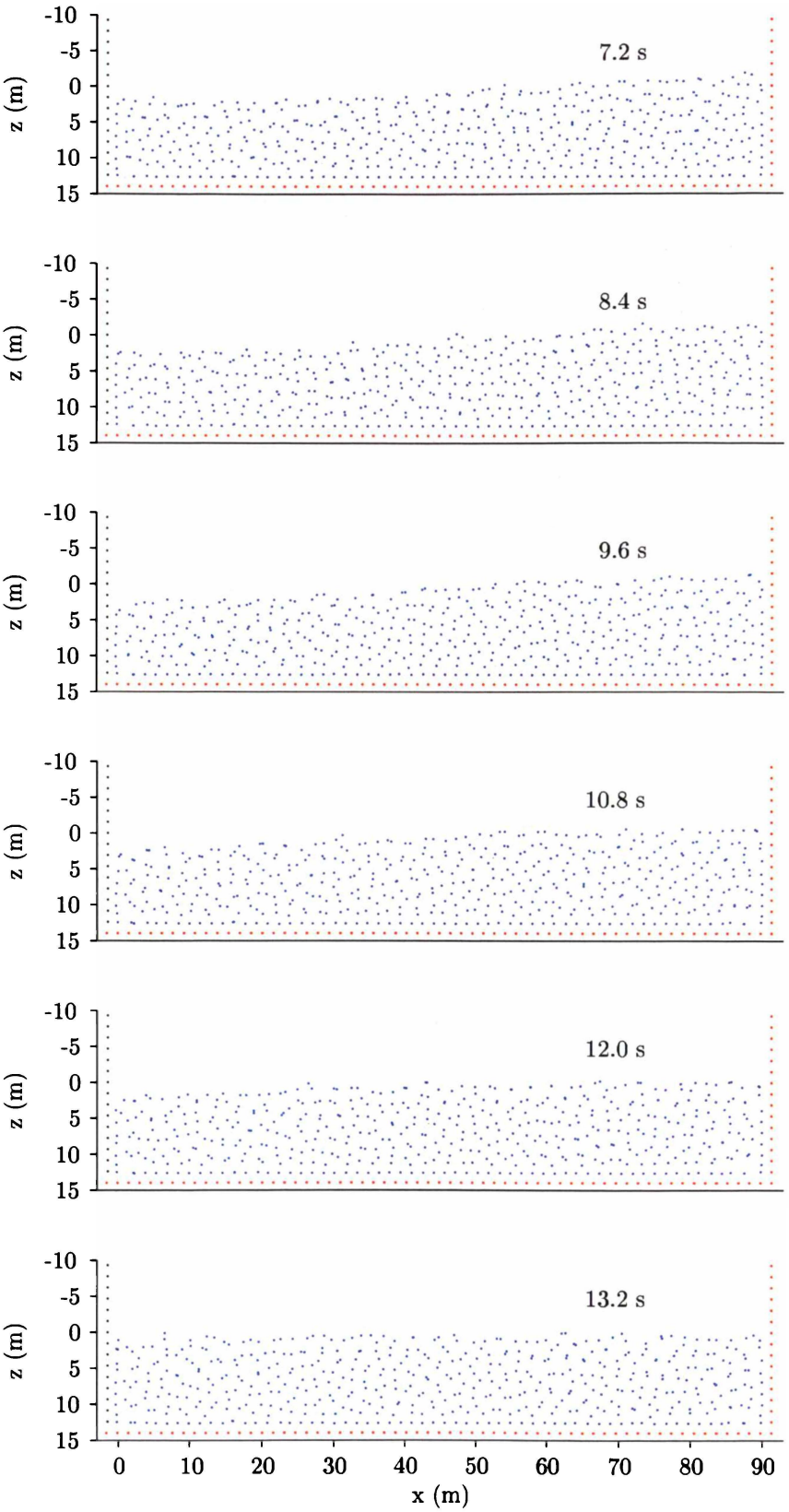

Figure 4.18: *Particle positions during a typical seiching basin simulation for time-steps from 7.2 s to 13.2 s.*

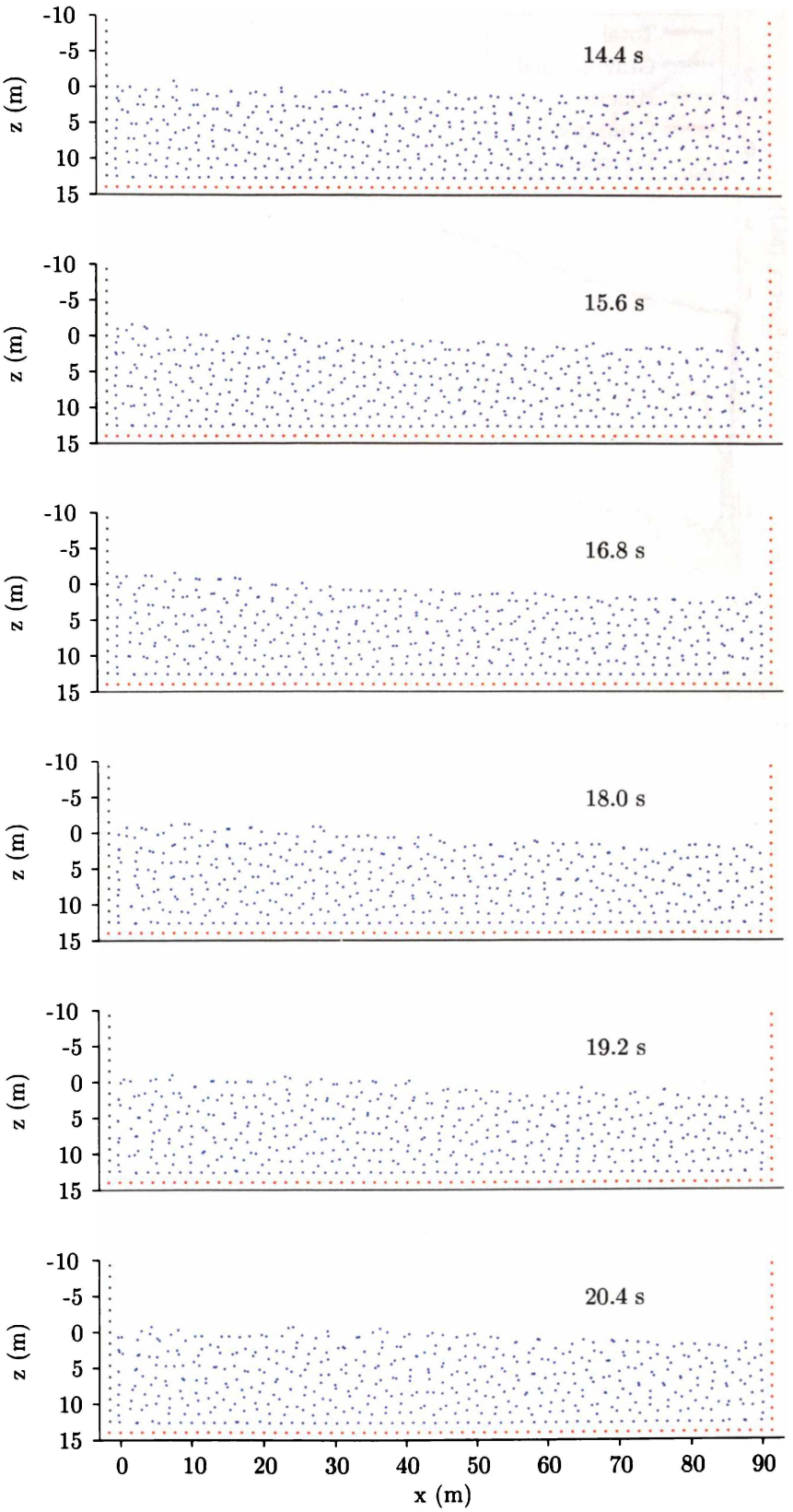

Figure 4.19: *Particle positions during a typical seiching basin simulation for time-steps from 14.4 s to 20.4 s.*

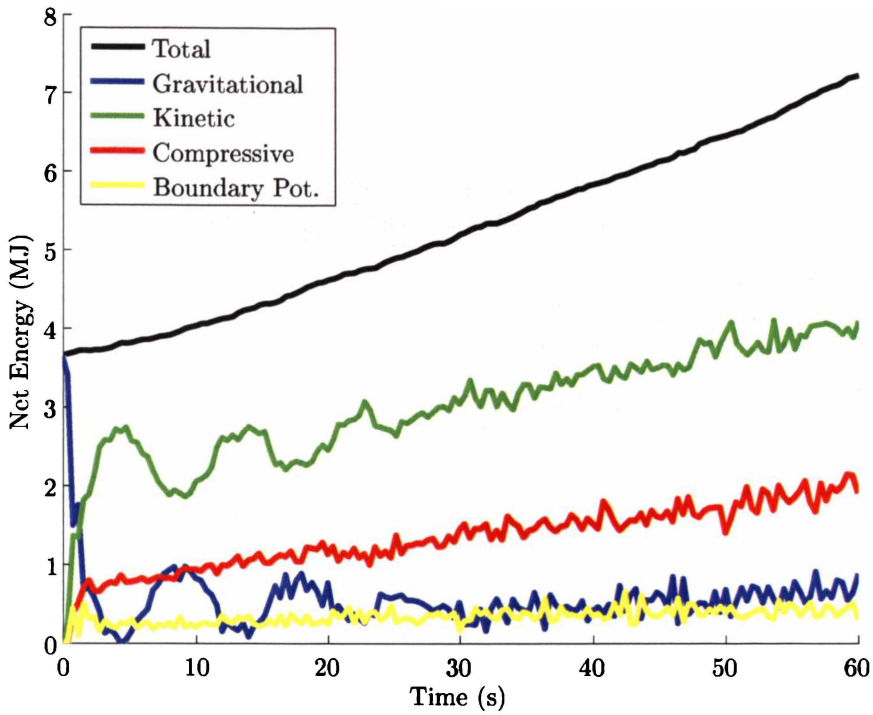

Figure 4.20: *Energies during a typical seiching Basin simulation. Note the increase in the net total energy of the system, the increase in noise and the reduction in the amplitude of the seiche.*

behind regions of low density into which the water column falls. Particles experiencing spurious transport to local crystallised states may also contribute a small amount to this drop. The falling water column represents the transformation of gravitational potential energy to kinetic energy which then appears as noise.

The noise is caused by numerical errors, by sound waves, and by the smoothing that is fundamental to SPH. The sound waves are a result of treating the water as a compressible liquid. When a kernel estimate is made, the property is smoothed over the support, so the momentum for a particle is divided (according to the kernel) between all other particles within a smoothing length from the particle.

The amplitude of the seiche reduces with time, which can be attributed to a combination of two effects: the generation of errors, and dispersion of the wave. For this simulation the water depth is about 12.4 m, the wave (approximately sinusoidal) has a height of 3 m from crest to trough, and the wavelength is around 90 m. For this wave in this basin it should not be expected that linear wave theory should apply (the wave is too high and too short), so the wave is not expected to be permanent. It is most likely that dispersion will account for the reduction in height: the wave splits into constitutive waves that travel at different velocities. The errors include those due to the summation interpolant, the timestepping method, neglected terms in the SPH equations (see Section 3.2.2) and rounding errors.

4.5.3 Time-Stepping in the Seiching Basin

The accuracy of the time-stepping method used here can be improved in two ways: by reducing the step-size and by increasing the number of repetitions of the corrector stage. Figure 4.21 contains plots of the net total energy for a number of seiching basin simulations, each with a different time-step.

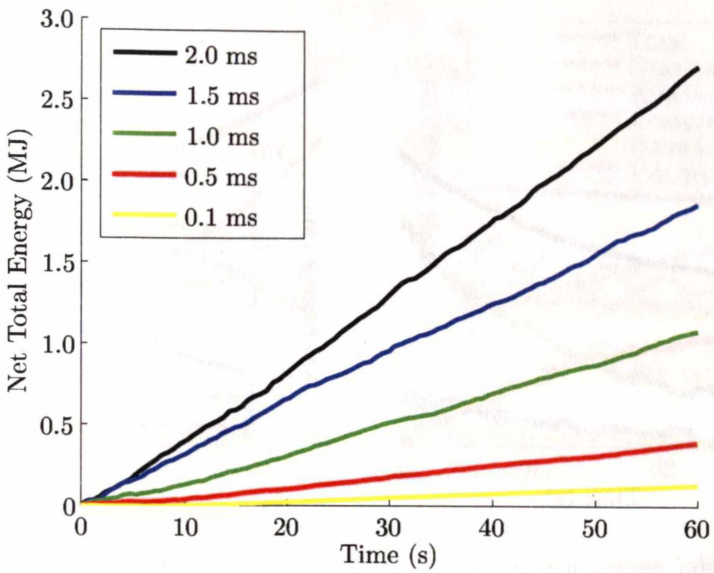

Figure 4.21: *Net total energy growth for different step-sizes for the seiching basin.*

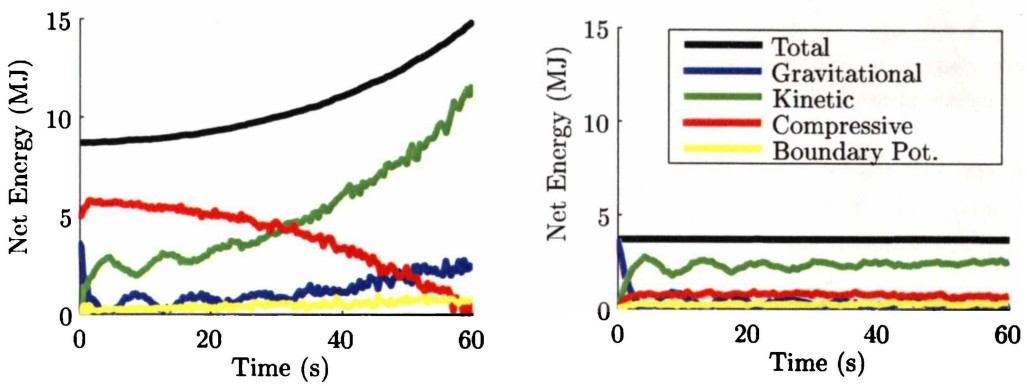

Figure 4.22: *Energies for the seiching basin with a long time-step (2 ms) on the left, and a short time-step (0.1 ms) on the right.*

It is clear that reducing the time-step means the rate at which net total energy grows reduces markedly, and means that noise in the system is generated at a lower rate. With less noise in the system the seiche is more easily distinguished but it is not clear that the seiche attenuates any slower.

4.5.4 Changing the Viscosity Parameters

Increasing the viscosity serves to damp the noise in the system. A number of simulations were run with different values for α, and for these simulations the gravitational potential energy is plotted in Figure 4.24. It can be seen that not only does the viscosity damp the noise, but also the seiche itself if α is chosen too large.

4.5.5 Changing the Number of particles

Increasing the number of particles means the accuracy of the quasi Monte-Carlo estimates increases, but only if the smoothing length is well chosen. Figure 4.25 shows the energies for

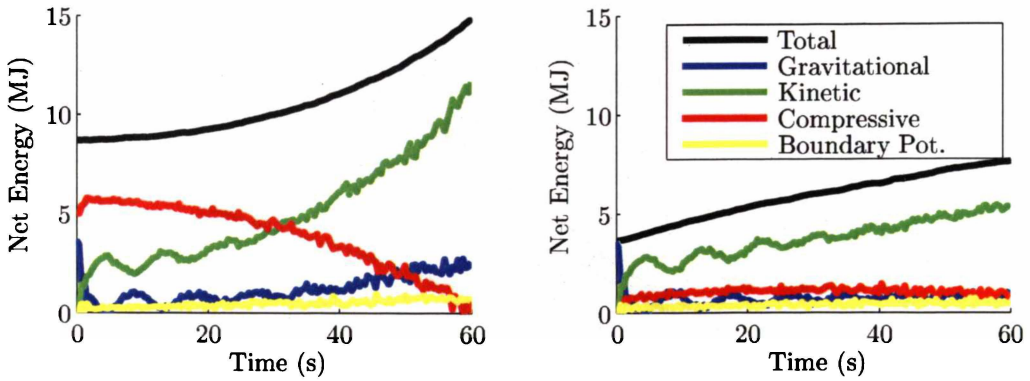

Figure 4.23: *Net total energy growth for one repetition of the corrector stage on the left, and for four on the right, for the seiching basin.*

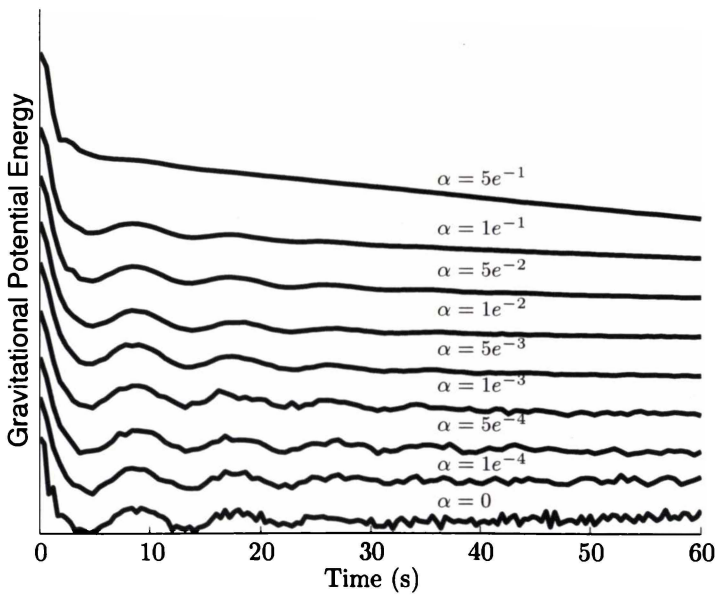

Figure 4.24: *The effect of viscosity on the gravitational potential energy for the seiching basin. The transition from over-damped with $\alpha = 0.5$ to under-damped for $\alpha = 0$ is clear. To each series a constant has been added, otherwise all curves lie over one another and are indistinguishable.*

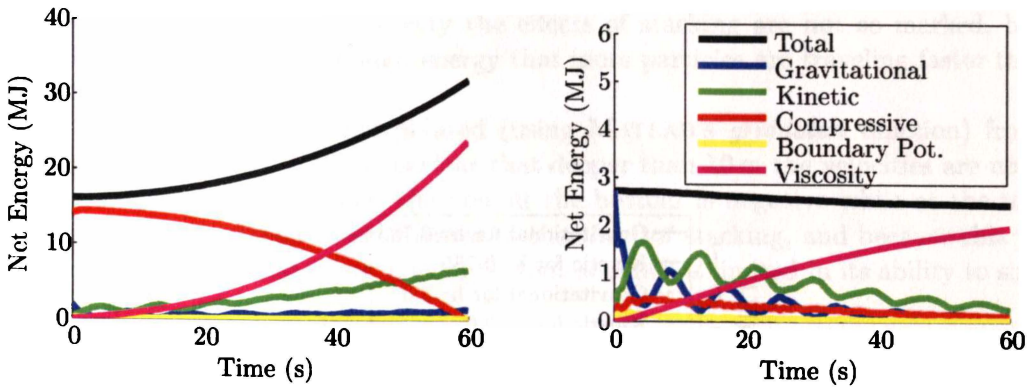

Figure 4.25: *Energies for the seiching basin using 2994 particles and smoothing lengths of 0.785 m on the left, and 2.0 m on the right. Note the different vertical scales.*

simulations with the same physical size as the previous ones, but with 2994 particles, each of 394 kg, and smoothing lengths of 0.785 m and 2.0 m. Both simulations ran for 120 000 time-steps with a time-step of 0.5 ms, velocity of sound 225 m/s and a viscosity of 5×10^{-3}. VMOSS was run for 20 000 time-steps and at every tenth step velocities were set to zero.

Most notably in Figure 4.25 is that the net total energy growth is high for the small h case. Most of this energy is appearing as a growth in kinetic energy which is then dissipated by viscosity. For the small h simulation there is a notable decrease in the compressive energy, a feature that often occurs in poor simulations.

The gravitational potential and kinetic energies in Figure 4.25 have details that are drowned out for the small h case, and hence these energies are plotted in Figure 4.26. The growth in the kinetic energy for the small h is clear, and the only other difference is the amount of noise, which is greater for the small h case.

Artificial clustering occurs in both simulations, evidenced by the sharp drop in the gravitational potential energies shown in Figure 4.26. However it is far worse in the $h = 2$ m simulation, for which a plot of positions is shown in Figure 4.27. This is not too surprising since the position and magnitude of maximum repulsion get further away and smaller as the smoothing length gets larger.

Notice the similarity of Figures 4.22 and 4.25. A shorter smoothing length means that the Courant time-step is shorter, but both of the simulations here use the same step-size, and so the ratio of step-size to Courant time-step is smaller for the $h = 2$ m case.

In comparison to simulations using fewer particles, far more oscillations occur here, so using more particles does provide a better solution, but care must be taken (as always) in choosing simulation parameters in order to achieve this improvement. Notably, the typical low resolution simulation took around 33 minutes, whereas the $h = 0.875$ m and $h = 2$ m took 1284 minutes and 2386 minutes respectively. Some sections of code in MARIAN take longer for larger numbers of neighbours, which explains why the $h = 2$ m simulation took longer.

4.5.6 Stacking in the Seiching Basin

The seiching basin simulation was recreated with particles initially placed on a triangular grid, rather than the rectangular grid typically used. The aim of this simulation is to demonstrate the effect of stacking on an attempt to simulate a flow with a region of low shear, where the particles are very likely to crystallise. For this simulation the boundaries are very rough and it is no surprise that a marked boundary layer occurs in the fluid, visible in Figures 4.28 to 4.30.

This simulation was run with viscosity turned on (using $\alpha = 0.001$) in order that noise in

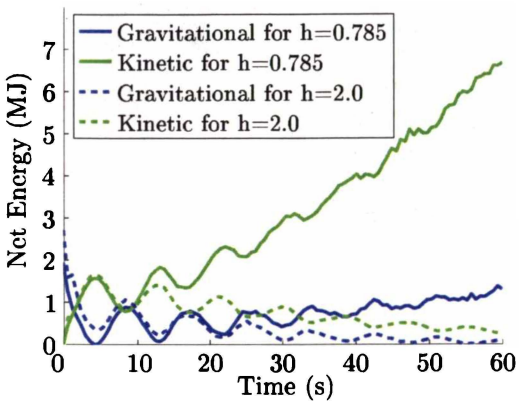

Figure 4.26: *Gravitational potential and kinetic energies for the seiching basin using 2994 particles and smoothing lengths of 0.875 m and 2.0 m.*

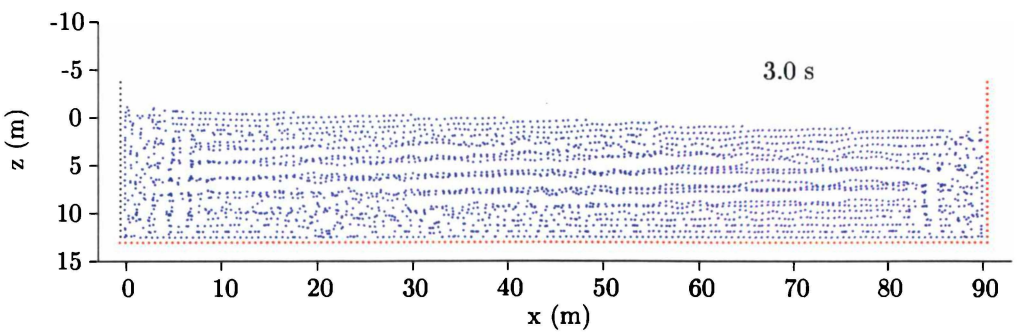

Figure 4.27: *Artificial clustering in the seiching basin using 2994 particles and a smoothing length of 2 m. The horizontal features are the result of artificial clustering.*

the system be reduced. Without viscosity the effects of stacking are not so marked, because the noise in the system provides enough energy that more particles are traveling faster than the critical velocity for crystallisation.

A plot of horizontal velocity interpolated (using MATLAB's ***griddatan*** function) from this simulation is shown in Figure 4.31. It is clear that deeper than 10 m the velocities are not what one would expect, and at some times the flow at the bottom is negative while at the top it is positive. The author attributes these features to the effect of stacking, and because this part of the flow is characterised by low shear, concludes that MARIAN is limited in its ability to simulate low-shear flow.

4.5.7 The Seiche Period

Merian's formula is an analytic expression for the periods of the modes in a seiching basin. In this section the period from a MARIAN simulation of the seiching basin is compared to the period obtained from Merian's formula.

The Period from Merian's Formula

Merian's formula for the period of a seiche in a rectangular basin is

$$T = \frac{2L}{n\sqrt{gH}} \tag{4.25}$$

where L is the length of the basin, g is acceleration due to gravity, H is the water depth and n indicates which mode we are interested in. It should be noted that Merian's formula derives from the dispersion relation for the basin and requires that the water be shallow so that $\tanh(n\pi H/L) \approx n\pi H/L$, which is true for the basin here to within 7.5%. The uncertainty in the period (ΔT) is given by

$$\Delta T = T\sqrt{\left(\frac{\Delta L}{L}\right)^2 + \left(\frac{\Delta H}{2H}\right)^2}. \tag{4.26}$$

It is not clear exactly what values H and L should take here. In SPH the left and right hand limits of the water can be taken as being the length of the basin but these are not defined exactly. It has been noted that in SPH the limits of a fluid are known to lie within the support of the outer most particle, but when we're examining the water column next to stationary boundaries an estimate using this approach is not correct. Instead we can observe that the limit of the water must lie between the outer-most particle and the position of the appropriate boundary points. Using these ideas with the 2994 point simulation we obtain

$$L \pm \Delta L = 90.63 \pm 0.51 \tag{4.27}$$

$$H \pm \Delta H = 13.78 \pm 1.24 \tag{4.28}$$

where the uncertainty in the position of free surface (used to find H) is dominated by the smoothing length, rather than the particle separation. The period of the nth mode is $15.59/n \pm 0.71/n$ s.

4.5.8 The Period from the SPH Simulation

For the seiching basin simulation with 2994 particles we can obtain the period of the oscillation by examining the net gravitational potential energy for the system. The left hand plot in Figure 4.32 shows the net gravitational potential energy for this simulation. Note that the time between

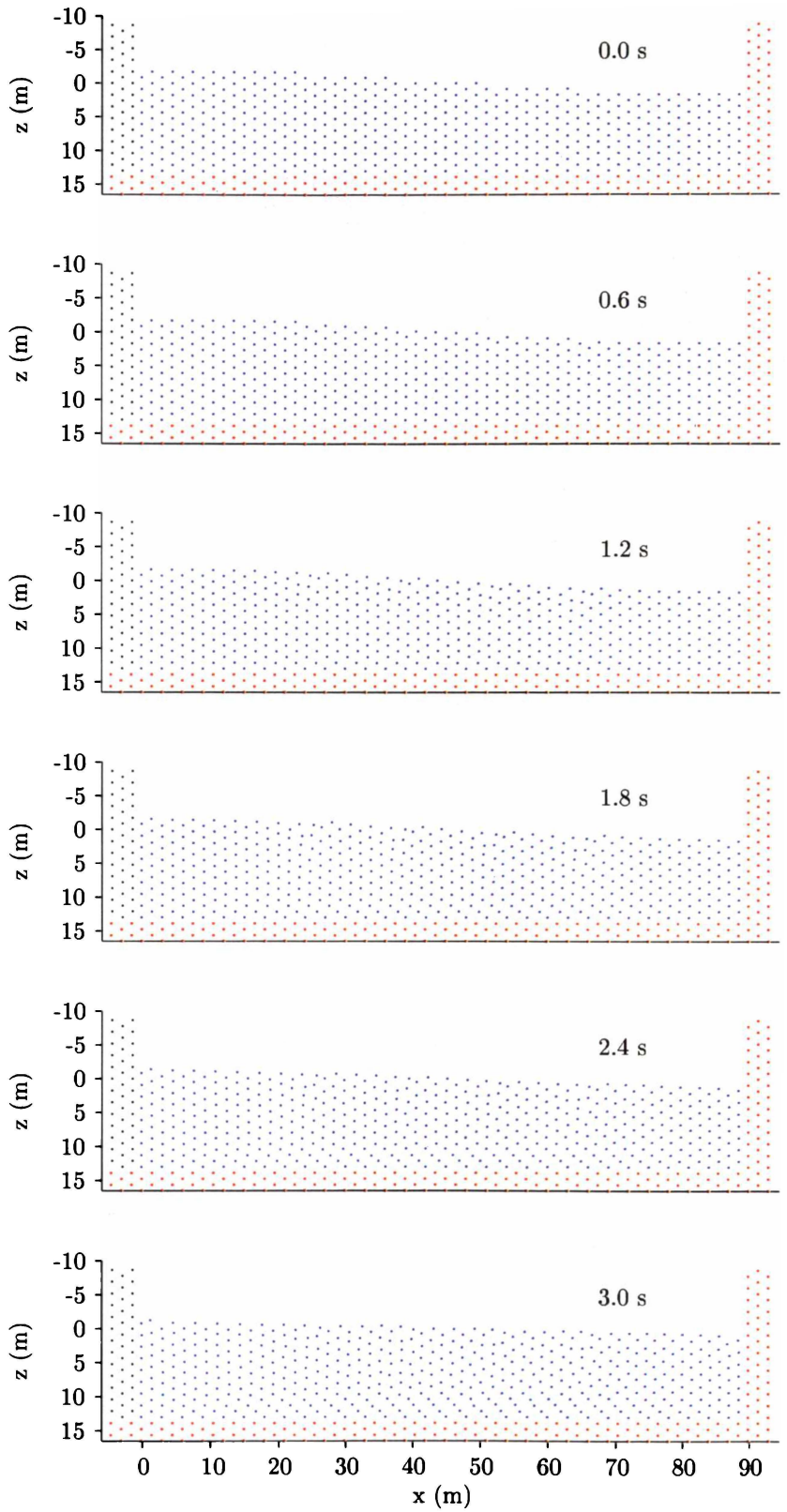

Figure 4.28: *Particle positions during a seiching basin simulation with particles initially arranged in a triangular grid for time-steps from 0 s to 3 s. Note the boundary layer, attributed by the author to particle stacking. See also the following two figures.*

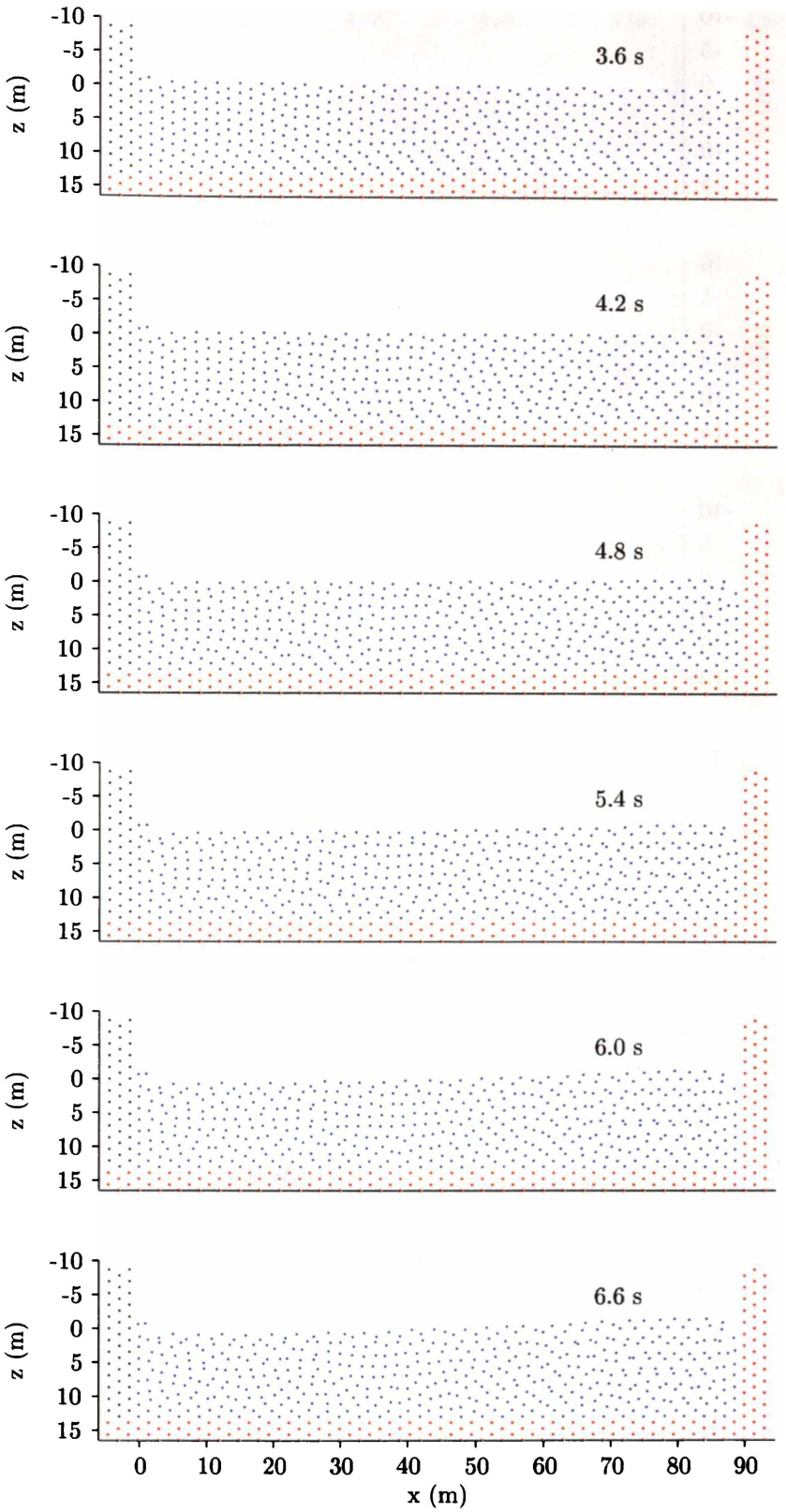

Figure 4.29: *Particle positions during a seiching basin simulation with particles initially arranged in a triangular grid for time-steps from 3.6 s to 6.6 s. Note the boundary layer, attributed by the author to particle stacking. See also the previous and the following figures.*

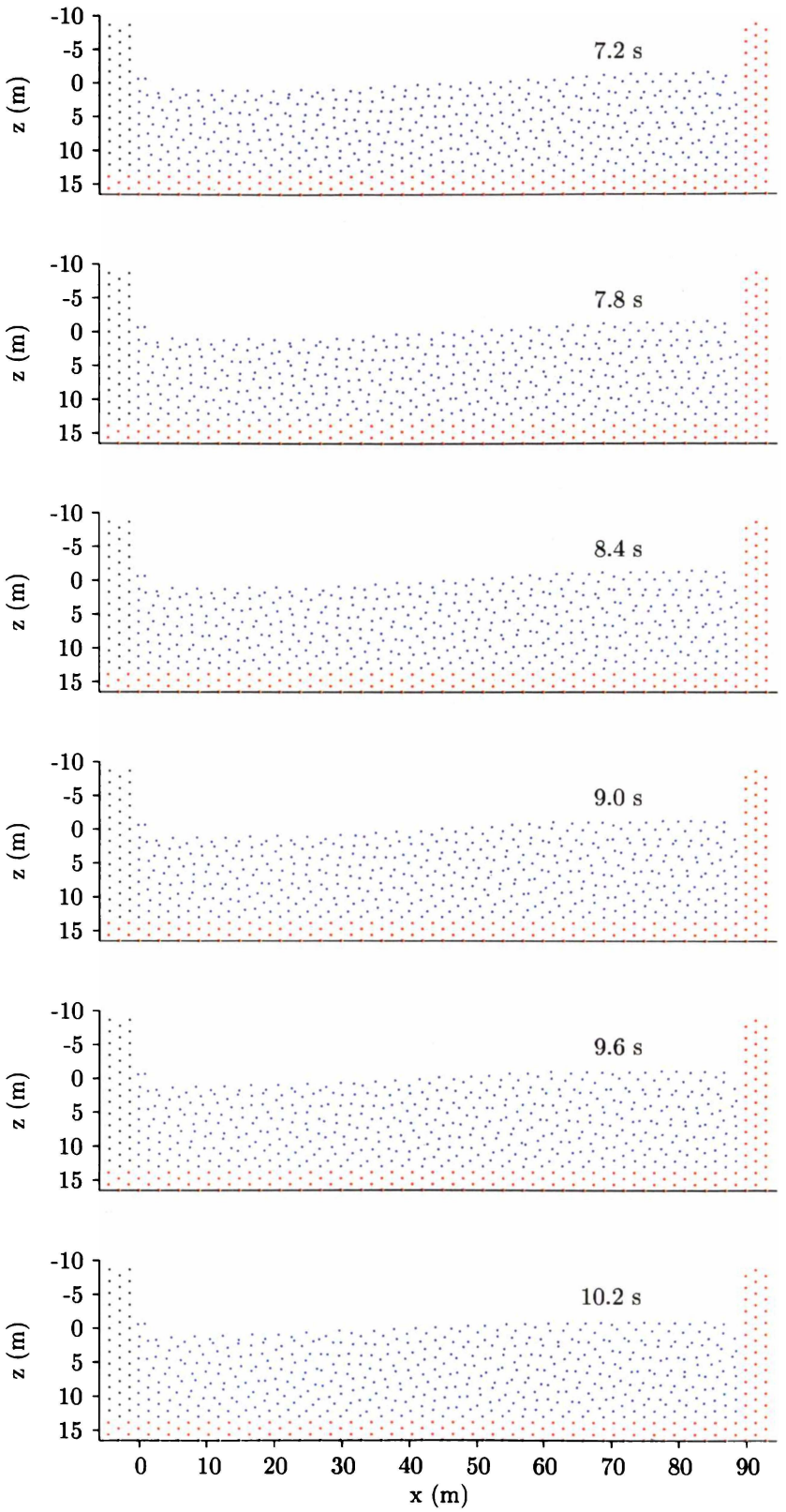

Figure 4.30: *Particle positions during a seiching basin simulation with particles initially arranged in a triangular grid for time-steps from 7.2 s to 9.2 s. Note the boundary layer, attributed by the author to particle stacking. See also the previous two figures.*

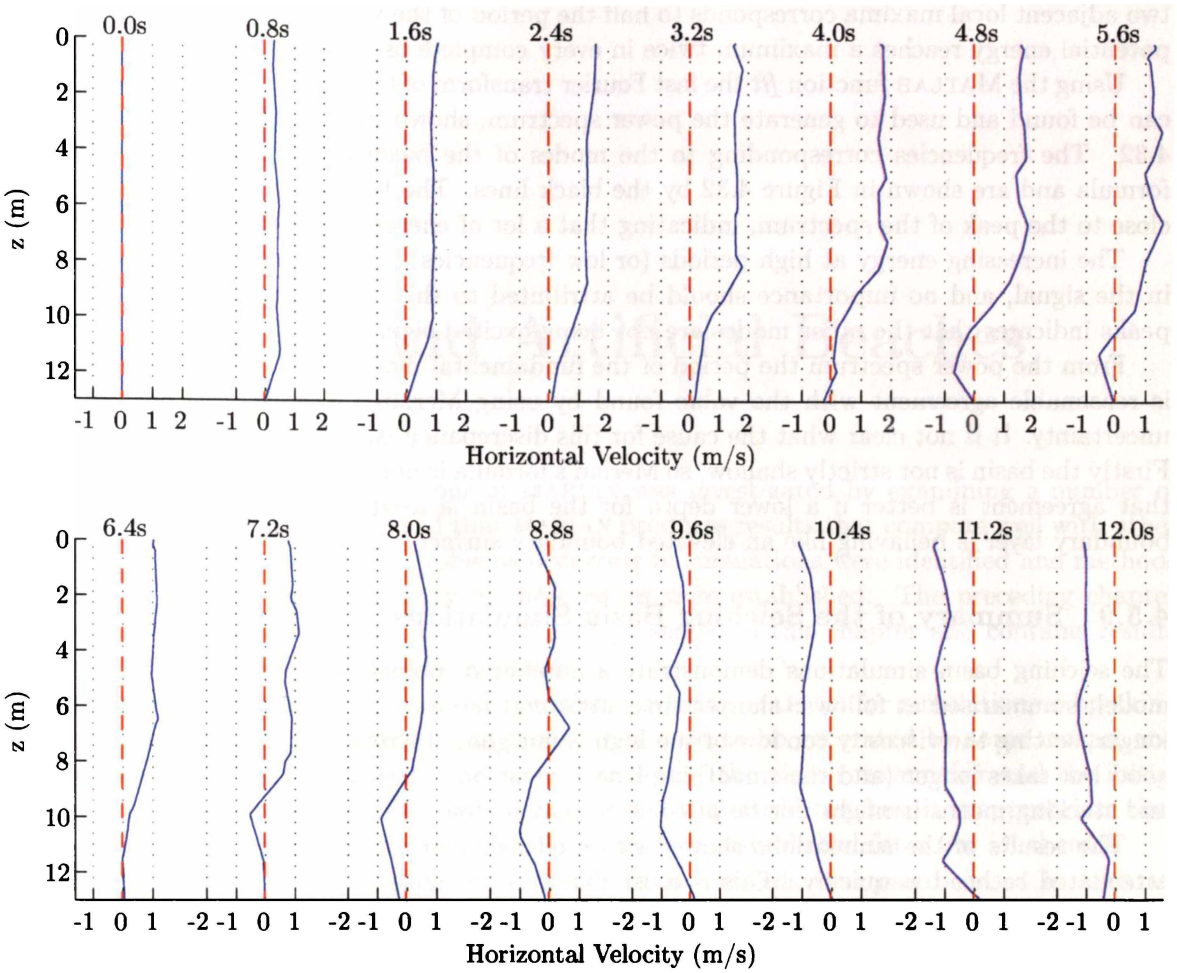

Figure 4.31: *Horizontal velocities at the centre of the basin interpolated from particle velocities for the triangularly distributed simulation. For each time the velocity curve (in blue) has been shifted along the x-axis, and the dashed red line represents the zero velocity line for each time. The time for each curve is written beside the zero velocity line. The velocity curves do not show expected behaviour, attributed by the author to particle stacking.*

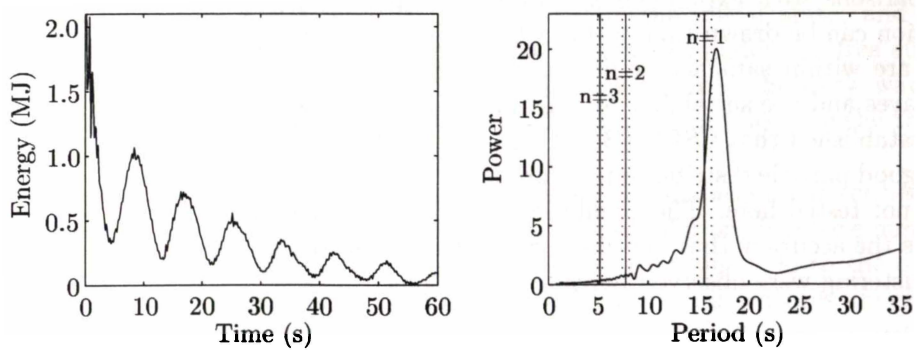

Figure 4.32: *The net gravitational potential energy for the seiching basin with 2994 points in the left plot. The time between two adjacent maxima is half a seiche period. The right plot shows the spectrum obtained from the gravitational potential energy using a fast Fourier transform. The full vertical lines are the periods for the first three modes predicted from Merian's formula and the dotted lines show the associated uncertainties.*

two adjacent local maxima corresponds to half the period of the wave, since the net gravitational potential energy reaches a maximum twice in every complete oscillation.

Using the MATLAB function *fft* the fast Fourier transform of the gravitational potential energy can be found and used to generate the power spectrum, shown in the right hand plot in Figure 4.32. The frequencies corresponding to the modes of the basin are evaluated using Merian's formula and are shown in Figure 4.32 by the black lines. The fundamental mode is reasonably close to the peak of the spectrum, indicating that a lot of energy is present in that mode.

The increasing energy at high periods (or low frequencies) is a result of the downward trend in the signal, and no importance should be attributed to this. The lack of obvious secondary peaks indicates that the other modes are not being excited significantly.

From the power spectrum the period of the fundamental mode is found to be $16.75\,s$. There is reasonable agreement with the value found by using Merian's formula, but not within the uncertainty. It is not clear what the cause for this discrepancy is, but the are two likely sources. Firstly the basin is not strictly shallow, so Merian's formula is not exact. Secondly, it is significant that agreement is better if a lower depth for the basin is used. It is quite possible that the boundary layer is behaving like an elevated boundary surface.

4.5.9 Summary of the Seiching Basin Simulations

The seiching basin simulations demonstrate a number of points that are expected in an SPH model, summarised as follows: shorter time-steps and more corrector steps are better but take longer, setting the viscosity too low or too high is not good, increasing the number of particles is good but takes longer (and the smoothing length must be chosen well to get the better results), and stacking means it might not be possible to resolve flows with low shear.

The results of the simulations showed expected behviour, except that the seiche amplitude attenuated rather too quickly. This is most likely to be caused by the accumulation of errors and by dispersion. The errors include those due to the summation interpolant, the timestepping method, neglected terms in the SPH equations (see Section 3.2.2) and rounding errors. The viscosity formulation used is also a significant contributor to this problem and is discussed in Section 6.3.4. On the positive side, the seiche period was reasonably close to that predicted by Merian's formula.

4.6 Conclusion

After comparisons with experimental, computational and theoretical data from the literature the conclusion can be drawn that MARIAN is running without significant programming mistakes, and errors are within satisfactory limits. Consequently MARIAN can now be used to examine breaking waves and the suitability for this task can be ascertained.

It was established that CSPH (Bonet and Lok, 1999) is characterised by poor energy conservation but good particle distributions. More recent developments to CSPH may improve this but these were not tested here. The simulations performed here demonstrated the role of stacking, which limits the accuracy that low-shear flows can be resolved to. Significant features caused by artificial clustering were observed in some simulations.

Chapter 5

Ideal Waves and Artificial Beaches

In the previous chapter the behaviour of MARIAN was investigated by examining a number of simple simulations. It was confirmed that MARIAN produces results that compare well with other SPH models from the literature. Problems occurring in simulations were identified and methods for avoiding or reducing the severity of these issues were established. The preceding chapter provides an important basis for the work that is presented in this chapter and contains results that indicate how simulations can be improved.

The aim of this chapter is to identify problems in using MARIAN for simulation of breaking waves on beaches. Before the waves can break, they need to be generated by some means, and the two methods that can currently be used in MARIAN (the piston wavemaker and the initial state method) are investigated. Before being applied to real-world beaches, it is appropriate that the behaviour of MARIAN in simulating breaking waves be established for a few ideal cases.

Here MARIAN is used to study wavemaking. Both methods that can be used in MARIAN to make waves are used to generate a range of solitary waves in a numerical wave tank, and the nature of the waves is examined. Cnoidal waves are generated in the numerical wave tank using the initial state method. Finally six simulations are conducted, in each of which the same solitary wave is created (by the initial state method). The wave tanks are modified to include plane-sloped beaches, with different slopes for each simulation.

5.1 Solitary Waves by the Initial State Method

One method for generating waves in SPH is by specifying the initial state, and in MARIAN this requires determining the velocities and pressures of particles under the free surface. This information can be found by using expressions from the literature. Solitary wave theory is sometimes used to approximate periodic waves in shallow water, and because the theory is well described in the literature, it is used here.

In this section the solitary wave solutions used to specify initial states are described, and seven simulations performed using these states are examined. The simulations involve a numerical wave tank, a simple rectangular basin in which waves propagate from one side to the other. These simulations are described quantitatively, and the causes of particular features are examined.

Conclusions are drawn with regard to the suitability of the method, and suggestions given as to how the model could be improved.

5.1.1 Solitary Wave Solutions

Both Monaghan and Kos (1999) and Dalrymple and Knio (2001) used the initial state method to generate solitary waves in their SPH models. It would be appropriate to use the same methods,

however Dalrymple and Knio (2001) do not state what theory they used, and the equations given by Monaghan and Kos (1999) are only first order.

A number of solitary wave solutions have been expressed in the literature, including a second order solution (Dean and Dalrymple, 1991), a third order solution (Grimshaw, 1971) and a ninth order solution (Fenton, 1972). Here the *order* of the solution indicates the highest power of α present in the solution, where $\alpha = H/h$, for H the wave height and h the water depth. The higher α is for a particular wave, the higher the order of the solution must be to provide a good approximation.

Here the second order solution of Dean and Dalrymple (1991) is used. It was chosen because it is still relatively simple (in comparison to the higher order solutions) and because it offers easily achieved improvements over the first order solution.

A Second Order Solitary Wave Solution

The free surface is given by (Dean and Dalrymple, 1991)

$$\eta(x) = H\text{sech}^2\left(\sqrt{\frac{3}{4}\frac{H}{h^3}}x\right) \tag{5.1}$$

in which the coordinate system moves with the celerity of the wave. The horizontal and vertical components of the velocity by

$$u(x,z) = \frac{C}{\sqrt{gh}}\left[\left(\alpha + \tfrac{1}{2}\tfrac{\alpha^2}{h^2}\left(h^2 - 3z^2\right)\right)\frac{\eta}{H} - \tfrac{1}{4}\tfrac{\alpha^2}{h^2}\left(7h^2 - 9z^2\right)\frac{\eta^2}{H^2}\right] \tag{5.2}$$

$$w(x,z) = C\sqrt{3\alpha}\tfrac{z}{h}\tfrac{\eta}{h}\tanh\left(\sqrt{\tfrac{3}{4}\tfrac{H}{h^3}}x\right)\left[1 + \tfrac{\alpha}{2}\left(1 - 7\tfrac{\eta}{H} - \tfrac{z^2}{h^2}\left(1 - 3\tfrac{\eta}{H}\right)\right)\right]. \tag{5.3}$$

In this section the vertical coordinate (z) is positive upwards and zero at mean sea level. The wave celerity is given by

$$C = \sqrt{\frac{gh}{(1-\alpha)}}. \tag{5.4}$$

The pressure is hydrostatic, so

$$P(x,z) = \rho g\left(\eta(x) - z\right). \tag{5.5}$$

5.1.2 Seven Solitary Waves

The solitary wave solution described in the previous section is used to create seven different waves in a numerical wave tank. The wave tank is the same for each wave, and the waves differ only in the value of α, which takes the values 0.1, 0.2, ... 0.7. The wave tank is 0.3 m deep, and 14 m long. The wave peaks are all initially at $x = 3$ m, chosen so that for all of the waves η is less than 1 % of H at the left hand vertical boundary. Details of examples initial states are shown in Figures 5.1 and 5.2.

All simulations used a time-step of 0.25 ms, requiring 18 000 time-steps, although fewer could have been used for the larger waves because they travel faster. The speed of sound was chosen to be 25 m/s, so the highest expected velocity has a Mach number slightly lower than 0.1. The simulations had in the region of 10 000 particles (the exact amount varies with wave height) each 0.4 kg in mass, initially placed on a rectangular grid with a spacing of 0.02 m. The smoothing length is 25 mm and so a particle has approximately 20 neighbours. The standard density is 1000 kg/m^3 and viscosity was not used. Normal/tangential boundaries are used, chosen to avoid corrugations. Neither the XSPH nor the CSPH variants are employed.

Some noise is expected in simulations since there is no viscosity, and energies are less useful since boundary potential and total energies are not defined. It might be anticipated that the

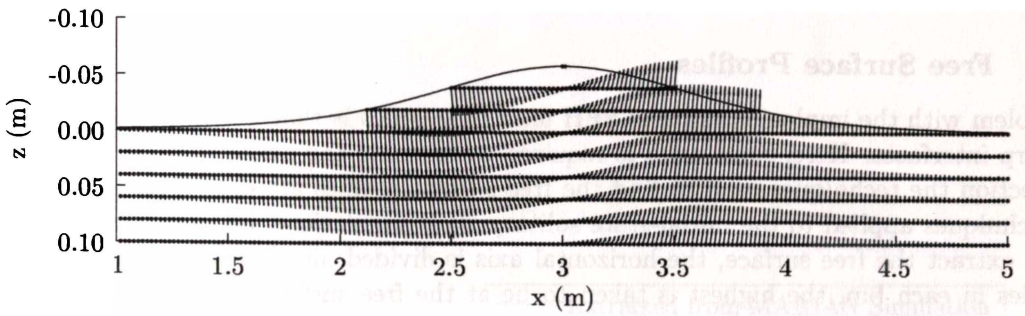

Figure 5.1: *Detail of the initial state for $\alpha = 0.2$ in which the dots are particle positions and the arrows are velocities. The curve shows $\eta(x)$, and the vectors are velocities. Note that different scales used are on the vertical and horizontal axes, and that only the top third of the wave tank is shown.*

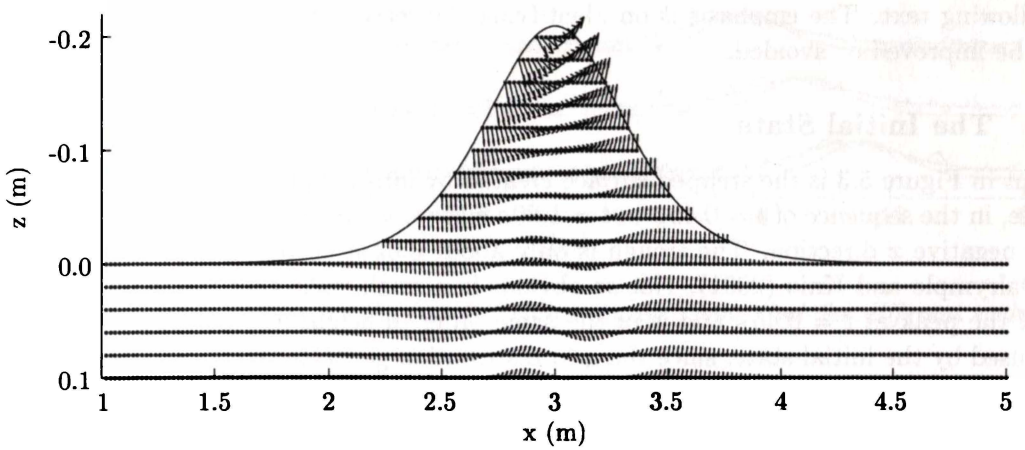

Figure 5.2: *Detail of the initial state for $\alpha = 0.7$ in which the dots are particle positions and the arrows are velocities. Note the change in the direction of particle velocities, and that the vertical and horizontal axes have different scales. The velocity vectors are drawn to a different scale from those in Figure 5.1.*

smaller waves (with an α of 0.1 and 0.2) will be difficult to resolve since the waves are only a little larger than the initial particle spacing. However the low order of the solution used here means the higher waves (with an α from 0.4 to 0.7) are poor approximations to solitary waves. This is evident in Figure 5.2, in which velocities are seen to change orientations three times along some rows, which should occur only once.

5.1.3 Free Surface Profiles

A problem with the implementation of SPH used in MARIAN is that free surfaces are not defined as sharp interfaces. Here, some work is required to estimate the position of the free surface. In this section the techniques used to find the free surface profiles are described, and examples of the techniques applied to the initial-state solitary wave simulations are presented.

To extract the free surface, the horizontal axis is divided into bins of equal size. Of all the particles in each bin, the highest is taken to be at the free surface. Taking the bin width too small means there is a chance bins are empty, while taking it too large means smoothing occurs. Here the bins are 0.1 m wide, five times the initial particle spacing (0.02 m).

For the profile at each time-step, the values A, B, C and D are found that minimise the difference between the equation

$$z = A\mathrm{sech}^2(B(x + C)) + D \tag{5.6}$$

and the profile. The curve generated by this means is plotted with the profile and also with the solitary wave solution shown. Figure 5.3 shows a plot with many axes, one for each of a selection of time-steps, containing the profile extracted by this method.

In general, the wave evolves quickly to become a good approximation to a solitary wave, especially from 0.45 s to 1.35 s. After that the amount of noise increases, the peak widens and reduces in height. The water level rises and the simulated wave travels slower than the second order solution predicts. Briefly, the wave is an approximation to a solitary wave, but not a great one. There are several notable features of Figure 5.3 and these are identified and discussed in the following text. The emphasis is on identifying the reasons for these features, and how they could be improved or avoided.

5.1.4 The Initial State

Evident in Figure 5.3 is the stepped surface created by initially placing particles on a grid. Also notable, in the sequence of $t = 0.45$ s to $t = 1.80$ s curves, is the trough of a wave that propagates in the negative x direction. The trough is only about 0.02 m deep, and is also visible in Figure 2 of Dalrymple and Knio (2001). Not so obvious is a small spike that forms on the right hand side of the peak at $t = 0.45$ s and is of the same order in height as the trough. These features are caused by the initial state, since the grid distribution generates incorrect pressure gradients.

The fitted wave at $t = 0$ s is poor, a consequence of the method used to generate it. The curve was fitted to the highest point in each bin, and the result has an even distribution above and below the curve. In creating the original state only particles below (deeper than) the initial profile are kept.

The effect of boundaries on the simulation is clear, with particles being repulsed at both ends of the wave tank. The consequence of this is the addition of spurious energy to the system, part of which evolves to become noise in positions of the free surface.

5.1.5 Noise

In all of the simulations here noise is a significant feature in the free surface profiles. In every case the amount of noise is small up to a certain time, and then increases markedly. In Figure 5.3

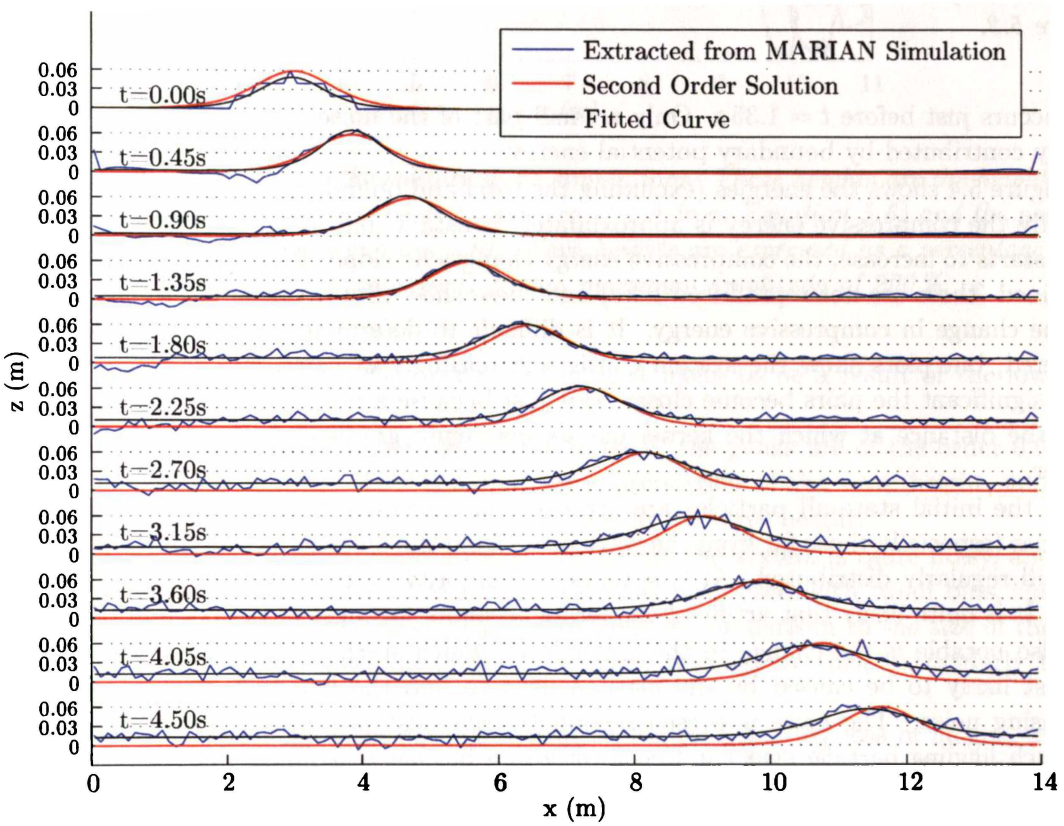

Figure 5.3: *Free surface profiles for the $\alpha = 0.2$ solitary wave. Note that the different scales used on the vertical and horizontal axes.*

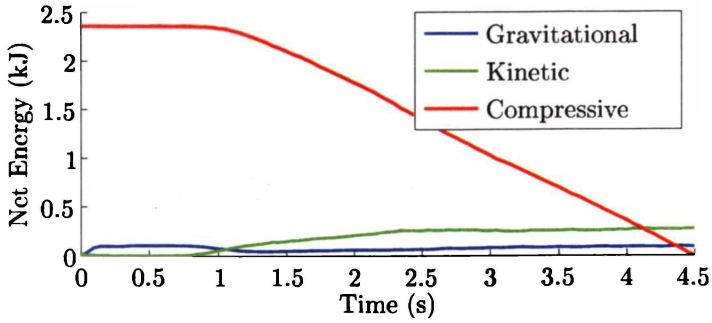

Figure 5.4: *Energies during the $\alpha = 0.2$ solitary wave simulation. Note the change in the slope of the curve for compressive energy, which occurs at the same time as noise starts increasing in Figure 5.3.*

this occurs just before $t = 1.35\,\mathrm{s}$. Only a small part of the noise is attributable to the spurious energy contributed by boundary potential energy.

Figure 5.4 shows the energies (excluding the total and boundary energies) for the simulation. Initially the compressive energy is approximately constant, but at around the same time as the noise starts to increase, the compressive energy changes its slope. Plots of particle positions were examined (these are not reproduced here) and these show that artificial clustering may account for the change in compressive energy. It is difficult to determine exactly which particles have clustered, but plots show the system consists increasingly of pairs, and as clustering becomes more significant the pairs become closer. But just because a pair of particles are closer together than the distance at which the kernel has its maximum gradient does not mean the pair is a cluster.

In the initial state all particles are arranged on a rectangular grid, and at $t = 1.0\,\mathrm{s}$ many particles (especially those far from the wave peak, from the boundaries and from the free surface) are still regularly distributed. At $t = 1.35\,\mathrm{s}$ the majority of all particles form clusters, usually pairs. It is quite likely that artificial clustering accounts for most of the noise in the system.

Also notable is the change in the water level, which starts to increase at $t = 1.35\,\mathrm{s}$. This is most likely to be caused by the method used to find the free surface in concert with the increasing noise. The noise is present in kinetic energies: particles begin to jiggle more, and one such jiggling particle near the free surface in a bin has a significant effect. The spikes that appear in the free surface may consist of a single particle that has been ejected briefly from the main body. It is very likely that the accumulation of numerical errors contribute to the growth of noise here. The use of viscosity would have reduced the rate of growth.

5.1.6 Wave Heights

The wave heights observed in the MARIAN simulations are not constant, as they would be if the waves were truly permanent. Wave heights are plotted in Figure 5.5 against position of the peak for the seven solitary waves simulated with MARIAN.

It is interesting that the waves in Figure 5.5 do not all start with heights 100 % of the specified. This is because the method used to calculate the wave height is poor for small waves when the particles are regularly distributed. This effect is only significant for the $\alpha = 0.1$ and $\alpha = 0.2$ waves.

Early in each simulation there is an increase in height, caused by several effects. The $\alpha = 0.7$ wave shows an increase to around 200 % in part because it breaks, with a jet being propelled upward. The $\alpha = 0.6$ wave starts shoaling, forming a sharp crest that disperses with time. This

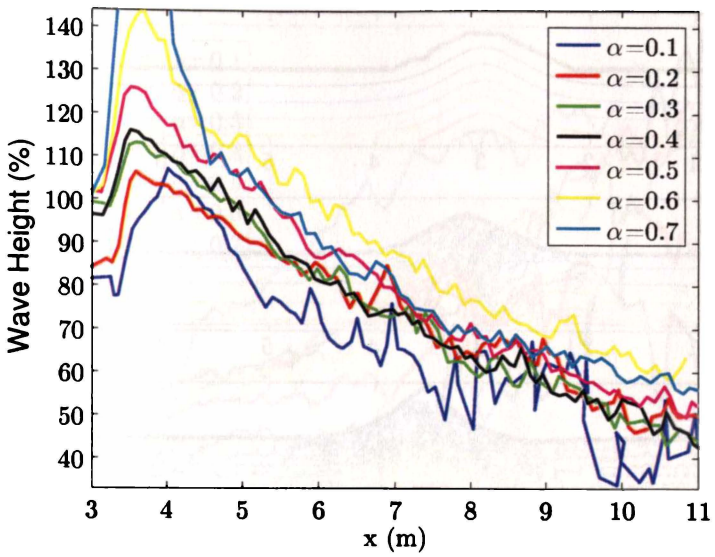

Figure 5.5: *Wave heights during the solitary wave simulations. These heights are the heights from the fitted curves, less the increase in water depth (so $D - A$ from Equation 5.6) and the positions are the peak positions (C from Equation 5.6). Wave heights are expressed as a percentage of the initial wave height. The $\alpha = 0.7$ wave peaks with a maximum amplitude of 200 %.*

sharp crest is not evident for smaller waves and is most likely to be a result of the low order solution used.

The small spikes that form on the right hand flanks of the peaks are of the order 0.02 m in size, the initial particle spacing. This is a significant proportion of the wave height, and when the spikes reach the wave crests they cause an increase in the wave height.

Evident in Figure 5.5 is an increase in noise. Primarily the system is quite noisy, and since the method for determining the free surface elevation has been used with small bins, the noise is not smoothed. Additionally, the wave heights are small in comparison to the size of the noise features, and the percentage of initial height is strongly affected by this, particularly for the $\alpha = 0.1$ wave.

Beyond the first 4.5 m, the most notable feature of Figure 5.5 is the loss in height, which is approximately linear over the distance shown and is very similar for all waves. There are several causes likely to contribute to this loss in height. Numerical errors are a likely source, and this could be established by decreasing the time-step, and increasing the number of particles. It is also likely that artificial clustering and particle stacking contribute, causing spurious transport of energy.

The low order solution used to create the initial state does not explain the height loss here, since there would be an observable improvement for smaller waves if this were the case, which there is not.

The final reason for loss of height lies on the method used in SPH to account for incompressibility. To a very good approximation water is incompressible, but here water is approximated by a compressible fluid, with the compressibility chosen so that density variations stay within 1 % of the standard density. Contour plots of the density are shown in Figure 5.6 for various time-steps. These contour plots are generated by using MATLAB's *griddata* subroutine. Early times show a peak in the density below the peak of the wave, with contour that get progressively noisier with time. But eventually artificial clustering dominates, and the contours plots show high and low densities scattered throughout the fluid. Compressibility in SPH is probably a cause for loss of height, but it is difficult to extract information because of the dominance of

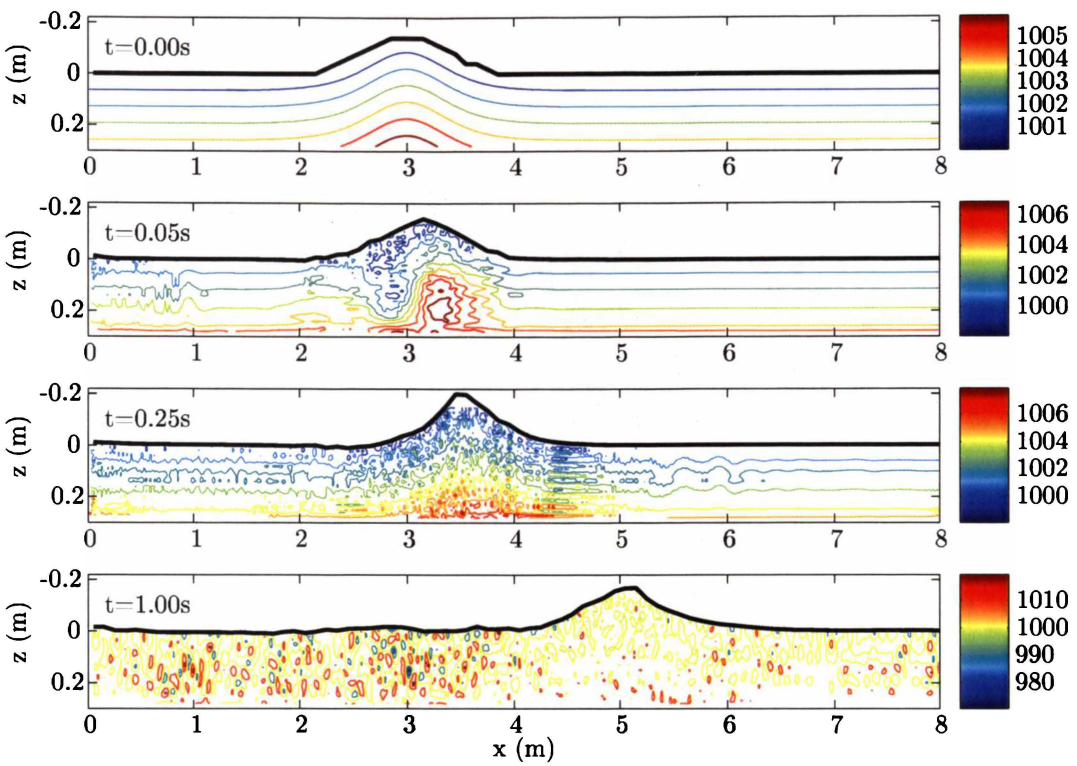

Figure 5.6: *Contour plots of the density during the $\alpha = 0.5$ solitary wave simulation. The black line is the free surface. The density quickly evolves into a state where artificial clustering dominates. Later in the simulation variations in density have grown to around 10 %, far in excess of the 1 % specified.*

clustering. It would be interesting to see whether MPS or PSPH simulations would experience the same loss in height.

5.1.7 Wave Shape

A related issue is the change in the length of the waves generated here. All the simulations of solitary waves created by the initial state method show changes in length, consistent with their not being permanent waves. This dispersion is evident in Figure 5.3.

The length of the solitary wave is determined by the coefficient of the argument in the solution, that is, B in Equation 5.6. For a solitary wave this should be constant, but here it is not. The length of the peak increases as B decreases. In Figure 5.7 the inverse values of B from the fitted equation are plotted for all waves, expressed as percentages of the prescribed values, that is, $100 \times B_s/B$ in which B_s is the prescribed value.

Figure 5.7 shows similar behaviour to Figure 5.5, and explanations for features are the same. The poor performance of the fitted expression at approximating the free surface in early stages for small waves means initial lengths are not 100 %. The early increase in height, while keeping length constant requires an increase in the value of B, explaining the decrease in length early in the simulations. The comparatively low resolution for the smallest wave accounts for its overly low, noisy length. No importance can be attributed to the length of the highest wave, since it breaks.

It is notable that the smallest two waves have lower relative lengths than other waves but it is not necessary to attribute importance to this. The most probable explanation for this feature

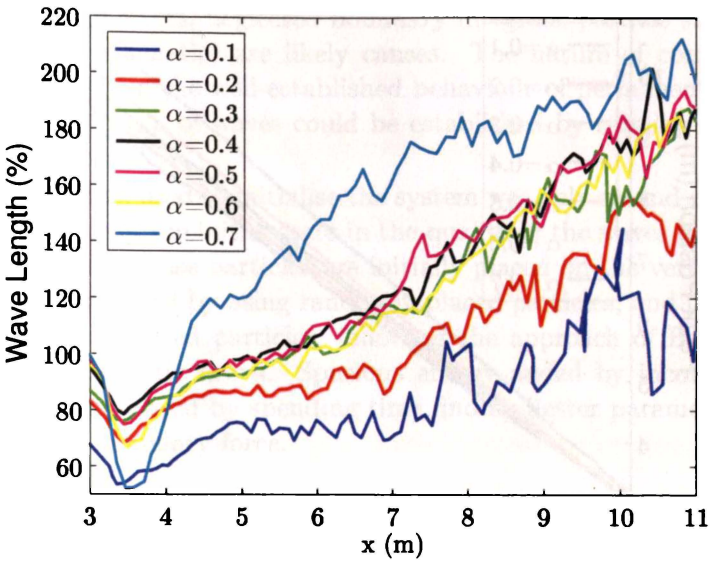

Figure 5.7: *Wave lengths during the solitary wave simulations. These lengths are the inverses of the coefficients of the fitted curves, (B from Equation 5.6) and the positions are the peak positions (C from Equation 5.6). Wave lengths are expressed as a percentage of the value for the argument from Equation 5.1, So if the specified value is B_s the quantity plotted here is* $100 \times B_s/B$

is the relatively low resolution used but it is possible that this feature means a more stable wave has been created, a wave that is closer to being permanent.

The most important feature of Figure 5.7 is the approximately linear increase in parameter B with distance, corresponding to an increase in the lengths of the wave peaks. The similarities between all of the waves (except for the highest) indicate it is unlikely that the low order solution used to initialise the simulations accounts for the increase in length. It is most likely that the increase in length is attributable to the same sources as is the loss of height, namely, numerical errors, particle stacking, artificial clustering, and compressibility.

5.1.8 Celerity

It is observed that the celerities of the wave forms extracted from the MARIAN simulations are not always in good agreement with the value predicted by the solitary wave solution used here. The positions of the centre of the fitted wave form, (C in Equation 5.6), are linear in time, and are plotted in Figure 5.8. The slope is the observed wave speed and are tabulated in Table 5.1 for each of the solitary waves. In every case the predicted speed is faster than the observed speed, and the difference increases with the specified wave height.

The main reason for celerities being lower than the ideal is not clear. It could be expected that because the wave heights change with time, the celerities change with time, but this is not observed. These low celerities are most likely to be attributable to numerical errors, particle stacking, artificial clustering, and compressibility. The initial state is unlikely to be a problem here, as it was for the wave height and length. Also possible is that the solution for the wave celerity (Equation 5.4) is over-predicting the wave speed since it is only a second order estimate, and once again, a higher order solution would establish this. This is suggested by the values in Table 5.1 since agreement is worse for higher waves, for which the second order solution is poorer.

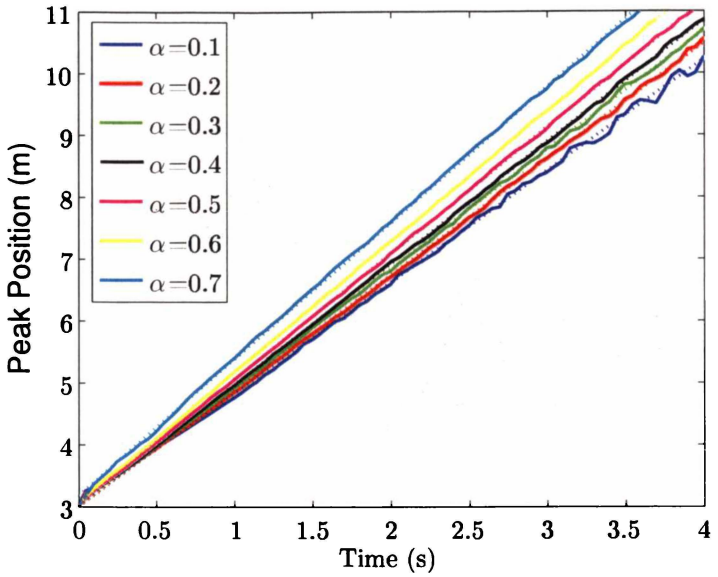

Figure 5.8: *Peak positions for solitary waves by the initial state method. The full lines are the observed values, and the dotted lines are the fitted lines from which celerities are calculated.*

α	Speed (%)
0.1	99.4
0.2	97.5
0.3	93.5
0.4	88.6
0.5	83.8
0.6	77.8
0.7	70.2

Table 5.1: *Observed wave speeds for solitary waves generated by the initial state method, expressed as a percentage of the theoretical speed for the second order wave solution (Equation 5.4).*

5.1.9 A Note on Simulation run-times for Initial State Solitary Waves

All of the initial state solitary wave simulations (each ran for 18 000 time-steps) took between 35 and 48 hours to complete (on computers with AMD Athlon XP2200+ CPU's). In each simulation dynamic time-stepping (to ensure stability conditions are met) was employed for between 1300 and 2700 time-steps, but half-steps are sufficient, and quarter steps or smaller are never required. These computers are a shared resource and run-times here may have been affected by loading on the machines.

5.1.10 Summary for Solitary Waves by Initial State

The seven simulations of the solitary waves propagating in numerical wave tanks showed a variety of features. In this section the overall performance are summarised, and steps that could be taken to improve the simulations are described.

The highest wave is of no use since it broke, while the others provide an approximation to expected behaviour, if not a good approximation. Constant wave heights and lengths are features of permanent waves that are not observed in the simulated waves, and the disagreement gets

worse with time. Numerical errors, neglected boundary integrals, particle stacking and artificial clustering (particularly problematic) are likely causes. The nature of compressibility in SPH may also cause deviations from the well-established behaviour of permanent waves. The role of compressibility in the dispersion of waves could be established by running simulations in MPS or Incompressible SPH models.

The solitary wave solution used to initialise the system was only second order, and improving this would establish the role of the initial state in the quality of the waves generated. The initial state also caused problems because particles are initially placed on the vertices of a rectangular grid. This effect could be reduced by using randomly placed particles, and the free surface could be created from evenly distributed particles, following the approach of Bonet and Lok (1999) in their simulations of an elliptic drops. Spurious energy added by incorrect specification of boundary forces could be improved by spending time finding better parameters, and by using a better formulation for the boundary force.

5.2 Solitary Waves by Goring's Method

A second method for generating waves in SPH is to use Goring's method. This method determines how a piston-type wavemaker should be controlled in order that solitary waves be produced. The basis for the method is simple: the piston is moved so that its velocity matches the depth-averaged water velocity under a solitary wave.

In this section solitary waves produced by Goring's method are examined. Solitary waves with seven different heights are generated in seven separate simulations. A numerical wave tank, similar to the one described in the previous section is employed. The waves propagate from one side to the other, the motion is examined and reasons for various features are discussed. The times required for the simulations to complete are presented. Conclusions are drawn with regard the suitability of the method, and suggestions as to how the method could be improved are given.

5.2.1 Verification of Code for Goring's Method

The literature contains several sources detailing how Goring's method works (Goring, 1978; Goring and Raichlen, 1980) and from these sources a code was produced. Figure 5.9 is a plot of piston trajectories for seven solitary waves and is exactly the same as Figure 7 of Goring and Raichlen (1980, P. 771). There is a decrease in the maximum amplitude of the solitary waves Goring and Raichlen (1980) created as they travel along the wave flume, and while the literature contains methods for reducing loss in wave height (Katell and Eric, 2002), these are not examined here.

5.2.2 Seven Solitary Waves by Goring's Method

Goring's method is used to create seven different waves in a numerical wave tank. The wave tank is the same for each wave, and the waves differ only in the value for α which takes the values 0.1, 0.2, ... 0.7. The wave tank is 0.3 m deep, and 14 m long. It is anticipated that Goring's method produces reasonably good solitary waves in MARIAN, but that they will take significant amounts of time to generate. Goring's method makes use of a first order solution, so the waves need to be of low amplitude to be accurate.

The differences between the simulations here and those described in the previous section are that here the initial state is still, flat water, and that during the course of the simulation the wave boundary points are moved. The initial state was generated by repeated application of VMOSS, punctuated by short simulations in which the system was allowed to evolve unconstrained.

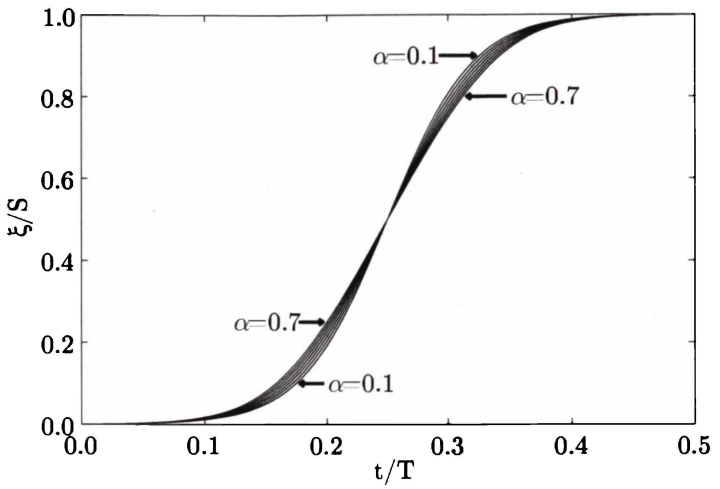

Figure 5.9: *Piston trajectories required for generating solitary waves by Goring's method. The position of the piston (ξ) is measured relative to the initial position and normalised by the total stroke length (S). Time (t) is normalised using the stroke duration (T). This figure matches Figure 7 of Goring (1978) without discernible differences.*

The simulation comprises 10485 particles, each of 0.4 kg, initially placed on a rectangular grid with a spacing of 0.02 m. The smoothing length is 25 mm and so a particle has approximately 20 neighbours. The standard density is 1000 kg/m^3 and a viscosity of 0.001 is used. Viscosity is included in an attempt to damp noise in the simulation, noise which was evident in the simulations of solitary waves from initial state. The XSPH variant is employed with $\epsilon = 0.5$ but CSPH is not.

The standard density is 1000 kg/m^3 and normal/tangential boundaries are used, chosen to avoid corrugations. The speed of sound was chosen to be 25 m/s, so the highest expected velocity has a Mach number slightly lower than 0.1. All simulations used a time-step of 0.25 ms, requiring between 39 000 (for $\alpha = 0.7$) and 80 000 (for $\alpha = 0.1$) time-steps, which differ between simulations because stroke durations are longer for smaller waves and smaller waves travel slower, and so take longer to travel to the end of the wave tank.

Here, as in the previous section on solitary waves by the initial state method, it is anticipated that the smaller waves will be difficult to resolve, while the large will be less accurate because they do not obey the small amplitude assumptions required of Goring's method. Simulation run-times are expected to be much longer, giving more opportunity for the generation of errors.

5.2.3 Free surface Profiles

The position of the free surface is identified in the simulations here by using the same methods described in the previous section on solitary waves by the initial state method. An example of results for a wave generated by Goring's method are shown in Figure 5.10, in this case for the $\alpha = 0.2$ case. Figure 5.10 has many similarities to Figure 5.3, and in the following sections these are noted, as are significant differences.

The Initial State

At the left and right hand limits of the free surfaces in Figure 5.10, there are *menisci*, as there are for later times in Figure 5.3, however here, these are not as large. Some of the spurious boundary potential energy does propagate into the fluid, but a large part has been removed by VMOSS.

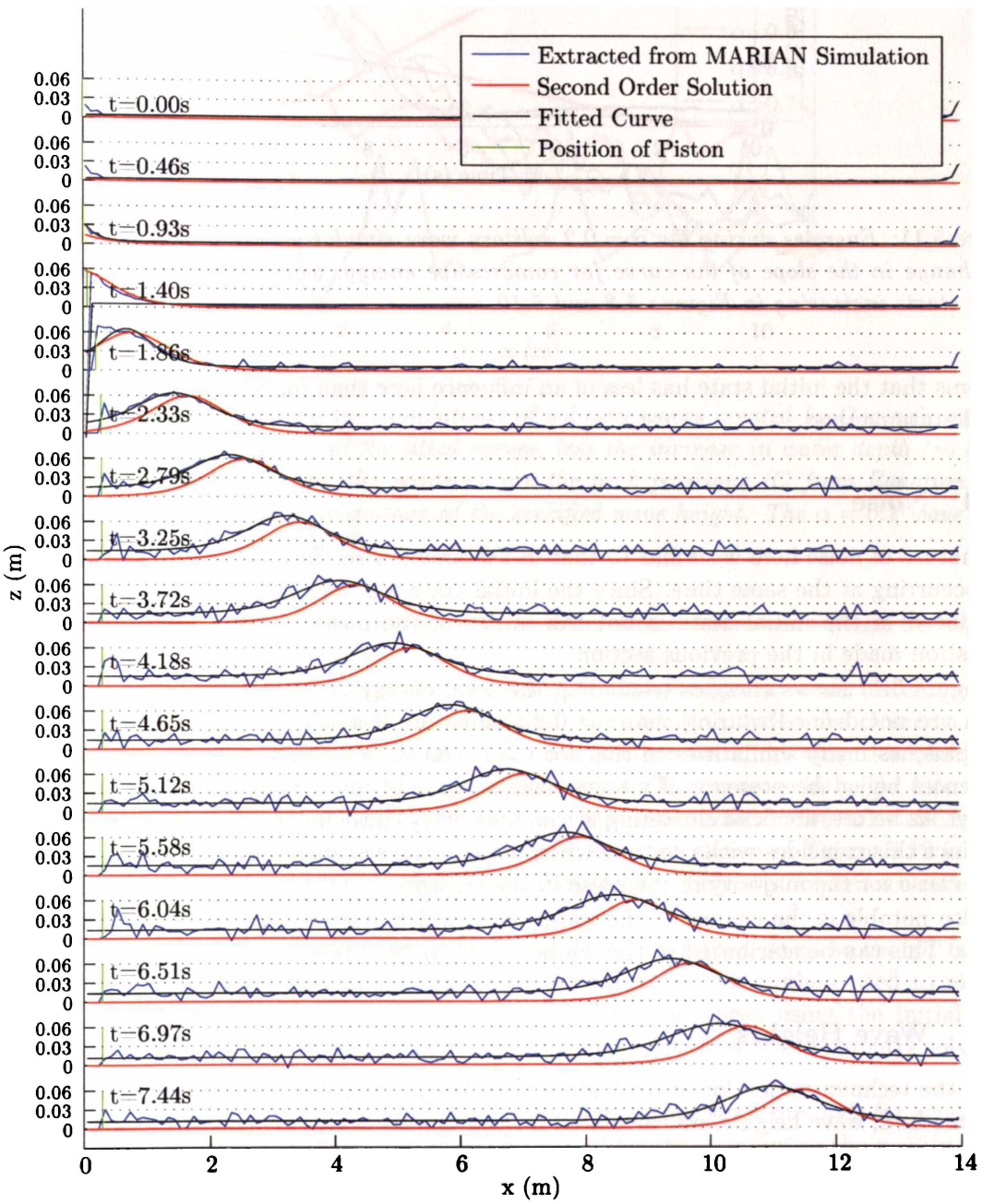

Figure 5.10: *Free surface profiles for the $\alpha = 0.2$ solitary wave created by Goring's method. Note that the different scales used on the vertical and horizontal axes.*

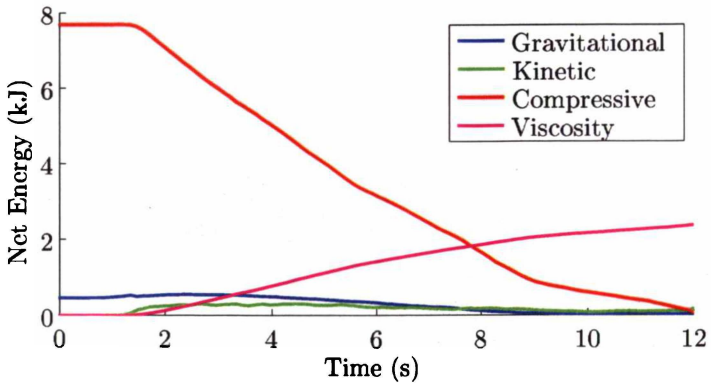

Figure 5.11: *Energies during the $\alpha = 0.2$ solitary wave simulation using Goring's method. Note the change in the slope of the curve for compressive energy, which occurs at the same time as noise starts increasing in Figures 5.3 and 5.10.*

It seems that the initial state has less of an influence here than for solitary waves created by the initial state method.

5.2.4 Noise

The nature of noise here is similar to that in the previous section, being of the same magnitude and occurring at the same time. Since the initial state was well smoothed here, spurious energy introduced in the initial state is not the source of the noise in the system, contrary to the suggestion made in the previous section.

Figure 5.11 shows energies (excluding the total energy and the boundary potential energy, which are not defined) during the $\alpha = 0.2$ simulation of a solitary wave by Goring's method. This plot has many similarities to that for the initial state solitary wave, Figure 5.4. The main differences being the presence of viscous dissipation, and the greater change in the compressive energy. As before, artificial clustering is the most likely cause for the drop in compressive energy, and since this coincides with the time when noise in the system starts to increase, it is the most likely cause for the majority of the noise in the system.

Also notable is the change in the water level, which starts to increase at $t = 1.35\,\mathrm{s}$, just as before. This can be attributed to the method used to determine the position of the free surface.

5.2.5 Wave Heights

Using the techniques developed for finding wave heights in solitary waves created by the initial state method, wave heights are calculated for the solitary waves generated by Goring's method, and the results are plotted against distance in Figure 5.12. There are some interesting features of this plot, and in this section these are discussed.

Figure 5.12 shows slightly different behaviour from the corresponding plot for the solitary waves by the initial state method, Figure 5.5. Both methods generate waves that start high but decrease as the waves propagate along the wave tank. Unlike in Figure 5.5, here the relationship with distance does not appear to be linear, however, when plotted with the same axis limits as Figure 5.5, it does. Prior to reaching $x = 3\,\mathrm{m}$ the wave may not have been completed (the piston has not stopped moving), and the fitted curve is less reliable.

For most of the wave heights (excepting the smallest two) the relative heights are initially very close to 100 %. The poorer initial heights for the smallest two waves are quite likely to be

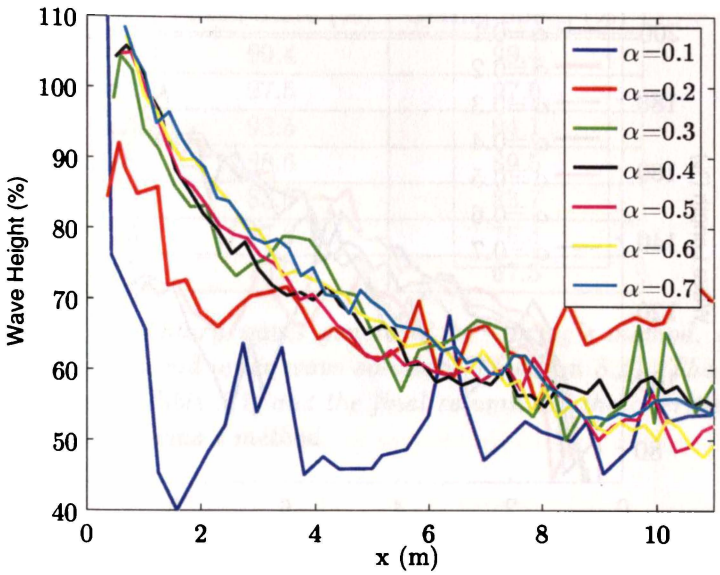

Figure 5.12: *Wave heights during the simulations of solitary waves created by Goring's method. These heights are the magnitudes of the fitted curves, less the increase in water depth (so height* $= D - A$ *from Equation 5.6) and the positions are the peak positions (*C *from Equation 5.6). Wave heights are expressed as a percentage of the specified wave height. The* $\alpha = 0.1$ *wave peaks with a maximum amplitude of 150 %.*

attributable to the relatively low resolution of these simulations. More tests would be required to establish this.

Also apparent is that the heights of the smaller two waves reduce at lower rates, perhaps meaning that once they have achieved stable heights, the resulting waves are closer to being permanent than the higher waves. Without conducting further tests, drawing this conclusion might be dangerous given the amount of noise in the signals. The increase in height late in the simulation for the $\alpha = 0.2$ wave is likely to be similar, there is too much noise to establish with certainty that the increase is really there. The greatest distance shown in Figure 5.12 is 11 m, since 3 m beyond this is the right hand vertical boundary, where the waves experience reflection, invalidating results.

The use of XSPH here means the free surface should be smoother, so noise should be less, but this is not observable to the naked eye, perhaps because of the method used to analyse simulations. The similarities between the simulations here and those using the initial state method indicate that XSPH does not cause severe problems, but does not apparently solve any either. The loss of height is due to some other effect.

Katell and Eric (2002) observed in laboratory experiments, that some methods for generating solitary waves are better than others, and used the loss of height to quantify the quality of the waves. They note that using Goring's method, solitary waves may be damped in height by 17 % at $x = 100h$, h being the depth. They achieved much better results with other methods. Here, at $x = 11$ m, which is $x = 36.7h$, the loss in height is between 43 % and 51 % for all but the smallest wave, for which the loss is 30 %. So the losses are far greater for the simulated waves than for those in laboratory experiments.

5.2.6 Wave Shape

As before, the wave shape is, in part, determined by the length of the wave, which is determined by the coefficient B (in Equation 5.6). As B gets smaller, the length of the waves gets larger,

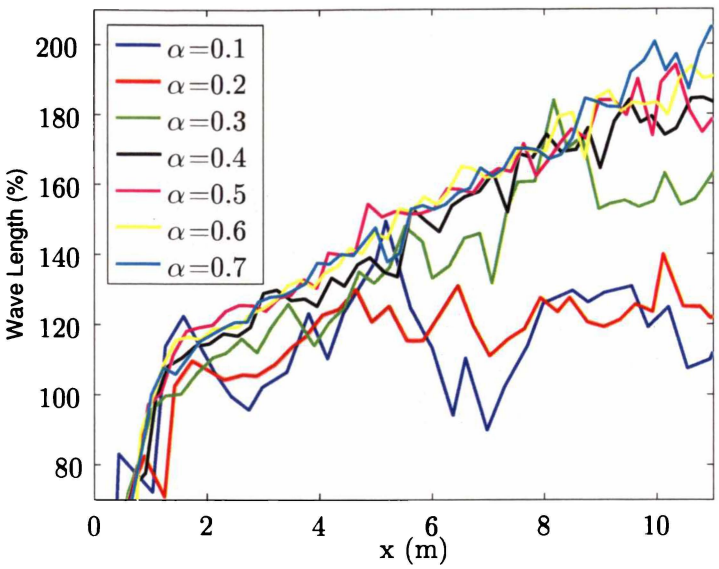

Figure 5.13: *Wave lengths during the Goring's method solitary wave simulations. These lengths are the inverse of the coefficients of the fitted curves, (B from Equation 5.6) and the positions are the peak positions (C from Equation 5.6). Wave lengths are expressed as a percentage of the inverse of the value for the argument from Equation 5.1, so if the prescribed value is B_s, the value plotted here is $100 \times B_s/B$.*

so in Figure 5.13, the inverse of B (expressed as a percentage of the prescribed value) is plotted against peak position for each of the waves as they propagate along the wave tank.

The curves in Figure 5.13 do not start at 100 % because early in the simulation, the waves have not been completed, and the peak of the wave (an imaginary peak since it may not have been created yet) is to the left of any points. Essentially the peak is being extrapolated from the front slope of the wave.

The rates of growth of the lengths are fairly similar for the larger waves, and are similar to those for solitary waves by the initial state method, Figure 5.7. Notably different are the smaller two waves here, but it is not clear whether the lower rates of growth in length for these waves are significant.

5.2.7 Celerity

The same approach used to determine the celerity of the waves generated by the initial state method is used here. Values are presented in Table 5.2 with the results for the initial state solitary waves. It is evident that agreement is very good for some waves, namely the smaller five waves, is poorer for the $\alpha = 0.6$ wave, and quite poor for the highest wave. The highest wave generated by the initial state method broke during the simulation (the same height wave by Goring's method did not), and no importance can be attributed to this value.

It is tempting to infer that the good agreement in celerities is an indication that the second order solution used in the initial state method is not problematic, but it must be remembered that Goring's method also relies on a low order solution.

5.2.8 Simulation Run-times Using Goring's Method

Simulation run-times for solitary waves are significantly longer than for waves created by the initial state method. The Goring's method waves take longer because of the time required to

α	Initial State (%)	Goring Speed (%)
0.1	99.4	99.4
0.2	97.5	97.6
0.3	93.5	94.1
0.4	88.6	89.5
0.5	83.8	83.7
0.6	77.8	76.4
0.7	70.2	67.5

Table 5.2: *Wave speeds for solitary waves generated by Goring's method, expressed as a percentage of the speed for the second order wave solution (Equation 5.4). The second column is a reproduction of the results in Table 5.1, and the final column are the celerities calculated for the solitary waves generated by Goring's method.*

generate the wave, and the time required for the wave peak to travel to 3 m, where the initial state wave was first placed. For smaller waves the time required to generate the wave is much longer than for larger waves, for example, for the $\alpha = 0.1$ wave 4.7 s are required, while for the $\alpha = 0.7$ wave 1.7 s are necessary. The time required for a wave to travel 3 m also depends on wave height, being approximately 1.7 s for $\alpha = 0.1$ and 1.0 s for $\alpha = 0.7$. In total, the Goring's method simulation for $\alpha = 0.1$ required an additional 6.4 s (compare this to 4.4 s for the wave to travel to the 11 m mark, 3 m from the other end of the wave tank), and the $\alpha = 0.7$ wave an additional 2.7 s (compared to 2.6 s to travel 8 m).

So for the smallest wave considered here, using Goring's method requires approximately 250 % longer, and for the largest wave, 200 % longer. This in addition to the smaller wave taking longer anyway.

5.2.9 Improvements to Goring's Method in MARIAN

Goring's method is not perfect and generates waves which are not permanent. Katell and Eric (2002) developed a method based on the Rayleigh equations (rather than the Boussinesq equations that Goring's method uses) and with this method generated waves in a laboratory wave tank that are more stable. This provides an avenue by which the implementation of Goring's method in MARIAN could be improved.

The majority of wavemakers have a single surface that creates waves. This surface may rotate about a hinge along the base (a flap wavemaker) or move parallel to its normal (a piston wavemaker). Possible also are plunger wavemakers that create waves as they are lifted out of, or pushed into the water and the front surfaces on these can be a variety of angles. But for the majority of wavemakers the front surface remains the same shape and retains the same orientation. This is for practical reasons: creating anything else is difficult. There is no reason why a computational model needs to be constrained like this though. Koshizuka and Oka (2000) took advantage of this in their MPS model and have wave boundary points that move with different amplitudes for different depths. This would be an improvement here also, and makes increasing the order of solution used an easier process.

5.2.10 Summary for Solitary Waves by Goring's Method

The solitary waves generated by Goring's method are very similar to those created by the initial state method. The same spreading and shrinking of the peak occur for both approaches. For the first approach the poor initial state decreases accuracy, while the main detriment to Goring's

method are the considerably longer run-times required.

5.3 Cnoidal Waves by the Initial State Method

In the preceding sections of this chapter, solitary waves were generated by two different methods, and the resulting waves were examined. Solitary waves have been used to approximate single peaks in periodic trains within the surf zone, so the topic is appropriate here. But it may be that periodic waves are desired, so here cnoidal waves are examined.

In this section the initial state method is used to generate five different cnoidal waves within the numerical wave tank. Their properties are examined and the quality of these waves established. Practical limits on the application of SPH in generating cnoidal waves are suggested, as are approaches for extending these limits and reducing undesirable features.

5.3.1 Cnoidal Waves

In the previous section it was established that two methods available in MARIAN for making waves can be used to generate similar quality of waves (with limits on the amplitude due to the order of solution employed), but that when Goring's method is employed, much longer run-times are required. It was observed that for solitary waves, irrespective of which method was used, the waves reduced in height and increased in length as they progressed along the wave tank, and did so rather quickly. This has concerning implications for simulating breaking waves since the wave may have changed considerably by the time it reaches its breaking point. Here it is determined whether the features seen in solitary waves occur for cnoidal waves also.

When Goring's method is used to create cnoidal waves the first wave peak is not a cnoidal wave (Goring and Raichlen, 1980): one must wait for the subsequent peaks to obtain a reasonable approximation to cnoidal waves. Figure 5.14 is reproduced from Goring and Raichlen (1980) and shows the evolution of a train of four cnoidal waves in a wave tank. Only the central two troughs and the central three peaks are permanent. The outer two troughs and the first peak change shape as they propagate. So to create a high quality wavelength from a cnoidal wave train, at least three wavelengths must be generated, of which the outside two are of no use. The time required to generate the first wavelength (which is of no use) adds considerably to the required runtime.

In this section cnoidal waves are generated by the use of the initial state method, avoiding the long run-times required by Goring's method. Here five waves are generated in a numerical wave tank and the propagation of these waves is examined.

5.3.2 A Second Order Cnoidal Wave Solution

As for solitary waves, solutions for cnoidal waves are usually expressed as a power series in α, and the number of terms that are retained determines the accuracy (and limits of applicability) of the solution. Here a second order solution is presented, from Dean and Dalrymple (1991).

The free surface is given by (Dean and Dalrymple, 1991)

$$\eta(x) = H\mathrm{cn}^2\left(\theta \,|k\right) \tag{5.7}$$

which is written in terms of the Jacobi elliptic function cn, the modulus k and where $\theta = \sqrt{\frac{3}{4}\frac{\alpha}{h^2}}x$. Other researchers use the complementary modulus $k' = \sqrt{1-k^2}$, or the elliptic parameter m,

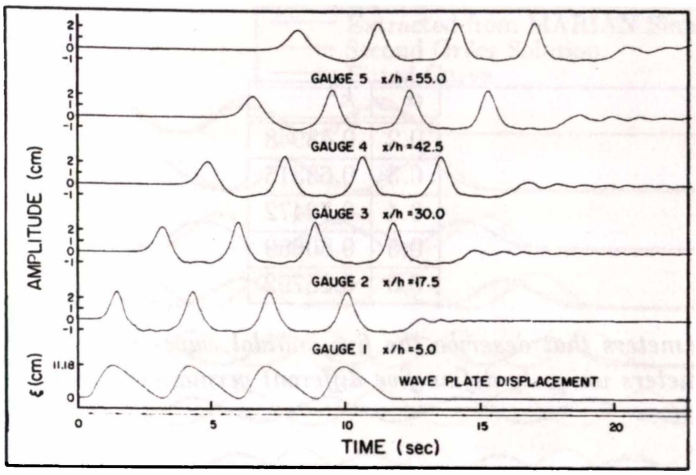

Figure 5.14: *The evolution of a cnoidal wave train, after Goring and Raichlen (1980). Here ξ is the piston displacement. The trough is below zero because all elevations in Goring's method are reduced to ensure a zero average level.*

given by $m = k^2$. The horizontal and vertical components of the velocity are given by

$$u(x,z) = C\left\{\frac{\eta}{h} - \frac{\eta^2}{h^2} + \frac{3}{2}\frac{1}{k^2}\left(\frac{1}{6} - \frac{1}{2}\frac{z^2}{h^2}\right)\right. \tag{5.8}$$
$$\left.\times\left[\left(1-k^2\right)\alpha^2 + 2\left(2k^2-1\right)\alpha\frac{\eta}{h} - 3k^2\frac{\eta^2}{h^2}\right]\right\}$$

$$w(x,z) = -C\frac{d\eta}{dz}\frac{z}{h}\left\{1 - 2\frac{\eta}{h} + \frac{1}{2}\frac{1}{k^2}\left[1 - \frac{z^2}{h^2}\right]\left[\left(2k^2-1\right)\alpha - 3k^2\frac{\eta}{h}\right]\right\} \tag{5.9}$$

in which

$$\frac{d\eta}{dz} = -\sqrt{3\alpha}\frac{\alpha}{k}\mathrm{cn}\left(\theta\,|k\right)\mathrm{sn}\left(\theta\,|k\right)\mathrm{dn}\left(\theta\,|k\right). \tag{5.10}$$

In this section the vertical coordinate is positive up and is zero at mean sea level. The wave celerity is given by

$$C = \sqrt{\frac{gh}{\left(1+\alpha\left(\frac{1}{k^2}-2\right)\right)}} \tag{5.11}$$

and the wavelength by

$$\lambda = 4hkK(k)\sqrt{\frac{1}{3\alpha}} \tag{5.12}$$

in which $K(k)$ is the complete elliptic integral of the first kind. The pressure is hydrostatic, so Equation 5.5 is appropriate.

5.3.3 Five Cnoidal waves

A family of simulations are generated, each specified with a different cnoidal wave. The different waves used are given in Table 5.3. These waves are chosen (somewhat arbitrarily), as being examples for which the numerical wave tank used here would be long enough. All of these waves have the same wavelength of 2 m, which was chosen so that the wave train could travel a reasonable distance before reflections from the right hand boundary occur. An example of an initial state is shown in Figure 5.15. Anticipating similar results as in Section 5.1 (solitary waves by the initial state method), the wave heights are chosen to avoid problems with being too small (noise is comparatively large) or too high (the solution is poor).

α	k^2
0.2	0.48948
0.3	0.63315
0.4	0.73472
0.5	0.80669
0.6	0.85792

Table 5.3: *The parameters that describe the five cnoidal waves examined here. With the depth (0.3 m), these parameters uniquely define five different permanent cnoidal waves.*

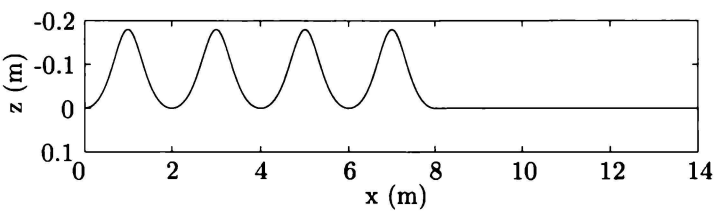

Figure 5.15: *The free surface chosen for the initial state for the $\alpha = 0.6$ cnoidal wave which travels from left to right. The wave train was placed so that a trough occurs at the left hand boundary, so velocities are zero along that boundary. Beyond 8 m the displacement is set to zero.*

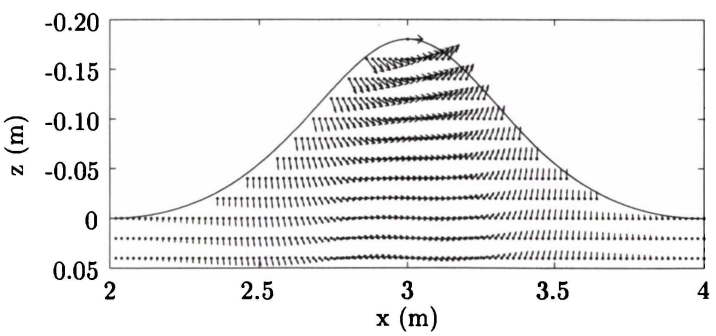

Figure 5.16: *Detail of the initial state for the $\alpha = 0.6$ cnoidal wave. The change in the direction of velocities occurs three times along some rows of particles, rather than once as should occur, the result of using a low order solution.*

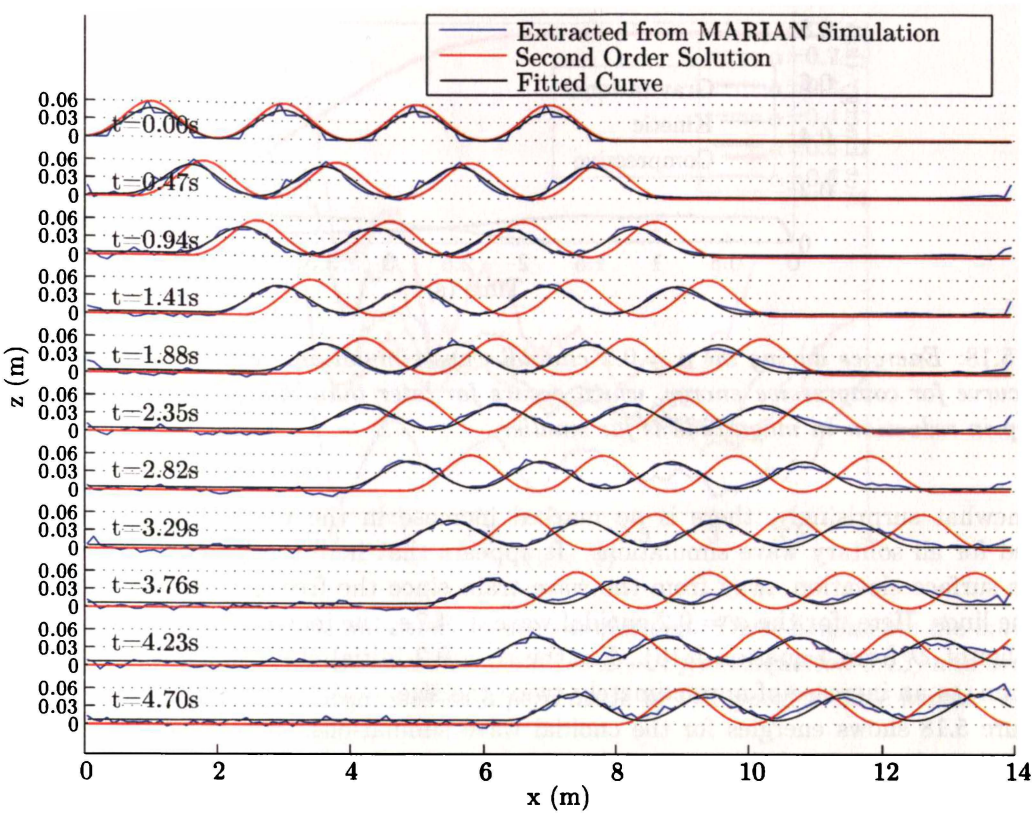

Figure 5.17: *Free surface profiles for the $\alpha = 0.2$ cnoidal wave by the initial state method. Note the different scales used on the vertical and horizontal axes. The second order solution (the red curve) is always 0.06 m high.*

In addition, the changes in direction for the initial velocity field that occurred for solitary waves which are too high (see Figure 5.2) also occur for cnoidal waves (see Figure 5.16). For the $\alpha = 0.4$ wave these are observable but small, and occur low in the water column. These features are worse for the higher $\alpha = 0.5$ and $\alpha = 0.6$ waves, occurring higher and being larger.

The initial states contain approximately 10 000 particles (0.4 kg each), the exact number being dependent upon the height of the wave, these particles being initially placed on a rectangular grid with a spacing of 0.02 m. A time-step of 0.2 ms was used with a smoothing length of 0.025 m. The simulations ran for between 16 500 steps and 23 500 steps, the exact number depending on the celerity of the wave in the simulation. The standard density is 1000 kg/m^3 and the speed of sound is 25 m/s. The normal/tangential boundary type was used and a viscosity of 0.0005 was employed. The XSPH variant was employed with $x = 0.5$, but CSPH was not.

The free surface was analysed using the same techniques as for solitary waves. The fitted curve is

$$z = A\mathrm{cn}^2\left(Bx + C\right) + D. \tag{5.13}$$

Figure 5.17 shows the free surface for the $\alpha = 0.2$ cnoidal wave.

5.3.4 Noise and Water level

Once again, note the increasing amount of noise in Figure 5.17. However the noise here (in comparison to that in Figure 5.3) is smaller in magnitude. For the solitary wave simulations the noise occurred both in front of, and behind the wave, while here the free surface in front of the wave train remains quite flat (neglecting the effect of the boundary).

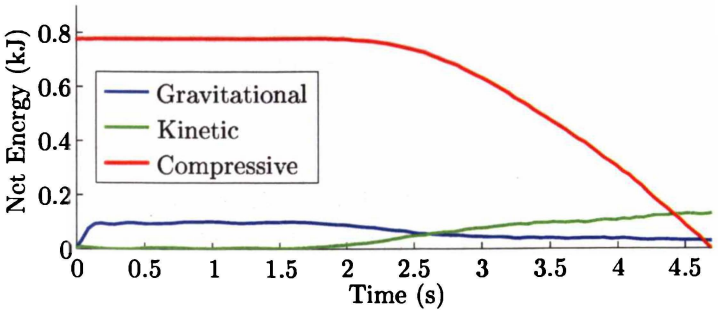

Figure 5.18: *Energies during the $\alpha = 0.2$ cnoidal wave simulation. Note the change in the slope of the curve for compressive energy, which occurs far later than that for the solitary waves and the drop in reduction in magnitude is far smaller.*

Somewhat surprisingly, there is no marked increase in the water level, something which occurred for all solitary wave simulations. It appears that here the fitted curve over-estimates the free surface elevation away from the wave train since the free surface lies between the red and blue lines. Here, for the $\alpha = 0.2$ cnoidal wave at 4.7 s, the increase in water level is 6.84 mm (the value of D from Equation 5.13). For the $\alpha = 0.2$ initial state solitary wave at 4.5 s it is 12.67 mm, so an increase of approximately twice the size.

Figure 5.18 shows energies for the cnoidal wave simulations. It was concluded in previous sections that the cause of the noise is due to the drop in compressive energy, and here, since the drop is smaller and occurs later, the noise is of lower magnitude. The lower magnitude of the noise here (in comparison to the simulations of solitary waves by Goring's method) cannot be attributed to viscous dissipation since here the viscosity is smaller (although it is still much larger than for real water). As before, plots of particle positions are produced, and once again, the drop in compressive energy can be linked to artificial clustering.

It is not clear why the drop in compressive energy occurs that much later for the cnoidal waves than it did for the solitary waves. Probably, if the simulation were left longer, the noise would increase to the same magnitudes and the water level to the same height as are observed in the solitary wave simulations.

5.3.5 Wave Height

The wave heights for the cnoidal waves are quite different to those for solitary waves. These heights are shown in Figure 5.19 in which they are plotted against time, rather than position, since position is not so well defined for a train of waves. There are several important features in this plot, and these are discussed in the following paragraphs.

The waves all start with slightly low heights. This occurs for the same reason as it did for solitary waves by the initial state method: the initial placement on a rectangular grid. For every wave the height increases early in the simulation, which can be explained (as for initial state solitary waves) in part, by *layers* of particles rolling over the wave crests, and in part by the low order of solution (for the higher waves).

But there is a notable feature of the wave heights for these cnoidal waves that did not occur for the solitary waves, the periodic increase and decrease in the first half of the simulations. It is not entirely obvious in Figure 5.18, but there is a corresponding beating in the net kinetic and gravitational potential energies, shown in Figure 5.20.

The beating in Figure 5.20 is caused by the addition of spurious energy as either boundary potential energy (even though, strictly speaking, this is not defined here) or initial compressive energy. Either way, better initial states and boundary potentials would solve the problem.

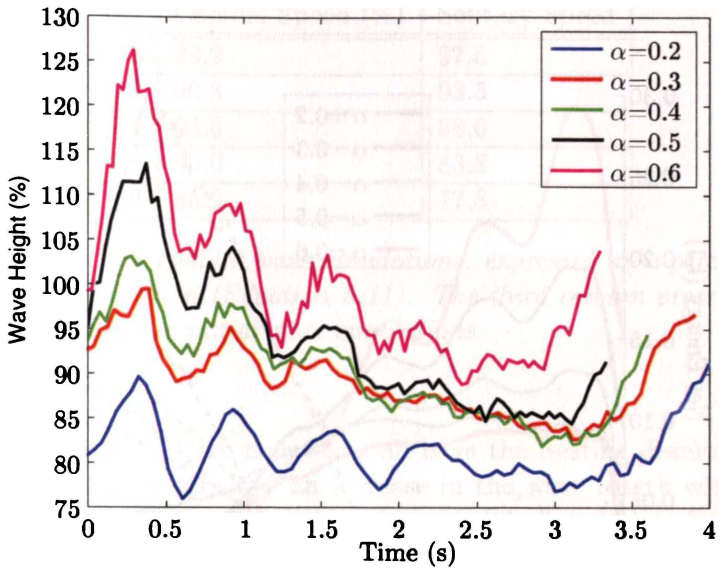

Figure 5.19: *Wave heights during the cnoidal wave simulations. These heights are the magnitudes of the fitted curves, less the increase in water depth (so $D-A$ from Equation 5.13) and are plotted against time. Wave heights are expressed as a percentage of the specified wave height.*

Spurious boundary potential energy causes the initial increase in gravitational potential energy early in each simulation and contributes about the same amount of energy in each simulation.

The drop in gravitational potential energy in the second half of the simulations in Figure 5.20 is strange since there is an increase in the water level at that time, but it should be recalled that noise (and the subsequent over-estimation of the free surface) is of increasing significance in the second half of the simulations.

Figure 5.19 shows that late in every simulation the wave heights start to increase. This occurs because the leading crest in the wave train has reached the far end of the tank and is being reflected. The resulting increase in the free surface height in that section of the tank drags up the fitted wave height, and can be seen in Figure 5.17.

A significant, and surprising feature of Figure 5.19 can be seen when it is compared to the corresponding figures for solitary waves, both by the initial state method (Figure 5.5) and by Goring's method (Figure 5.12). The loss in wave height is far less for the cnoidal waves. For the $\alpha = 0.2$ cnoidal wave the height always lies between 83 % and 100 % of the prescribed height, in comparison, values for the $\alpha = 0.2$ solitary wave are between 46 % and 106 % of prescribed values.

It can be seen in Figure 5.17, that the leading crest in the wave train loses height significantly faster than do the trailing crests. This occurred in all cnoidal wave simulations. It is certain that the leading wave is not a permanent cnoidal wave, and this may explain the loss in height. But like the solitary waves, the leading crests are propagating into still water, which is where particle stacking would be most significant if it were to play a role.

5.3.6 Wave Shape

In Figure 5.17 the cnoidal waves retain their shape well during the simulation. This is demonstrated by the good match between the free surface (the blue line) and the fitted curve (the black line), which is a cnoidal wave. Figure 5.21 shows the wavelengths of the fitted curves, against time (actually it shows the relative value of parameter B from Equation 5.13, but this is closely related to the wavelength).

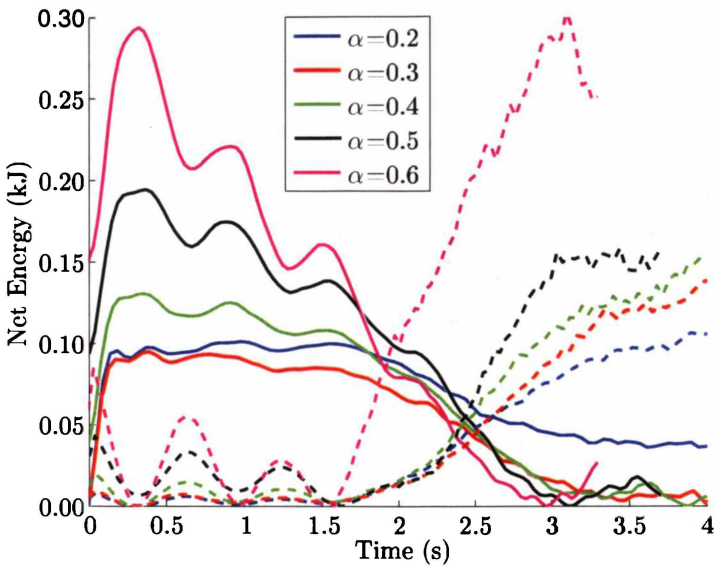

Figure 5.20: *Kinetic (dashed lines) and gravitational potential (full lines) energies for the cnoidal wave simulations. Lines with the same colour are for the same simulation. Note the periodic beating in these energies, which are exactly in phase with the beating in the wave heights in Figure 5.19.*

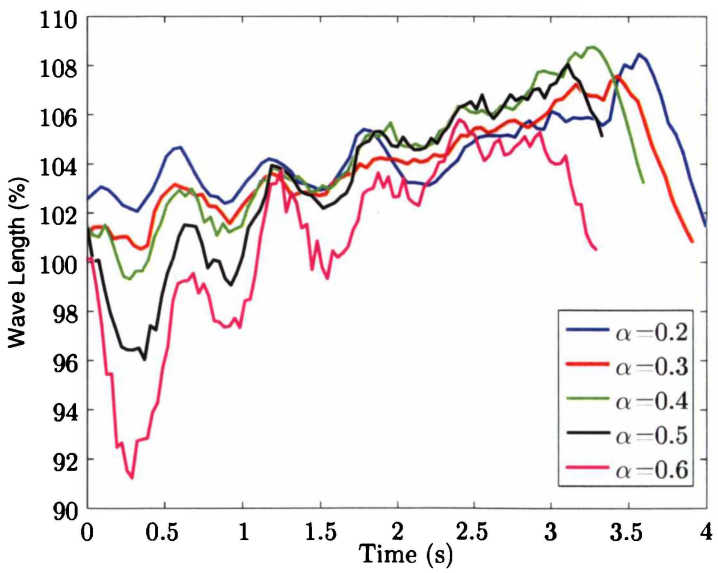

Figure 5.21: *Wavelengths in the cnoidal wave simulations, the value B from Equation 5.13 expressed as a percentage of the prescribed value.*

α	Cnoidal Speed (%)	Solitary Speed
0.2	99.2	97.5
0.3	96.8	93.5
0.4	93.6	88.6
0.5	90.0	83.8
0.6	85.9	77.8

Table 5.4: *Wave speeds in the cnoidal wave simulations, expressed as a percentage of the speed for the second order wave solution (Equation 5.11). The third column presents for comparison the solitary wave speeds from the initial state simulations.*

All of the lines in Figure 5.21 are noisy, and all have the beating discussed in the previous section. But more significantly, all show an increase in the wave length with time, essentially describing a dispersive wave. However in comparison to the related plots for solitary waves the growth is very slow here. It may be that much of the growth can be attributed to the method used to find the free surface, and the errors it introduces.

Returning to Figure 5.17, in the later times shown, it can be seen that the fitted wave is virtually always under the free surface. If not constrained by the still water in front of and behind the wave train, the fitted curve would have a higher offset (parameter D in Equation 5.13). It is also clear in Figure 5.17 that, later in the simulation, the troughs in the cnoidal wave train are not at zero, where they should ideally be. Given the small magnitude of this effect, it is most likely that the method for finding the free surface is to blame, over predicting wave heights within the train. However it may be that the troughs really are rising during the simulation. More testing would be necessary to establish this.

5.3.7 Wave Celerity

The position of the first peak in the wave train was identified and used to determine the celerity of the wave train for each of the cnoidal wave simulations. From the peak positions a straight line was fitted, the slope of which represents the celerity of the wave. Once again, the higher waves travel faster, and the peak position is very clearly linear in time, indicating a constant wave speed. The wave speeds calculated from the simulations are presented in Table 5.4.

There are several significant features of Table 5.4. Just as for the solitary wave simulations, the wave speeds here are always lower than they should be and the higher the wave is the lower the relative wave speed. Also, the cnoidal wave speeds are closer to their predicted values than the corresponding solitary wave speeds are. As before, it is possible that the predicted wave speed is larger than it should be, a result of the low order of solution employed. That the celerities are in less agreement for higher waves supports this.

5.3.8 Summary for Cnoidal Waves by Initial State

The results for the cnoidal waves simulations are not extremely good, but surprisingly, are better than those for the solitary waves. Many of the suggestions for how the solitary wave simulations could be improved apply here also: a kernel that avoids artificial clustering, including the neglected boundary terms, better initial states and a better method for identifying the free surface.

For both solitary waves and cnoidal waves the observed celerities are slower than expected, and the disagreement increases for larger waves. As before, the cause of this is not clear but the

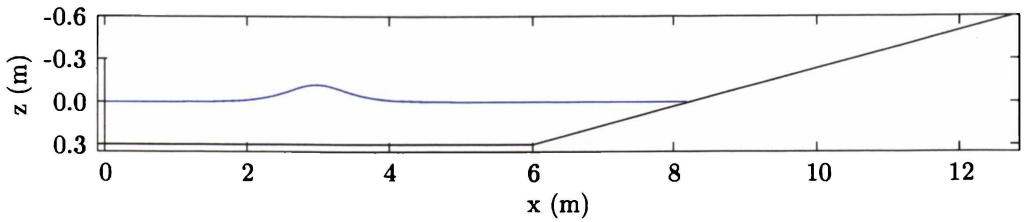

Figure 5.22: *The free surface in the initial state for an $\alpha = 0.4$ solitary wave on beach with slope 1:7.5. The dry section of the beach is large to allow for run-up.*

low order of the solution they are compared to contributes. Noise grew faster in the solitary wave simulations than in the cnoidal wave simulations, and it is not clear why. The cnoidal waves are a much better approximation to permanent waves than the solitary waves, and once again, it is not clear why.

5.4 Six Plane Slope Beaches

In the previous sections the generation and propagation of solitary and cnoidal waves in MARIAN were examined. This is an important step since these waves are used to simulate waves breaking on beaches. In this section a number of simulations of solitary waves breaking on plane beaches are described. These simulations are chosen because laboratory experiments have been described in the literature against which comparisons can be made.

The initial state method is used to generate the same wave on six beaches, each with a different slope. The transformation of the waves is examined as depth induced breaking occurs. Comparisons against the literature are conducted, establishing if the simulations reflect reality. Reasons for various features are put forward.

5.4.1 The Initial States

In all, six beaches are examined, with slopes of 1:7.5, 1:10, 1:15, 1:20, 1:25 and 1:30. The initial state contains a solitary wave with $\alpha = 0.4$ in water 0.3 m deep. The peak is initially at $x = 3$ m, placed so that the ratio $\eta(x)/H$ is small at the left hand vertical boundary (the wave is traveling to the right). The beaches start at $x = 6$ m, where, once again, the ratio $\eta(x)/H$ is small. An example of an initial state is shown in Figure 5.22 for the 1:7.5 beach. The dry section of the beach is large, so as to allow the wave to run-up uninterrupted.

Between 9000 and 13 000 particles are used (each with mass 0.237 kg), the exact number being dependent on the beach slope: more for gentler slopes. All simulations use a time-step of 0.2 ms and a smoothing length of 0.03 m. The standard density is 1000 kg/m^3 and the speed of sound is 25 m/s. Normal/tangential boundaries are used and a viscosity of 0.0005 is employed. XSPH is used with $\epsilon = 0.5$ but CSPH is not. The simulations run for between 14 000 time-steps and 31 000 time-steps, depending on the beach slope.

These simulations stop when the waves have reached water shallow enough that they should have broken, judged by the value of the ratio of wave height to water depth. In the following sections the transformation predicted in these simulations are examined, reasons for the various features are offered, and suggestions for improvements put forward.

5.4.2 Solitary Waves Breaking on Plane Beaches: Expected Behaviour

As a solitary wave approaches a plane sloped beach, it experiences four major regimes of evolution (Synolakis and Skjelbreia, 1993), shown schematically in Figure 5.23. Synolakis and Skjelbreia

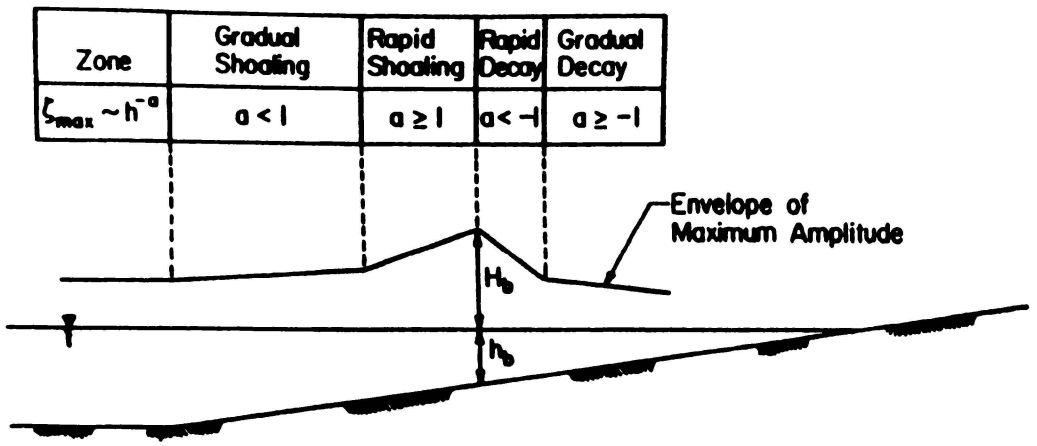

Figure 5.23: *Definition sketch showing four major regimes in solitary wave transformation (after Synolakis and Skjelbreia, 1993). Here ζ_{max} is the maximum value of the wave amplitude, H_b is the wave height at the breaking point and h_b is the water depth at the breaking point. The parameter a indicates how ζ_{max} scales with depth according to $\zeta_{max} \sim h^{-a}$.*

(1993) state that comparisons to Figure 5.23 are only zeroth order accurate. A wave in the gradual shoaling regime experiences a gradual increase in height till it enters the zone of rapid shoaling, when the height increases rapidly with a corresponding increase in the steepness of the face, and at the conclusion of the zone of rapid shoaling the wave breaks. Following breaking, the wave passes through zones of rapid decay and then gradual decay. The simulations performed here should show similar behaviour to this.

Li and Raichlen (1998) obtained a photograph of a breaking solitary wave with $\alpha = 0.3$ on a beach with a slope of 1:15, a photograph that is reproduced in Figure 5.24, and one that has an easily observed barrel. For the beach with slope 1:15 a plunging wave is expected, but because the wave here has $\alpha = 0.4$, it tends slightly more to spilling side of plunging, since steep waves have higher Iribarren numbers.

The steeper the beach, the more the wave is pushed toward the collapsing side of the breaking spectrum, and conversely, the gentler the beach the more it is pushed to the spilling side of the spectrum. It has been observed that some solitary wave do not break, meaning that they surge (Synolakis, 1987). It has been determined that solitary waves do not break during run-up (as opposed to during the backwash) if (Synolakis, 1987)

$$\alpha < 0.479(\tan\beta)^{10/9}. \tag{5.14}$$

So for all of the beaches considered here, a wave with $\alpha = 0.4$ should break during run-up.

5.4.3 Simulations of Solitary Waves Breaking on Plane Beaches

In the following sections the simulations of solitary waves breaking on plane beaches are examined. Firstly, the breaker over the 1:15 slope is examined and the lack of any barrel is demonstrated. By comparison with a similar wave from the literature it seems that this wave should have plunged. The maximum wave height envelopes are drawn for the waves on each of the beaches, and this is be compared to expected behaviour. The velocities within the wave are plotted to clarify exactly what transformation the wave has undergone. Finally, suggestions for the causes of the observed behaviour are given.

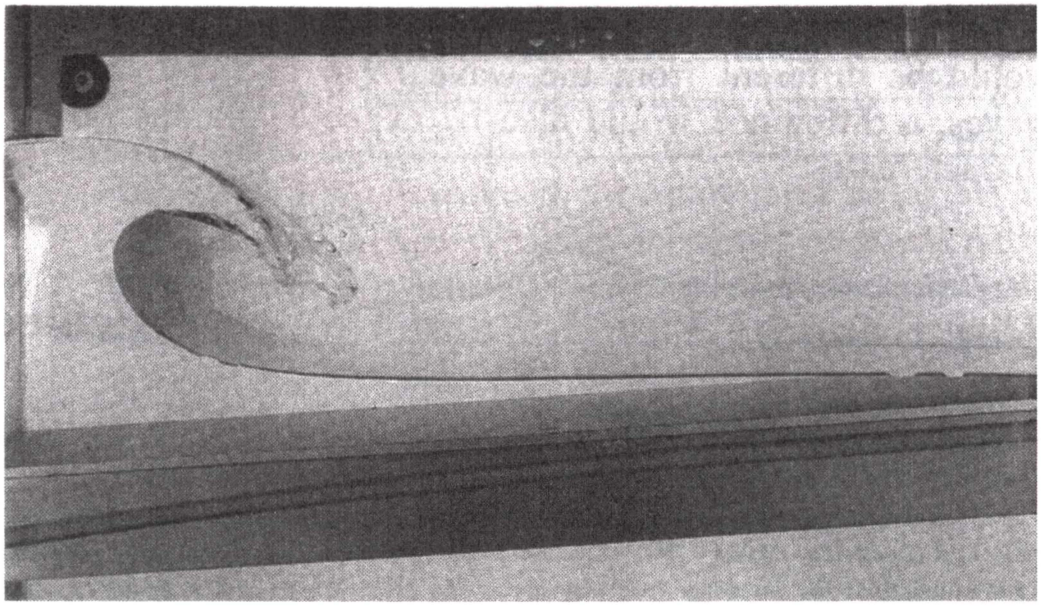

Figure 5.24: *A plunging solitary wave ($\alpha = 0.3$) on a beach with slope 1:15 (from Li and Raichlen, 1998).*

The Free Surface for Solitary Waves on Plane Beaches

Using the same techniques described in the previous sections, the free surface was identified and an example (for the 1:15 sloped beach) is plotted in Figure 5.25. The rightmost particle is initially at x=10.2 m, and occurs in a layer only one particle thick. Spurious boundary and gravitational potential energies generate some of the disturbance that occur near this point, but it is primarily caused by a single layer of particles being pushed up the beach.

It seems some transformation has occurred as the wave passes by the red line, however. The red line is the point at which Equation 2.6 is satisfied, but its position is variable. More accurate expressions than Equation 2.6 are available that account for beach slope, but this is not important here. Of importance is the height at the breaking point, and this is not known because no clear breaking occurs. The red line in Figure 5.25 was generated by using the initial wave height.

For most of the beaches considered here the transformation of the wave is similar. As it passes the estimated breaker position the wave becomes more peaked (but does not increase in height), the rear slope becomes more gentle, and the face of the wave becomes steeper. For the steepest beach the wave does not experience the same transformation, and instead looks very much like a surging breaker.

Looking for Barrels on the 1:15 Beach Breaker

The aim in this section is to show that no barrel is visible in this simulation, however Figure 5.25 cannot be used to establish this. The simple method used to identify the free surface cannot resolve situations when the free surface occurs more than once in a single bin. The particle positions are plotted in Figures 5.26 and 5.27 and it is clear that no significant barrels exist, so the wave has not plunged as it should have (see Figure 5.24).

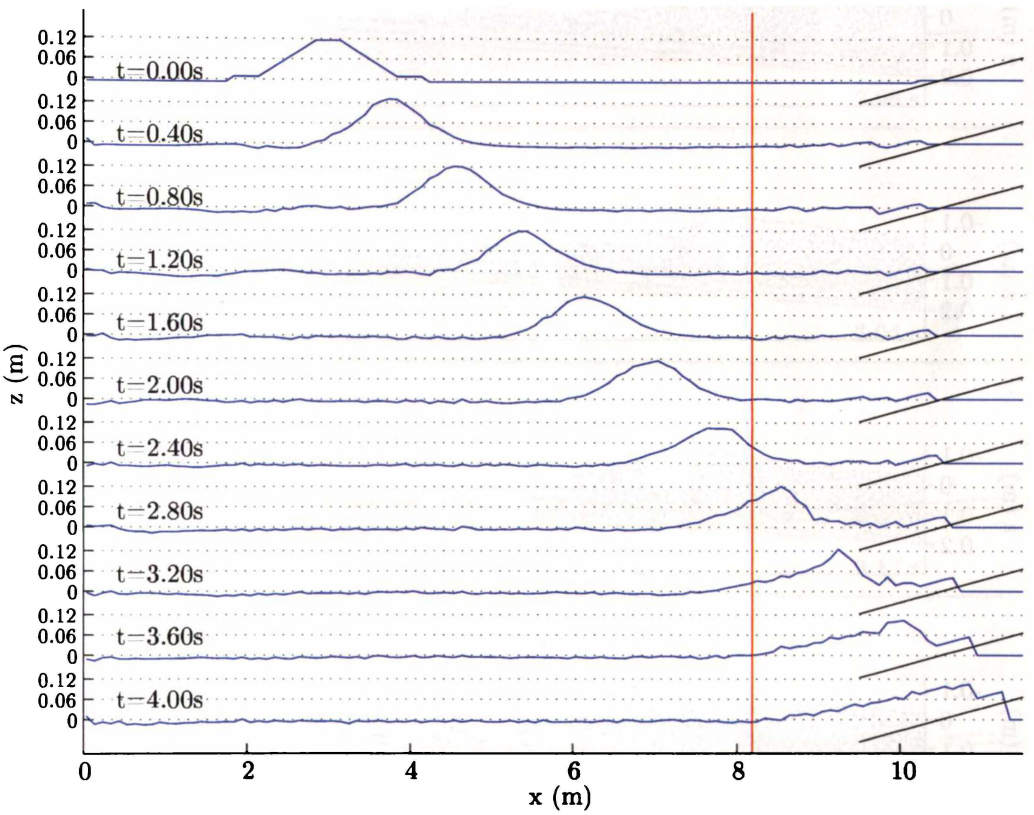

Figure 5.25: *Free surface profiles for* $\alpha = 0.4$ *on a beach with a slope of 1:15. The vertical red line is approximately where the wave should break, according to Equation 2.6. The beach starts at* $x = 6\,m$*, and is shown by the black lines, but only beach-ward of* $9.5\,m$ *to ensure the plots remain legible.*

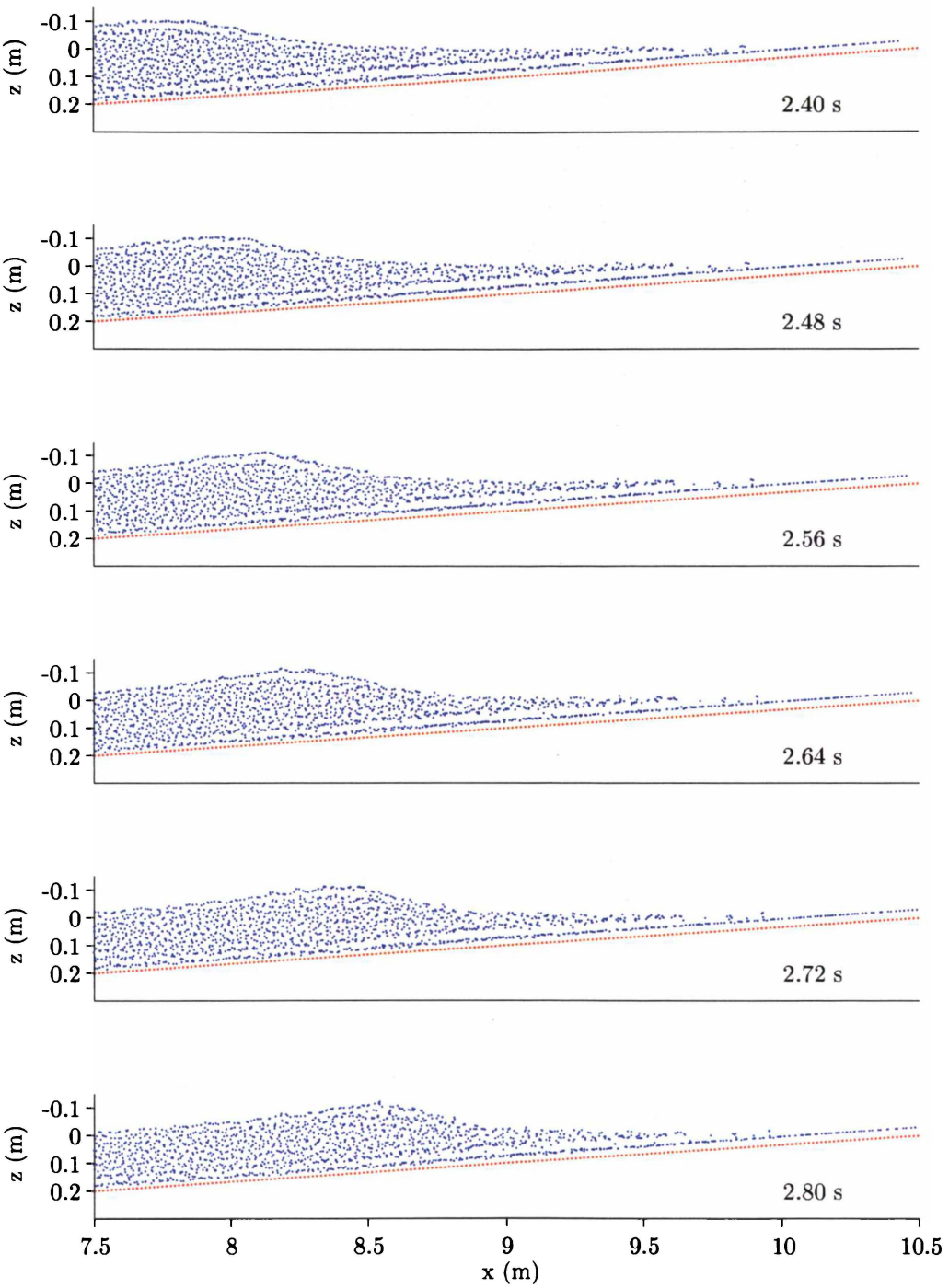

Figure 5.26: *Particle positions for an $\alpha = 0.4$ solitary wave on a 1:15 beach. Continued in Figure 5.27. Every 400th time-step is plotted (even though every 200th step was written to file) for simplicity.*

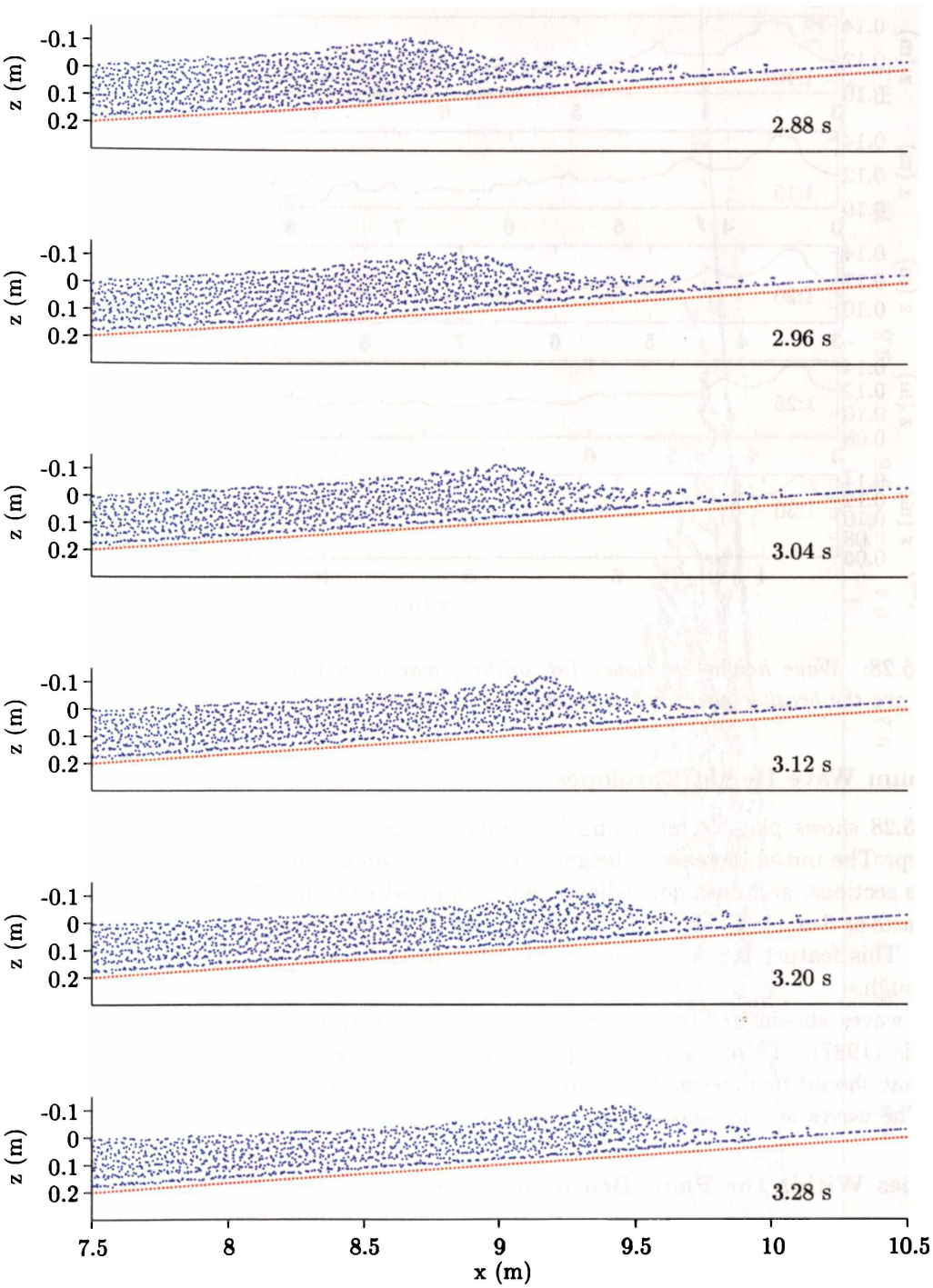

Figure 5.27: *Particle positions for an* $\alpha = 0.4$ *solitary wave on a 1:15 beach. Continued from Figure 5.26.*

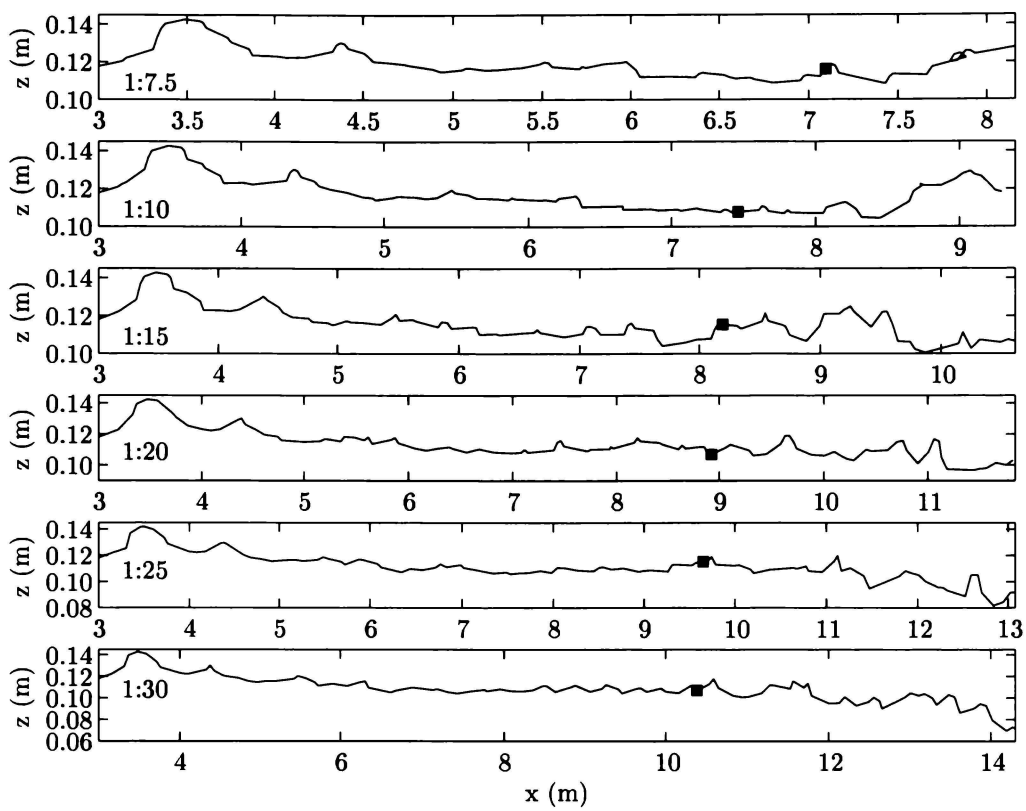

Figure 5.28: *Wave height envelopes for solitary waves on the plane slope beaches. The black squares are the breaker positions estimated from Equation 2.6.*

Maximum Wave Height Envelopes

Figure 5.28 shows plots of maximum height envelopes, the highest point on the wave at each time-step. The initial increase in height early in each simulation occurs for reasons discussed in previous sections, and does not indicate depth-induced shoaling. This is certain since the beach slope starts at 6 m. At the far right of the top two plots the envelopes start to rise, demonstrating run-up. This feature is not present in the other plots because they are not allowed to run for long enough.

The waves shown in Figure 5.28 do not experience transformation like those described in Synolakis (1987). Throughout these plots, the theme is gradual decay. There are no marked peaks that should be present if the waves are experiencing rapid shoaling and decay (see Figure 5.23). The waves are not shoaling as expected.

Velocities Within the Plane Beach Breakers

It is clear that the waves here are not breaking as expected, and since some sort of transformation occurs when the waves pass their estimated breaking point, it is appropriate to ask: what transformation actually occurs? In Figure 5.29 the velocities of particles in the crest of a wave are plotted to establish how particles are moving.

In Figure 5.29 the velocities for 2.4 s appear in a pattern similar what is expected. Particles in the crest are the fastest, and are moving toward the beach. In the trough in front of the crest and in the trough behind the crest the velocities are small. At 3.2 s the trough in front of the crest is only one layer of particles deep, and velocities within the crest are similar here, being directed up the beach with comparable magnitudes. The velocities for 3.2 s bear close

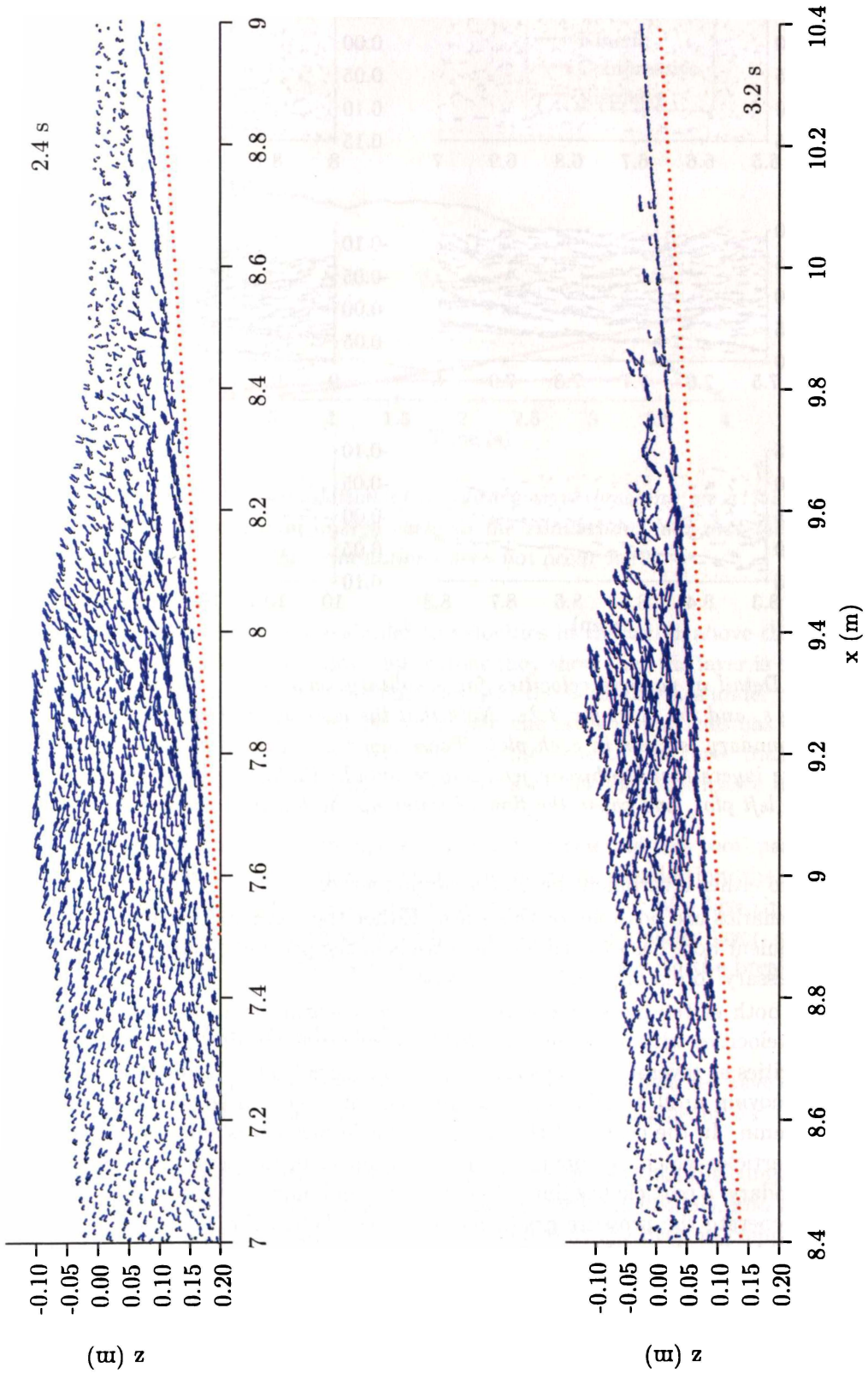

Figure 5.29: *Particle velocities within the wave crest of a solitary wave for the 1:15 beach simulation at 2.4 s and 3.2 s.*

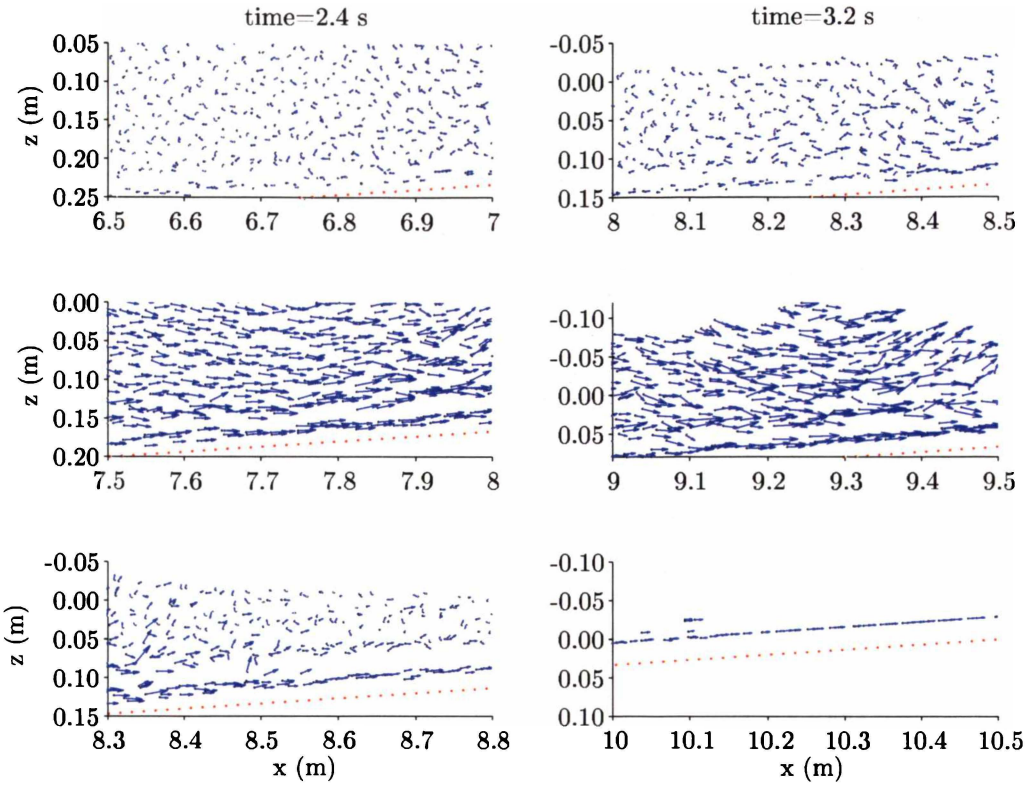

Figure 5.30: *Detail of particle velocities for a solitary wave on a 1:15 beach. The left column are all for 2.4 s, and the right for 3.2 s. Note that the axes are different in each plot. The layer next to the boundary is clear in each plot. These layers are caused by a numerical error, and it is within these layers that toothpaste jets may be found. Toothpaste jets are particularly evident in the bottom left plot, and cause the flow of water up the beach in the bottom right plot.*

resemblance to either a turbulent bore or a surging wave.

So two scenarios are possible for this wave. Either the wave spilled (with a tiny barrel) and at 3.2 s a turbulent bore has evolved, or the wave is in the process of surging. More investigation would be necessary to distinguish between these.

However, both of the plots in Figures 5.29 have a feature that is both surprising and concerning: the velocities within the layer of particles that runs parallel to the beach. For 2.4 s and 3.2 s the velocities at several positions are plotted in Figure 5.30. This layer of particles is caused by particles moving until a sufficient pressure gradient is generated to stop further collapse of the water column. In the centre of the column a single particle is surrounded on all sides by it neighbours, particles which, by moving, can cause changes in the pressure and pressure gradient. Near the boundary, a particle has limited neighbours, and more movement is required from fewer particles to generate the pressure gradients necessary. While the existence of these layers has been noted in the literature (Monaghan, 1994), the jets occurring within them has not.

The existence of the boundary layer has been noted in the literature (Monaghan, 1994) and is also visible in results presented by Dalrymple and Knio (2001). Monaghan (1994) suggests that the boundary layer is caused by the discretised boundary (ghost boundary particles are used). But here the boundary is a normal/tangential boundary, and is not discretised in the same manner as a ghost particle boundary. An explanation for the presence of layers near boundaries would be useful, but does not occur in the literature.

The two top rows of plots in Figure 5.30 do not show anything too surprising: velocities

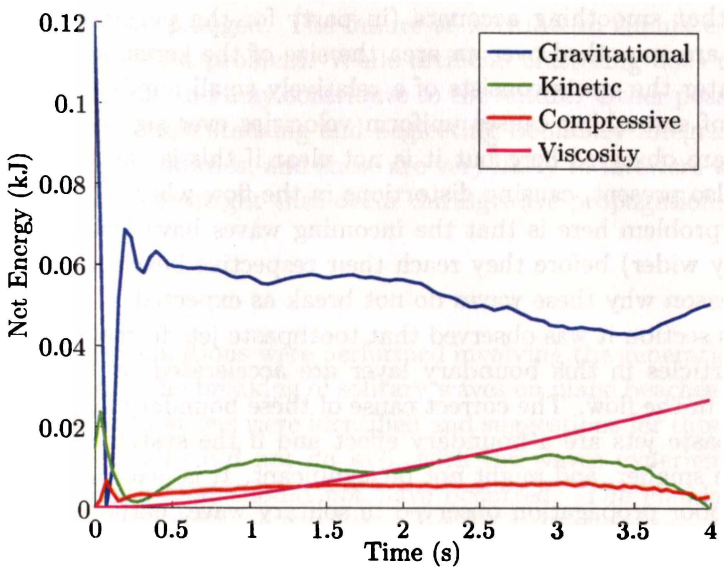

Figure 5.31: *Energies during a simulation of a solitary wave breaking on a 1:15 plane beach. Note the large drop in gravitational energy early in the simulation, and that the large drop in compressive energy observed in other simulations does not occur here.*

within the boundary layer particles are similar to velocities in the water above them.

The bottom row of plots does not show this, rather they show that the layer is being squeezed forward. In the bottom left plot the layer of particles near the boundary lies under particles which have relatively low velocities. In the bottom right plot the layer of particles has been squeezed out from under the wave, forming a jet that flows up the beach. The force that performs the squeezing is provided by the weight of the crest, and is rather like toothpaste being squeezed from a tube.

When simulating breaking waves energy is required to accelerate this toothpaste jet, energy that is extracted from gravitational potential energy of the wave. In addition, the jet is traveling in the opposite direction to the flow expected at the trough in front of a breaking wave, which is back toward the crest. Once a toothpaste jet has formed in a boundary layer, smoothing and viscosity begin to accelerate the particles above, further interfering with the breaking process.

Why These Waves Do Not Break as Expected

The simulations described here of solitary waves breaking on plane beaches have one important feature: the waves did not break as expected. In this section the reasons for this failure are examined.

When examining the propagation of solitary waves it was noted that the compressive energy experienced a marked change, since it was approximately constant initially, and after a certain time it begin to reduce, almost linearly. It was suggested that the drop in compressive energy could be attributed to artificial clustering, and might contribute to the poor performance of the model in simulating propagating solitary waves. This effect cannot account for the failure here, since there is no drop in compressive energy present, as is evident in Figure 5.31.

Here the dominant feature is the drop in gravitational potential energy early in the simulation, a drop due to spurious energies being added to the initial state. Artificial clustering does occur in these simulations; at later times many clusters are observed, and these may contribute to the failure. So it appears that the conclusion drawn earlier, that the cause of the drop in compressive energy is clustering, is incorrect.

It is possible that smoothing accounts (in part) for the peculiar breaking observed here. Particle velocities are smoothed over an area the size of the kernel support. When the wave is in very shallow water the crests consists of a relatively small number of particles, and here the cumulative effect of smoothing causes uniform velocities over significant numbers of particles. Similar velocities are observed here but it is not clear if this is caused by smoothing. Particle stacking may be also present, causing distortions in the flow where the shear is low.

An important problem here is that the incoming waves have lost a lot of height (and have grown considerably wider) before they reach their respective beaches. The loss of height is the most significant reason why these waves do not break as expected.

In the previous section it was observed that toothpaste jets formed along the bottom boundary, where the particles in this boundary layer are accelerated by the wave crest causing an unphysical feature in the flow. The correct cause of these boundary layers is not discussed in the literature. Toothpaste jets are a boundary effect and if the systems contained more particles, the effect would be smaller, and might not be significant. It is possible that toothpaste jets also contribute to the poor propagation observed in solitary waves earlier in this chapter, but since cnoidal waves are not as poor, it is not clear that this is the answer. It is possible that the corrugations that occur when ghost particle boundaries are used would slow the toothpaste jet, reducing the importance of this effect. Small artificial features on the bottom boundary, perhaps just a single boundary point, might also provide enough of a barrier to prevent the toothpaste jets from forming.

A Note on SPH Breaking Waves in the Literature

There are a number of examples of simulations of breaking waves from SPH models in the literature, and it is pertinent to discuss how the results here compare to these.

Dalrymple and Knio (2001) contains a solitary wave generated by the initial state method, a wave that is higher than the maximum stable wave height given the water depth. This wave begins shoaling immediately, so the effects that contribute to the poor results here do not have time to act. In the same publication, a solitary wave generated by Goring's method is presented but the details of how the wave height changes with time are not.

Monaghan (1994) contains a simulation of very short, steep waves (with a wave length of 14.9 m in 5 m of water and a height of approximately 2 m) generated by a piston type wavemaker (it is not clear how the piston motion was determined) breaking on a steep (1:10 slope) beach. Similar simulations have been generated using MARIAN (but are not presented here), but clearly there are limits as to what waves can be simulated.

Typical waves on beaches are far less steep and much longer than Monaghan's wave, and are far smaller than Dalrymple and Knio's wave. To be a useful tool in coastal modelling, SPH models need to be able to simulate all manner of waves, not just extreme examples. The results here are not in disagreement with the literature, since they simulate quite different scenarios.

It is appropriate to mention simulations using MPS models (Koshizuka *et al.*, 1998; Gotoh and Sakai, 1999), in which the breaking of long cnoidal waves is examined. These waves bear much closer resemblance to real swell waves than those of Monaghan (1994) and of Dalrymple and Knio (2001). The MPS waves are not extreme and break with observable barrels (even if they are not always very clear). It is worth noting that the MPS simulations employed density boundaries and do not appear to experience the boundary layers and toothpaste jets observed in this work.

5.4.4 Summary of Solitary Waves on Plane Beaches

The simulations performed here did not show the expected behaviour. The waves either spilled (with barrels irresolvable given the particle numbers and sampling rate used here), or they

surged, when they should have plunged. The failure of MARIAN to simulate the propagation of solitary waves is a very significant problem. While artificial clustering does not cause a drop in compressive energy, it is present and may contribute to the failure. Other possible reasons for the failure include smoothing, particle stacking and neglecting boundary integrals. Toothpaste jets are observed at the bottom boundaries, and these are very likely to interfere with wave breaking, as are the massive losses in wave height that occur during wave propagation.

5.5 Conclusion

In this chapter a number of simulations were performed involving the generation and propagation of solitary and cnoidal waves. The breaking of solitary waves on plane beaches was also examined. Unphysical features of the simulations were identified and suggestions for these were put forward.

The solitary wave simulations did not do well, with the waves experiencing large losses in height and increases in length that should not have occurred. The cnoidal wave simulations showed better performance, but it is not clear why. Artificial clustering was identified as an area where significant improvements could be made. Boundary integrals neglected in the development of the SPH equations may also provide a significant source of error and contribute to poor performances. The role of particle stacking is ambiguous, but toothpaste jets are likely to have a major effect in breaking wave simulations. So while the literature contains discussion of simulations showing some breaking waves, it is clear that SPH models have difficulty in simulating some breaking waves. The viscosity formulation used is also a significant contributor to this problem and is discussed in Section 6.3.4.

Higher particle numbers should offer an improvement to the simulations, however this comes at the cost of much longer run-times, and these are already relatively time consuming simulations to run. Also, any system with a low height to width ratio will always be expensive to simulate when relatively high resolutions are desired, which is the case here.

Chapter 6

Real Beaches

In the previous chapter the generation and breaking of waves using MARIAN was examined. The simulations were restricted to constant depth numerical wave tanks and to plane-slope beaches, but provide valuable insight into the limitations of MARIAN, and the problems it experiences.

In this chapter the use of MARIAN is extended to simulating waves breaking on a real-world beach. The aim is to provide modelling experiences from which an assessment can be made on the suitability of this SPH model for modelling waves in the coastal zone. Such a an assessment needs to be based on a comparison with other modelling methods available.

A local beach, namely Whangamata bar (actually an ebb tidal delta), was selected as an appropriate site where breaking waves could be examined in the field, and also simulated using MARIAN. The bathymetry was surveyed and from this a beach profile extracted. This process is described in the first section of this chapter, and in the second section a discussion is given of the method by which parameters describing the barrels of real breaking waves were collected from the field site. The following section describes the output from MARIAN using the Whangamata bar data. Finally a comparison between the real world profiles and the simulated results is conducted.

6.1 Whangamata Bar

Whangamata bar is an ebb tidal delta that lies at the mouth of Whangamata Harbour, and is indicated on the map in Figure 6.1. Well known to New Zealand surfers as a quality surf break, the bar focuses waves to a peak where they may plunge, if wind and tide conditions are right. An aerial photograph taken some years ago is shown in Figure 6.2, and since it was taken at low tide, the bar is visible. This site was selected because it affords the opportunity to take photographs from the cliff base, while looking down the beach and along a wave crest. Between the photographer and the wave peak is a channel, so unless waves are very large there is still water between the photographer and the peak. An example of the photographs that were taken from this position is shown in Figure 6.3.

Photographs are collected so that details of the wave profiles can be extracted. These details are compared to output from MARIAN from simulations using an appropriate beach profile and input waves. In this section the collection of the bathymetry is discussed, and the swell event for which waves were photographed is described. Finally SWAN (Booij *et al.*, 1996) simulations are described which were conducted to estimate where the waves will break.

6.1.1 The Bathymetry at Whangamata Bar

The bathymetry at Whangamata bar was collected by hydro-graphic surveying using a Knudsen MP 200 depth sounder with a Trimble MS750 RTK GPS, using the associated Trimble Hydropro

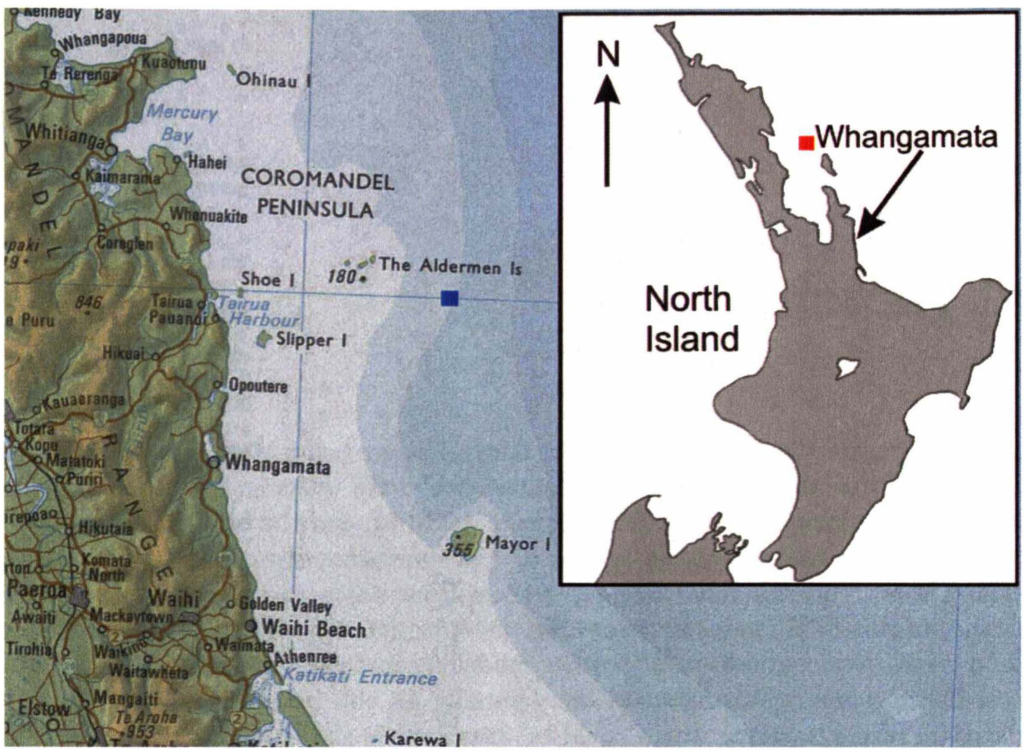

Figure 6.1: *The East Coast of the North Island, New Zealand, showing the position of Whangamata. Reproduced from Infomap 265-1, Department of Survey and Land Information. The red and blue squares indicate the approximate positions of the Mokohinau wave buoy and the output cell for NOAA Wavewatch data respectively.*

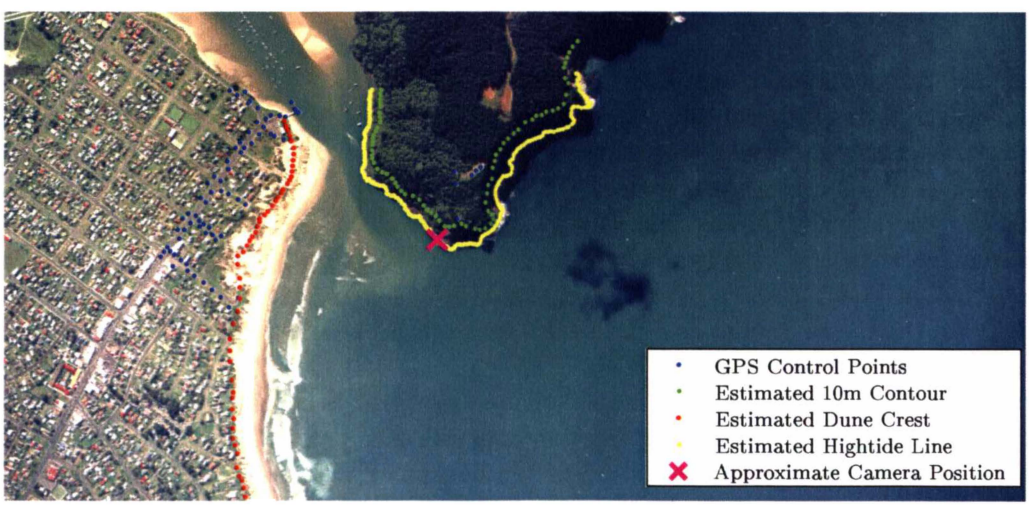

Figure 6.2: *An aerial photograph showing Whangamata town, harbour and bar. The superimposed points are the estimates for the coastlines along the cliff, and also the dune crest. The control points used to determine the estimates are also shown, as is the approximate position from which photographs were taken. Aerial photograph provided courtesy of Environment Waikato.*

Figure 6.3: *An example of the photographs of breaking waves at Whangamata bar taken from the base of the cliff, this one well after the wave first broke. The barrel is clearly present.*

software. A Trimble Pathfinder Pro XR was used for some additional land-based surveying, namely, collecting control points for use in rectifying positions from an aerial photograph.

The hydro-graphic survey was conducted at high tide but was limited by the draught of the survey boat to water deeper than 1 m. In order that SURFER (used to process the surveyed data, Golden Software Inc., 2004) be assured of providing reasonable beach slopes, information is required for bathymetric heights above mean high water. This does not need to be accurate, since approximate values are enough to ensure the beach face is reasonably well represented. Heights of the beach above water were collected using the RTK GPS along the high tide mark. The coast along the cliff was problematic since satellite coverage was sufficient for the RTK GPS only in certain sections along the cliff.

A coastline for the cliff was estimated by digitising positions from an aerial photograph. For this task a number of control points were required, and these were collected using the Trimble Pathfinder Pro XR. This system has a higher uncertainty than the RTK GPS and so a high number (75) of points was used. It became evident during data processing that the points surveyed along the beach were insufficient to provide a reasonable beach face, so additional positions along the dune crest were digitised from the aerial photograph and rectified. The control points and the digitised positions along the cliff and dune are superimposed on the aerial photograph in Figure 6.2.

The rectified coordinates for positions along the coastline at the foot of the cliff required adjustment in order that they fit the surveyed data, since a bias is apparent. The surveyed data and the rectified data were processed using SURFER, and a bathymetry is plotted, with land from the aerial photograph superimposed, in Figure 6.4.

In processing the bathymetric data, SURFER predicted that the depth in the channel was 0 m, while in reality depths of 3 m to 4 m are present. Points were digitised from the aerial photograph that appeared to follow the (somewhat arbitrarily chosen) 2 m contour. These points were rectified and given the depth of 2 m. With these points included the bathymetry has a much more realistic channel. The ***fudged*** data are shown in Figures 6.4 by the yellow dots.

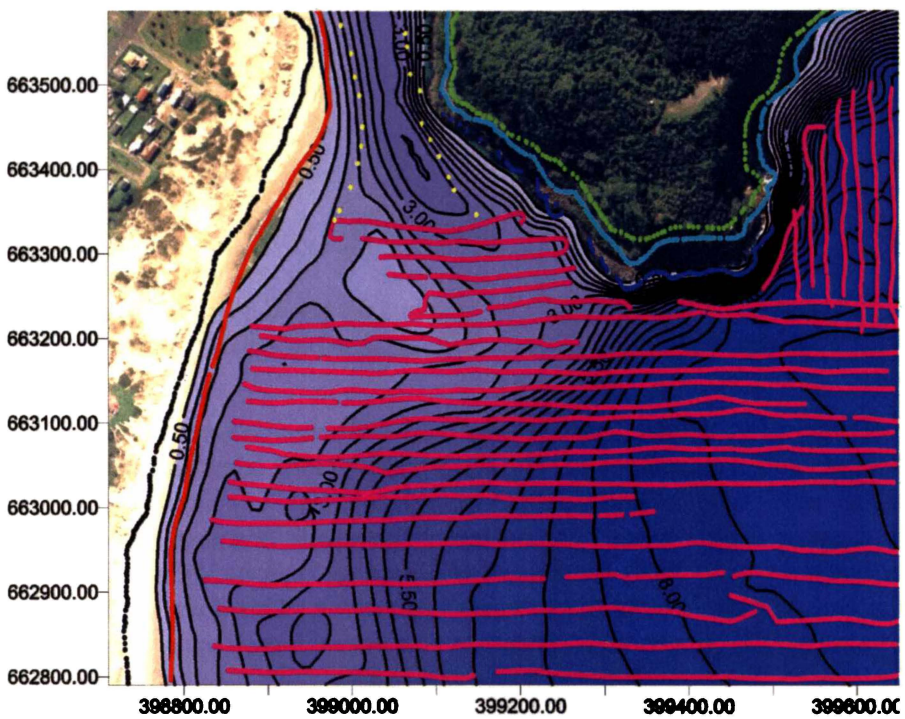

Figure 6.4: *The raw surveyed data, rectified data, processed bathymetry, and the aerial photograph of Whangamata. The black, green and light blue points were rectified from the aerial photo, the red, dark blue and pink points surveyed using the RTK GPS (and the depth sounder for pink). The yellow points were fudged to get a reasonable approximation to the channel. The axes show NZMG Northing's and Easting's.*

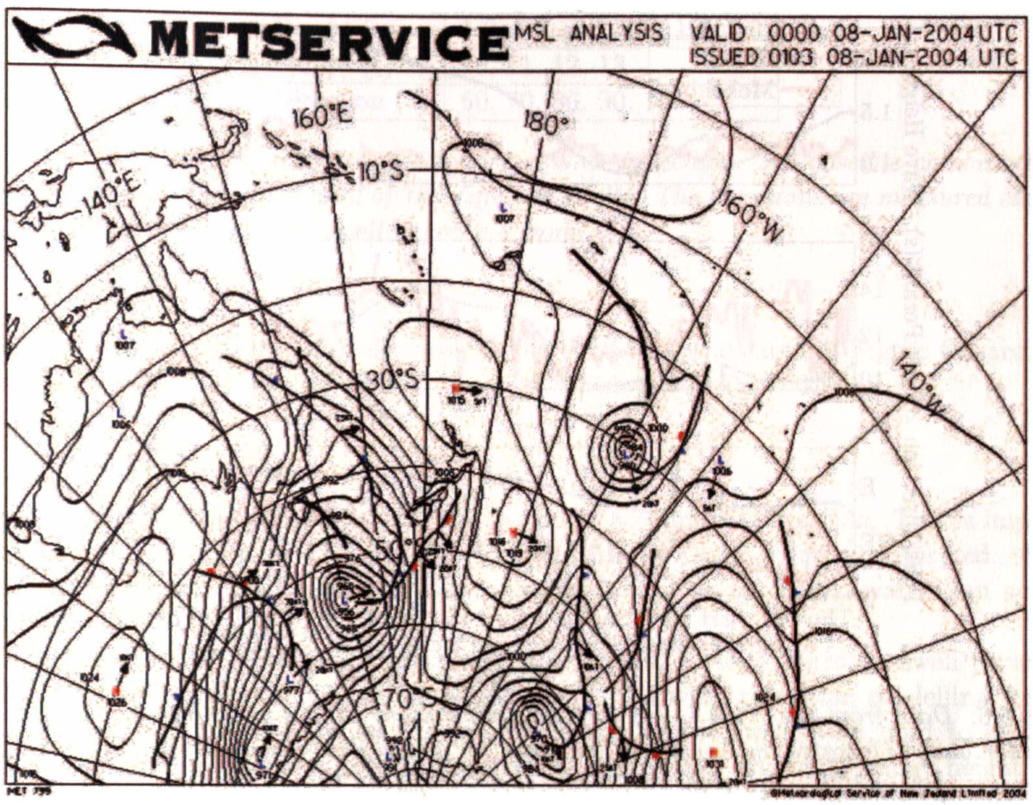

Figure 6.5: *A weather map showing Tropical Cyclone Heta (30° S 160° W), the source of the swell event observed here, the day after it was downgraded to an ex-tropical cyclone. Heta's path is shown by the heavy black line.*

The hydro-graphic survey described here was conducted on 17th September 2002, and the first significant swell event (generated by Tropical Cyclone Heta) when photographs were collected, was 12 January 2004. It was anticipated that the bathymetry might have changed substantially over that period, and a new bathymetry was to be taken after the swell event. However before this additional survey was conducted Tropical Cyclone Ivy generated a swell that was referred to in a local newspaper as being the largest in 30 years. It was decided that such a swell would have had too large an effect on the bathymetry and plans for the second survey were abandoned.

6.1.2 A Swell Event at Whangamata Bar

Whangamata bar receives swells from directions between 10° and 110°, relative to true North. Such swells come typically from tropical cyclones, ex-tropical cyclones and from storms passing over Northland (in the North of the North Island). Swell at Whangamata is relatively rare, and sometimes several months can pass without a significant swell event. Macky and Latimer (1995) determined from wave buoy data that 70 % of the time wave heights are less than 1 m in the region of Whangamata, but this was during the El Nino of 1990-94 which is dominated by offshore winds in the region of Whangamata.

The source of the swell event considered in this study was Tropical Cyclone Heta, for which a mean sea level pressure analysis is presented in Figure 6.5 for the 8th of January, the day after it was downgraded from tropical cyclone status. Given the distance from Whangamata bar to the cyclone, it was anticipated that the swell would be well developed, and have a long period.

Prior to the arrival of the swell at Whangamata bar, it was intended that current meters would be placed to gather detailed information about the swell for which photographs were to

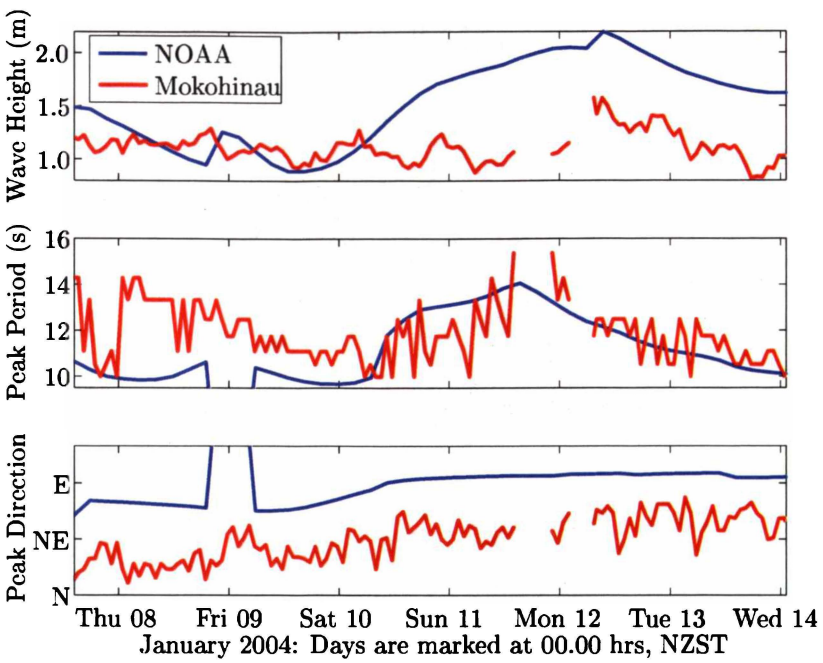

Figure 6.6: *Data from the NOAA WAVEWATCH hindcast (Tolman, 2002) and the Mokohinau waverider buoy (Operated by Auckland Regional Council). Note the missing data from the buoy on Sunday 11th and Monday 12th.*

be taken. However the sea conditions preceding the swell event were not calm enough to deploy instruments. As noted in the previous section, swell events at Whangamata are not frequent, and due to time constraints it was not considered feasible to wait for another, more obliging swell.

There were two alternatives available for wave data, these being data from the Mokohinau Island Wave Buoy (operated by Auckland Regional Council) in the Hauraki Gulf, and NOAA wave hindcast data. Data from these sources is plotted in Figure 6.6 and their approximate positions are shown in Figure 6.1. The storm swell is clear in the NOAA data, with a marked increase in the period, and a corresponding increase in the wave height.

Data from the buoy and the hindcast agree reasonably well, both showing peaks in wave height and period occurring at the same time. But there are notable differences in the wave heights and the wave direction. The Mokohinau buoy is sheltered when waves are from the south east or south, but here the hindcast and the buoy records show a swell from the East and North East, so sheltering cannot be significant. The buoy is much further North than Whangamata, so some difference in the direction of the swell might be expected, but this should show the buoy reading a swell shifted in the other direction, that is, more to the south east. The difference may be explained by the large distance which the swell has traveled, and since Heta was far from land, the accuracy of the hindcast may be low.

It was predicted (by www.buoyweather.com, which uses NOAA WAVEWATCH forecasts) that the swell would arrive late afternoon on Sunday 11th of January, and that it would peak in height at midday, Monday the 12th. However local residents reported that the swell peaked on Sunday the 11th, and that the waves had dropped to about half the size by Monday, reasonably consistent with the buoy data and the hindcast which both show the swell peaking at midnight on Sunday 11th. The wind was a light onshore at midday Monday, but had turned to a light offshore by 1pm.

So there is some uncertainty as to what swell parameters best describe the swell seen at

Height (m)	1.0, 1.2, 1.4, 1.6, 1.8, 2.0
Period (s)	10, 11, 12, 13
Direction (°)	60, 70, 80, 90, 100

Table 6.1: *Parameters describing swells used in* SWAN *modelling. All possible combinations of these variables were used, for a total of 120 different swells. The directions are measured clockwise from North and state where the swell is coming from.*

Whangamata bar. A number of swells close to the conditions indicated in Figure 6.6 are examined, swells for which height, period and direction are shown in Table 6.1.

6.1.3 SWAN Modelling

Because MARIAN is a two dimensional model, it cannot account for refraction. This is important at Whangamata bar and in order that it is accounted for, SWAN modeling is performed. SWAN is a numerical model that can be used to simulate wave refraction in coastal waters, an acronym of Simulating WAves Nearshore.

The bathymetry for Whangamata bar is used as input to SWAN, with the swell parameters from Table 6.1, in order to simulate wave refraction. The wave refraction modelling was performed in order to generate estimates of velocity fields for the swell event, and to determine where the wave would first break, the position of the wave peak. Figure 6.7 shows height/depth predicted by SWAN for one of the swells from Table 6.1. The position of the peak was estimated using Equation 2.6, but as noted in Chapter 2, this is very approximate. There are only slight variations in the peak positions for the swells and these are averaged to produce a mean peak position. This is plotted in Figure 6.9 as a black square.

The velocity fields for a swell predicted by SWAN can be used, with the *stream2* function in MATLAB, to determine the wave orthogonals: stream lines along which velocities are always tangential. The wave orthogonal closest to the mean peak position is used to generate the beach profile for use with MARIAN. Figure 6.8 shows the wave orthogonals for the swell with a wave height of 1.4 m, a period of 13.0 s and coming from 70°. The focusing effect of the bathymetry is clear.

For each of the 120 swells in Table 6.1, the wave orthogonal that passed through the mean peak position was found. These are referred to as *peak orthogonals* and are plotted in Figure 6.9.

For each of the swells in Table 6.1, the depths along the peak orthogonal are plotted against distance along the orthogonal in Figure 6.10, the resulting curves being beach profiles. Because the distance along the orthogonal is a relative quantity, all of the beach profiles have been normalised at a point that was chosen to minimise the difference between the profiles on the central section.

All of the beach profiles are similar, in that they start with a flat section of about 8 m depth (slope $\approx$ 1:320), followed by a sloping section (slope $\approx$ 1:22), and finish with a gently sloping section (slope $\approx$ 1:77). Because there is uncertainty here in the exact nature of the swell, and there is variation for individual waves, no profile is correct for all waves, so a typical profile was selected, shown in Figure 6.10 by the green curve. This typical profile is used in the MARIAN simulations.

6.2 Breaking Waves at Whangamata Bar

In the previous section the methods used to create a beach profile for Whangamata bar were described, a profile that is be used to simulate waves breaking with MARIAN. To provide a

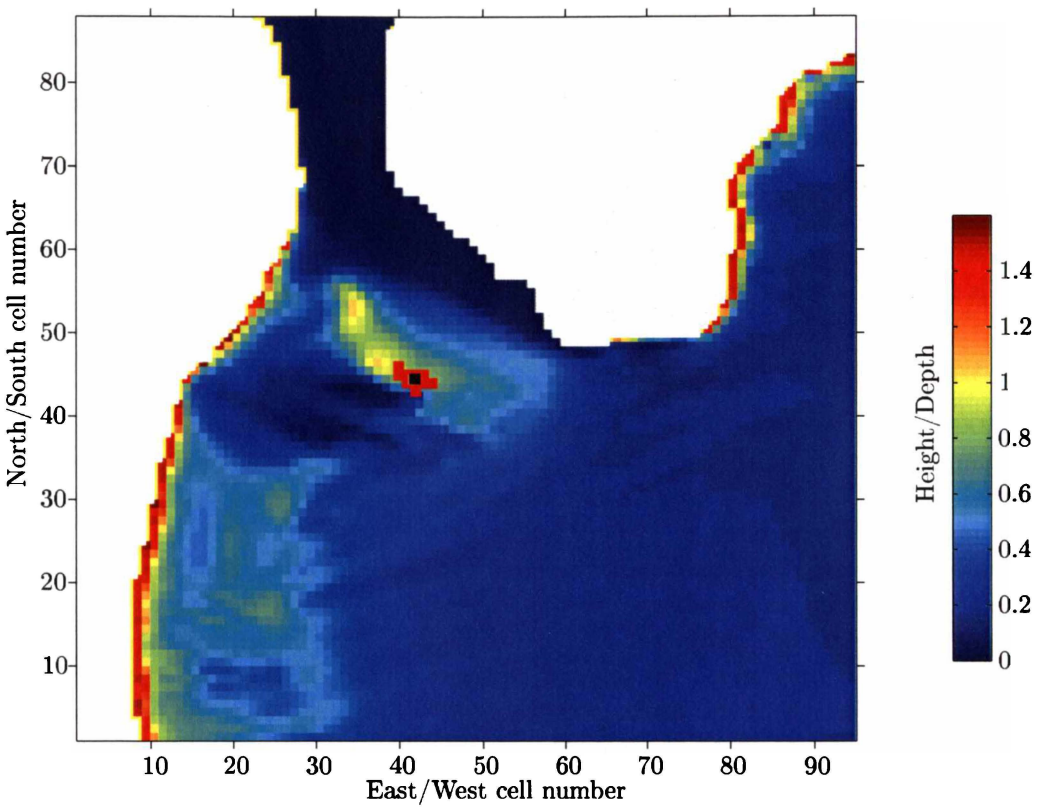

Figure 6.7: *Wave height to depth ratio from the* SWAN *simulation having an incident swell with a height of 1.4 m, a period of 13.0 s and coming from 70°. The red squares near the centre are all of the peak positions and the black square is the mean peak position. The peak positions are placed at the most seaward cell for which the height to depth ratio is greater than 0.78.*

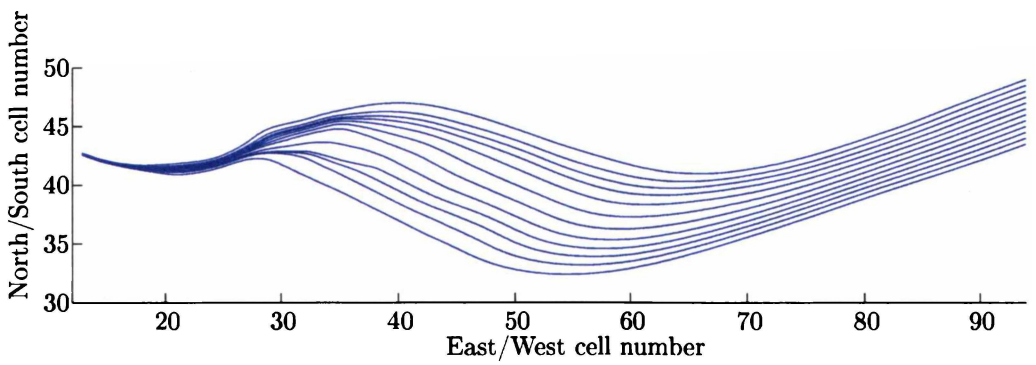

Figure 6.8: *Wave orthogonals for the* SWAN *simulation having a swell with a wave height of 1.4 m, a period of 13.0 s and coming from 70°.*

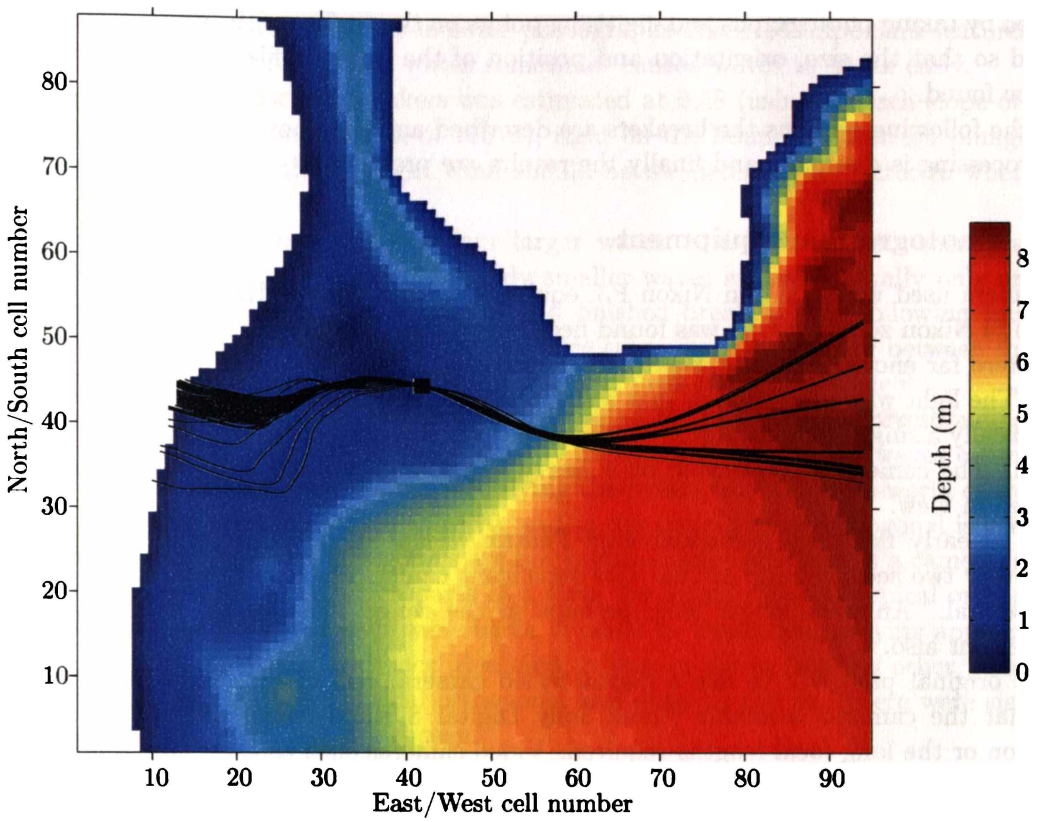

Figure 6.9: *Peak orthogonals from* SWAN *simulations for all swells, superimposed over a plot of depths. The mean peak position is marked with black square.*

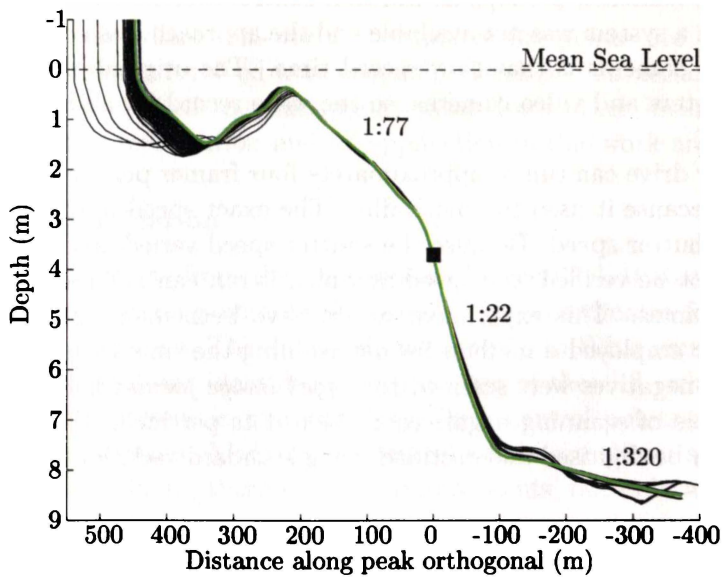

Figure 6.10: *Depths along the peak orthogonals, from* SWAN *simulations for all swells. The black square indicates where the orthogonals were normalised and the green curve shows depths along the typical orthogonal.*

comparison with the simulations, the details of real breaking waves were desired. These were obtained by taking photographs and digitising points on the surface of the waves, points that were rectified so that the size, orientation and position of the wave profile in real-world coordinates could be found.

In the following sections the breakers are described and the photography are examined. The post-processing is discussed and finally the results are presented.

6.2.1 Photographic Equipment

The camera used was a 35 mm Nikon F3, equipped with a Nikon MD-4 motor drive and a 80-200 mm f4 Nikon zoom lens. It was found necessary to use this at 200 mm all the time, since the waves were far enough away that a shorter focal length would have the waves too small in the frame. The light was good enough that a shutter speed of 1/250th or faster was always possible.

Typically a single wave would be photographed for several seconds, from just before it started to peak. The camera was mounted on a heavy tripod and was panned in order that the wave remained in view.

On an early field trip a manual-wind Pentax was used, but this could manage only one frame every two seconds, and at that rate significant camera shake occurred, so the motor drive was essential. An even longer lens, perhaps in the range 300-400 mm, would have been an improvement also.

The original plan was to use a digital video camera, but in the first field trip it became clear that the cameras available (three Sony Digital 8 Handycams) did not have either the resolution or the long focal lengths required. Video cameras such as these could have been used with a waterproof housing if the photographer had been prepared to get wet, but this approach would have meant a moving camera position. A stationary camera position was desired because rectifying images would be easier.

Three video cameras were to be used in order that positions on a single wave could be identified from three different viewpoints, so it would have been possible to accurately determine the coordinates of the wave. Using this technique with 35 mm cameras would require a method for synchronising the cameras, perhaps an infrared control system, such as those used by studio photographers. Such a system was not available and the approach was abandoned. An additional merit in using video cameras is that they record time. The original experimental plan was to synchronise wave meters and video cameras, so the wave record for an individual wave could be matched to footage.

The MD-4 motor drive can run at approximately four frames per second, but it was not used at this high speed because it used too much film. The exact speed of the motor drive is in part determined by the shutter speed. Because the shutter speed varied, and because the rate of the motor drive could not be verified, the speed at which it ran can not be used as an estimate for the time between frames. This experiment would have been much improved, providing more data, had the author employed a method for determining the time each frame was taken.

Once developed, negatives were scanned to *tagged image format* files (tiff files) using an HP Precisionscan, capable of scanning negatives at 2400 dots per inch. With images in this form, desired positions can be digitised and rectified using standard techniques.

6.2.2 The Waves on the Day

The experiment began at around 12.30, Monday the twelfth of January, 2004. At this time the wind was a light onshore, the tide was mid way and ebbing, the breakers were 1.5-2 m at the peak and were spilling. According to local knowledge, low tide provides the best surf at Whangamata bar, since the waves are steeper, and more likely to plunge. At around 1 pm the wind changed to a light offshore, also contributing to the likelihood of waves plunging. The last photographs

were taken at around 2 pm. The waves drew a reasonably large crowd of surfers, between 20 and 30 at any time, and their presence in some photographs obscured important features of the waves. In addition, their antics in the water sometimes caused waves to break early.

The Iribarren number for the breakers was estimated at 0.48 (using a beach slope of 1:22, a wave height of 1.5 m and a wavelength of 170 m), right on the boundary between plunging and spilling. It is not surprising that a slight wind should be the determining factor in whether the waves spilled or plunged.

The swell comprised sets of three to four larger waves, and between sets there were lulls lasting up to several minutes, during which only smaller waves arrived. Usually only one wave could be photographed in a set, since once it had finished breaking, the following waves had already started to break. A total of 41 sequences were taken, each containing between three and fourteen frames of the same wave. The lulls grew longer later in the experiment.

In analysing the sequences, the aim is to extract wave heights and the corresponding barrel shapes, orientations and positions. This is done by digitising points on the waves estimated as lying on the peak orthogonal, and then by rectifying the points to obtain real-world coordinates. The peak orthogonal varies from wave to wave, and so the typical wave orthogonal is used as an approximation. Of course peak orthogonals are not visible through the lens of a camera, so there is a significant uncertainty as to which parts of a wave are actually on the typical orthogonal.

However where the peak of a wave was visible, it could be considered as lying approximately on the typical wave orthogonal, and once identified, it was an aid in locating other points. The process of locating points on the typical orthogonal was done by eye and there were significant uncertainties in where the points are chosen.

A Sequence of a Breaking Wave at Whangamata Bar

A good example of a sequence of frames showing a single breaking wave is shown in Figures 6.11 and 6.12.

6.2.3 Rectification of Photographs of Breaking Waves

Rectification is the process where real-world coordinates for points are extracted from photographs. While the techniques involved were originally developed for aerial photographs (Baker *et al.*, 1980), they can be applied to photographs taken from other inclinations. Here a few important ideas relevant to rectification and its application in this work are discussed.

Control Points for Rectification

In order that digitised positions from the photographs be rectified, it was necessary to gather control points. Collecting control points involves finding the coordinates for positions that can be later identified in photographs. This involved another trip to Whangamata, where the RTK GPS was used to record positions identified from the base of the cliff (where the photographs were taken from) as being present in photographs. These control points are shown in Figures 6.11 to 6.12 by the yellow crosses. Many of the control points are shown several times, once in each of a number of frames. In all, there were 47 control points, but only a fraction of these are visible in any single frame.

An assistant carried the RTK GPS along the beach with a target attached. This target was readily visible (using a camera with a long lens to sight with) from the base of the cliff. Positions were marked on prints of photographs, and from this record the control points could be identified in other photographs. Positions were selected to be control points according to how recognisable they were, usually where lines intersect, such the line through the edge of a house and the top of the dune.

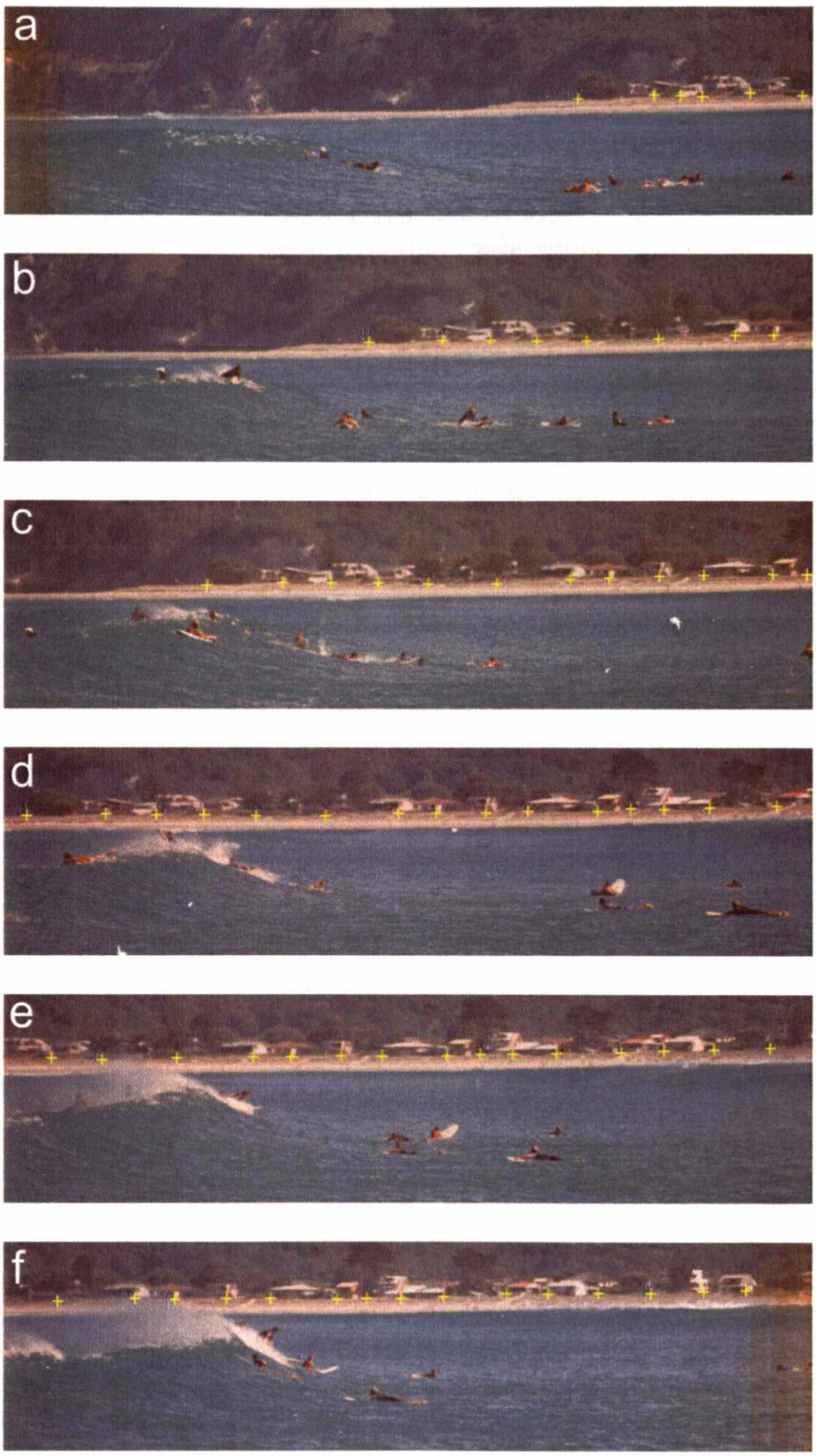

Figure 6.11: *The first six frames in a sequence showing the same wave at different stages of breaking. The frames are roughly 1/2 s apart, and the letters indicate the order. Figure 6.12 contains the last six frames from the sequence. The yellow crosses are the positions of control points used for rectification.*

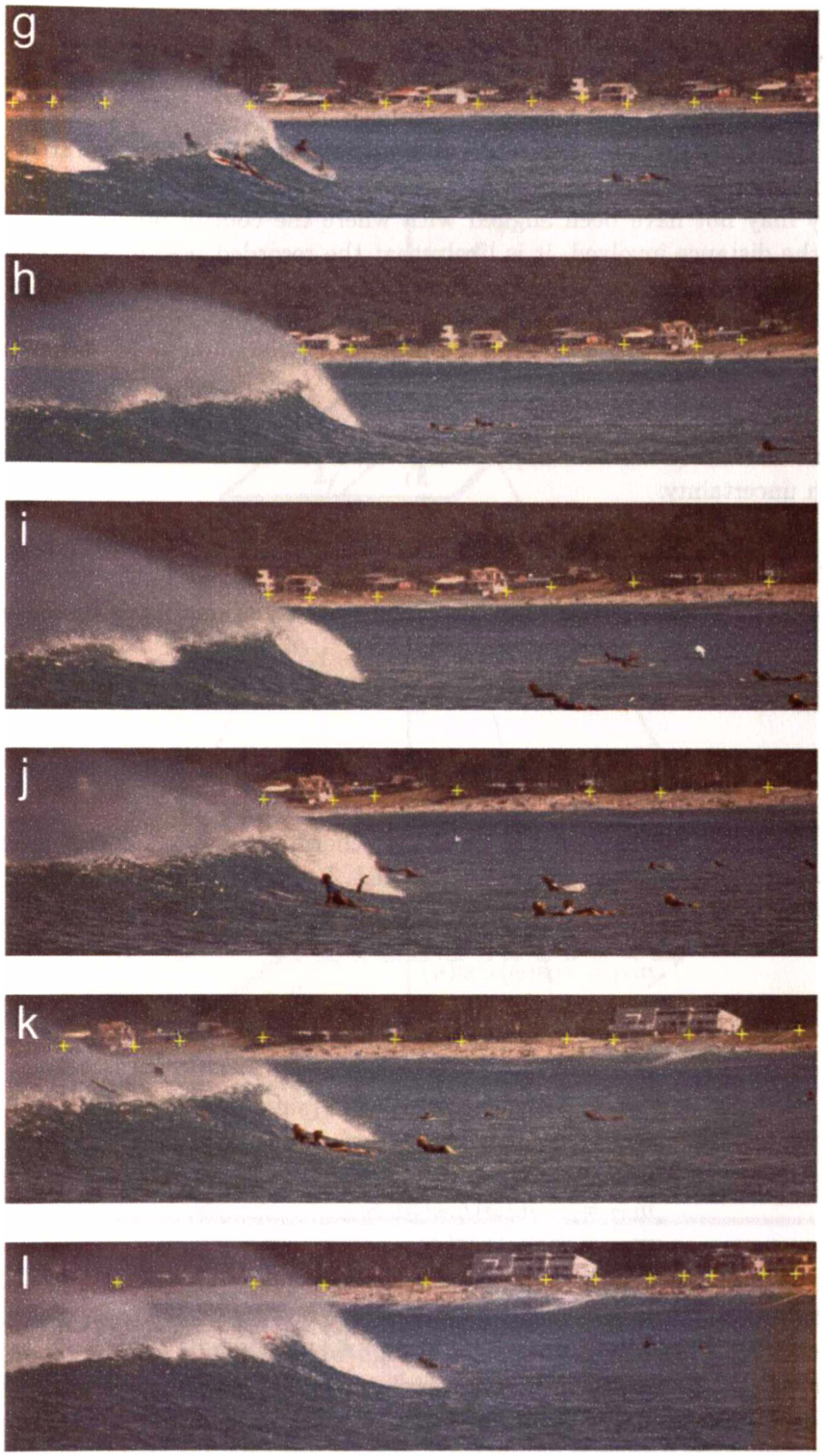

Figure 6.12: *The last six frames in a sequence showing the same wave at different stages of breaking. The frames are roughly 1/2 s apart, and the letters indicate the order. Figure 6.11 contains the first six frames from the sequence. The yellow crosses are the positions of control points used for rectification.*

Quite clearly in Figures 6.11 and 6.12, the control points all exist along a narrow band, which, for rectification, is not ideal because the angle about that line cannot be resolved. Because the camera was panned when photographs were taken, and it is necessary that control points be identifiable in each frame, it was not feasible to collect control points using a boat, even though that would have provided a better spread of points.

The error generated by the RTK GPS is negligibly small, being accurate to within 5 cm, but other sources of error are not. While positions recorded using the RTK GPS are as good as exact, they may not have been aligned with where the control points are believed to have been. Given the distance involved, it is likely that the recorded positions were out by as much as several metres in the horizontal. Also important are the errors generated by the incorrect digitisation of control points from photographs, which will cause errors in the region of several metres at the control point. Typically the number of control points in any frame is low, between nine and fifteen.

The control points here are not good since they are not well distributed around the points of interest (the barrels of the breaking waves), there are not many of them, and they have a relatively high uncertainty.

The Collinearity Equations

The process of rectification is governed by the collinearity equations, given by (Baker *et al.*, 1980)

$$x_i - x_p + f\frac{m_{11}(X - X_i^C) + m_{12}(Y - Y_i^C) + m_{13}(Z - Z_i^C)}{m_{31}(X - X_i^C) + m_{32}(Y - Y_i^C) + m_{33}(Z - Z_i^C)} = 0 \tag{6.1}$$

and

$$y_i - y_p + f\frac{m_{21}(X - X_i^C) + m_{22}(Y - Y_i^C) + m_{23}(Z - Z_i^C)}{m_{31}(X - X_i^C) + m_{32}(Y - Y_i^C) + m_{33}(Z - Z_i^C)} = 0 \tag{6.2}$$

in which f is the focal length, x_i and y_i are image coordinates, X, Y and Z are the real-world coordinates of a point of interest, and X_C, Y_C and Z_C are the real world coordinates for the camera position. These variables are shown diagrammatically in Figure 6.13. The elements of m are

$$\begin{aligned}
m_{11} &= \cos(\phi)\cos(\kappa) \\
m_{12} &= \cos(\omega)\sin(\kappa) + \sin(\omega)\sin(\phi)\cos(\kappa) \\
m_{13} &= \sin(\omega)\sin(\kappa) - \cos(\omega)\sin(\phi)\cos(\kappa) \\
m_{21} &= -\sin(\phi)\cos(\kappa) \\
m_{22} &= \cos(\omega)\cos(\kappa) - \sin(\omega)\sin(\phi)\sin(\kappa) \\
m_{23} &= \sin(\omega)\cos(\kappa) + \cos(\omega)\sin(\phi)\sin(\kappa) \\
m_{31} &= \sin(\phi) \\
m_{32} &= -\sin(\omega)\cos(\phi) \\
m_{33} &= \cos(\omega)\cos(\phi).
\end{aligned} \tag{6.3}$$

Figure 6.13 shows how the three different coordinate systems are related: real-world coordinates, the image coordinates, and the camera coordinates. The last of these has its origin at the *perspective centre* for the camera, the point through which all light passing through the camera travels, a point that is also taken to be the camera position. If a light ray travels through a lens and camera without deviation, it is on the camera axis, and the intersection of this line with the image plane is called the principal point.

The collinearity equations (Equations 6.1 to 6.3) can be used to do several things. If enough control points (at least six) are identified in the real world and in the image, both the camera position (X^C, Y^C and Z^C) and orientation (ω, ϕ and κ) can be found. If the camera position is known, the collinearity equations can be solved for the orientation. Once the camera position and orientation are known, points digitised from the image can be rectified, where the real-world coordinates for the points are determined.

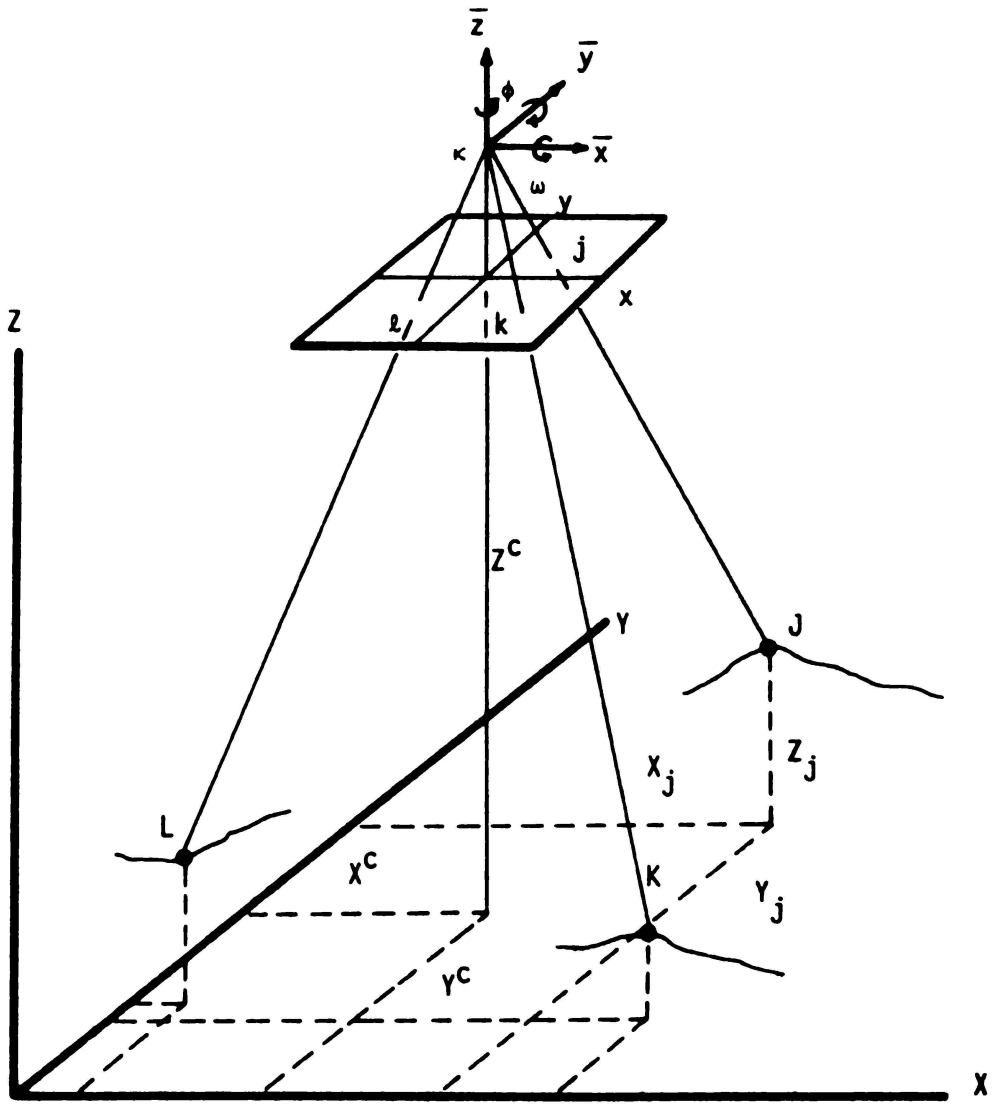

Figure 6.13: *The coordinate systems and parameters involved in rectification of photographs.* X^C, Y^C *and* Z^C *are the coordinates for the camera in real world coordinates, points of interest are labeled* J, K *and* L, *and for example, the real-world coordinates for point* J *are* X^J, Y^J *and* Z^J. *The skewed rectangle high in the diagram is the image plane, with image coordinates* x, *and* y. *The axes with coordinates* $\bar{x}$ *etc. are camera coordinates and* ω, ϕ *and* κ *are rotations with respect to real world coordinates. From Baker* et al. *(1980).*

Lens Calibration and the Principal Point

Lens calibration is the process by which distortions generated in a image (tangential, radial, barrel and pin-cushion distortions) are identified, and hence images can be corrected (Baker *et al.*, 1980). In fact this process calibrates both the camera and scanner. The calibration process reveals the *internal* camera parameters, which quantify the types and degree of distortions present. In addition the principal point can be calculated, the point on the negative coincident with the camera axis.

Here the camera was not calibrated since the errors in the rectification process (caused by the poor control points) far outweigh improvements from correcting images. This means, however, that the principal point is unknown. On many aerial photographs *fiducial marks* (black triangles at the border) indicate the principal position, but the camera used here does not have the facility to generate these. The principal point can, however, be estimated as being near the centre of the negative (Wong, 1980). This is the approach used here.

The Camera Position and Orientation

The position of the camera is required for the rectification process. Using the real-world positions for the control points and the positions digitised from photographs, the position and orientation of the camera can be calculated. This is achieved by solving the collinearity equations, Equations 6.1 to 6.3, and a method for doing this is presented by Baker *et al.* (1980).

The camera position is estimated to be 399285 E, 663305 N, a position that can be used as a starting value when finding solutions to the collinearity equations. There is a significant uncertainty in this position, being about 2 m in the direction tangential to the shoreline, and about 5 m along the shoreline. When photographs were taken (between mid and low tide) occasional waves would wash onto the rocks and onto the feet of the author. The tripod is approximately 2 m high and the tide was in the region of 1.5 m below mean high water, so the height of the camera is estimated to be 0.5 m above mean high water.

The camera orientations ω, κ and ϕ, are approximately 90 °, 180 ° and 0 ° respectively. These approximate values can be found by referring to Figure 6.13 and Figure 6.2, the aerial photograph. First, looking straight down on the aerial photograph, ω is the angle needed to look at the horizon (about 90 °). The photographs were taken with the film negative approximately vertical and with the top and bottom edges approximately horizontal, so ϕ is approximately zero. A value for κ of about 180 ° is required to ensure that the camera is facing in the correct direction, approximately South.

For each of the 27 selected photographs, the position and orientation can be found, and the average of these positions is 399242 E, 663270 N and -17.1 m. All calculated camera positions were between 1.6 m and 74 m from the mean position. The mean position is within 100 m of the estimated camera position but is in the middle of the harbour channel, moreover, it is not just wet, but at approximately 13 m below the bottom of the channel, it is positively subterranean.

Figure 6.14 shows the distribution of camera positions calculated here, with the average camera position shown also. It is clear that there is much less variation in the calculated positions along the shoreline, than there is tangential to the shoreline. This is to be expected, and is a result of the approximately linear distribution of control points in space. A better distribution of control points would improve the estimated camera position significantly.

6.2.4 Barrels and Heights at Whangamata

Two pieces of information are extracted from the photographs, namely, the wave height (at the waves highest point) and the barrel dimensions. The photographs from Whangamata are examined and those frames containing clear views of the barrels are identified. Twelve waves

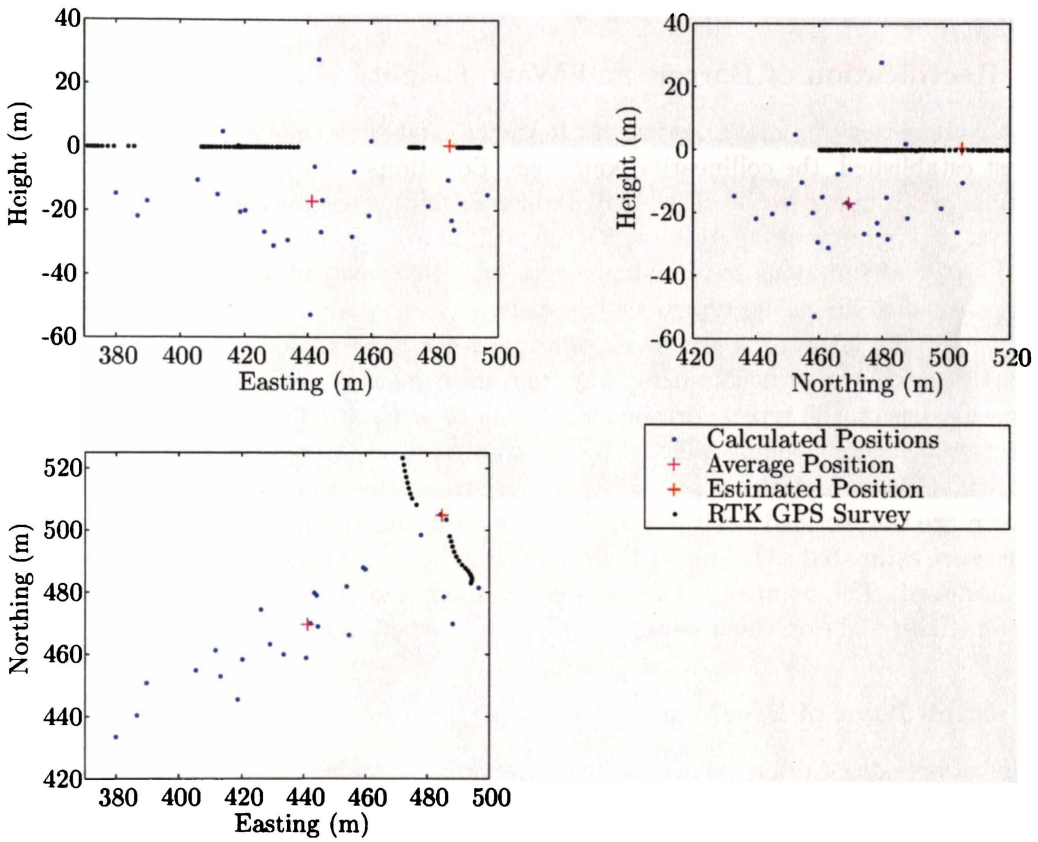

Figure 6.14: *Estimated, calculated and average camera positions. Positions surveyed along the shoreline at the foot of the cliff are also shown. To ensure simple tick labels, Northing's are expressed relative to 662800 m and Easting's relative to 398800 m, using NZMG. Heights are with respect to mean high water.*

fitting this criteria are present, each having one or two frames with good views of the barrels. The points judged as lying on the peak orthogonal were digitised and then rectified. Figure 6.15 shows an example of the digitised points, superimposed on the scanned photograph. Longuet-Higgins' cubic (Equation 2.1 with a rotation and translation) is fitted to the data for the barrel, so the length, width, angle and position of the barrel are identified.

When digitising points around barrels, streaks of foam and wavelets on the faces of the waves are used for reference. As a wave starts to shoal, water is accelerated up the face of the wave, and features like streaks of foam and wavelets become stretched in the direction the wave is propagating. Also, foam streaks are often quite long, having been generated by the preceding breaking wave. These features can be used to identify points that lie in the same vertical plane. This process is very subjective, and has a high uncertainty.

6.2.5 Rectification of Barrels and Wave Heights

Once the camera position and orientation are known, and with the image coordinates of a point of interest established, the collinearity equations (Equations 6.1 to 6.3) can be written as a pair of equations, each linear in the three real-world coordinates for the point of interest. Typically, when aerial photographs are used to rectify the positions of various points, the height is specified, and so the pair of equations have a single solution. Here, instead of specifying the height, the point is required to lie on the typical orthogonal.

It is anticipated that the horizontal coordinates for the rectified points are good along the typical orthogonal, while the distances away from the camera are only as good as the assumption that the wave lies on the typical orthogonal. It can be expected that the heights of the rectified points are poor, being strongly affected by the choice of the camera position.

Figure 6.16 shows a plot of depths along the typical wave orthogonal and the rectified barrels. Also shown are the rectified positions of the wave crest and trough, digitised from images when the waves were estimated as having at their maximum height. In the following sections these data will be discussed. The point of breaking is established, and the height of the wave at breaking is determined. In addition the orientation, shape and position of the barrel is discussed.

Waves at the Point of Breaking

The point of breaking can be defined as being the point at which the wave is highest, something which can be estimated by comparing frames from the sequences. Once a selection has been made, the relevant point digitised and rectified the point of breaking is determined. The height of the wave is calculated with respect to the trough, which also needs to be digitised and rectified.

The breaking points are digitised for twelve waves as are the crests and troughs of the waves. The choice of how the camera position is calculated (or what it is) significantly affects the breaking point and the wave heights. The breaking heights and points are calculated for the twelve waves using each of the three camera position options, and are plotted in Figure 6.17 for all 36 combinations.

Irrespective of the camera position, the rectified data has all of the waves breaking beachward of 360 m, but no further than 440 m. This is right above the point in the bathymetry where the 1:77 sloped section meets the steeper 1:22 sloped section. So it seems that the waves start shoaling over the 1:22 sloped section, some may break over the gentler-sloped section.

When the average camera position is used (and the orientation calculated for each frame) and when both the orientation and position are calculated for each frame, the resulting heights seem a little on the small side. Judging by eye, most of the photographs show waves higher than 1.5 m. However for these two approaches to finding the camera positions, the waves are 17 m and 27 m below mean sea level respectively. When the estimated camera position is used, the

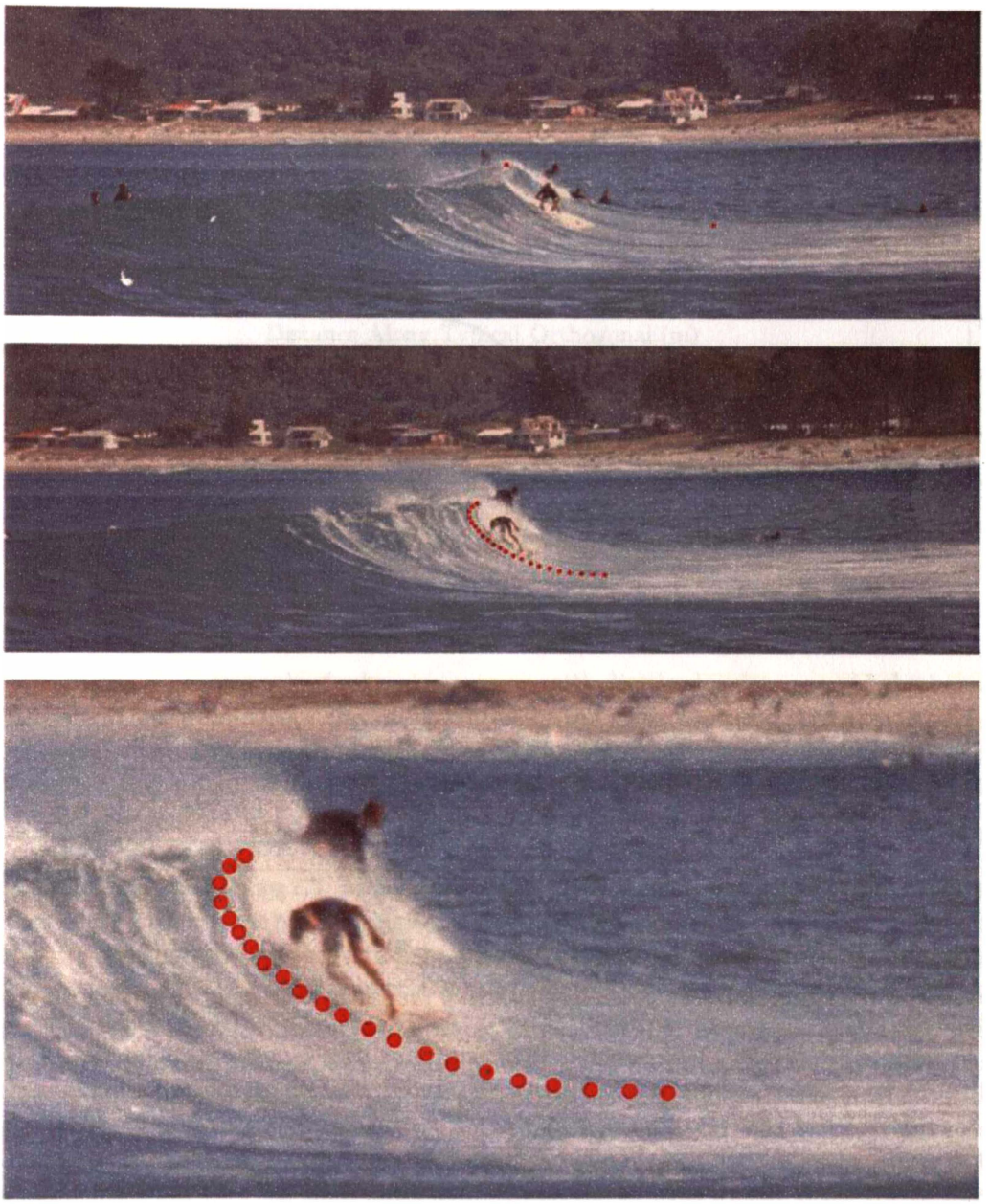

Figure 6.15: *An example of a frame selected for digitising wave height (top), a frame showing the barrel and digitised points a short time later (middle), and detail of the barrel (bottom).*

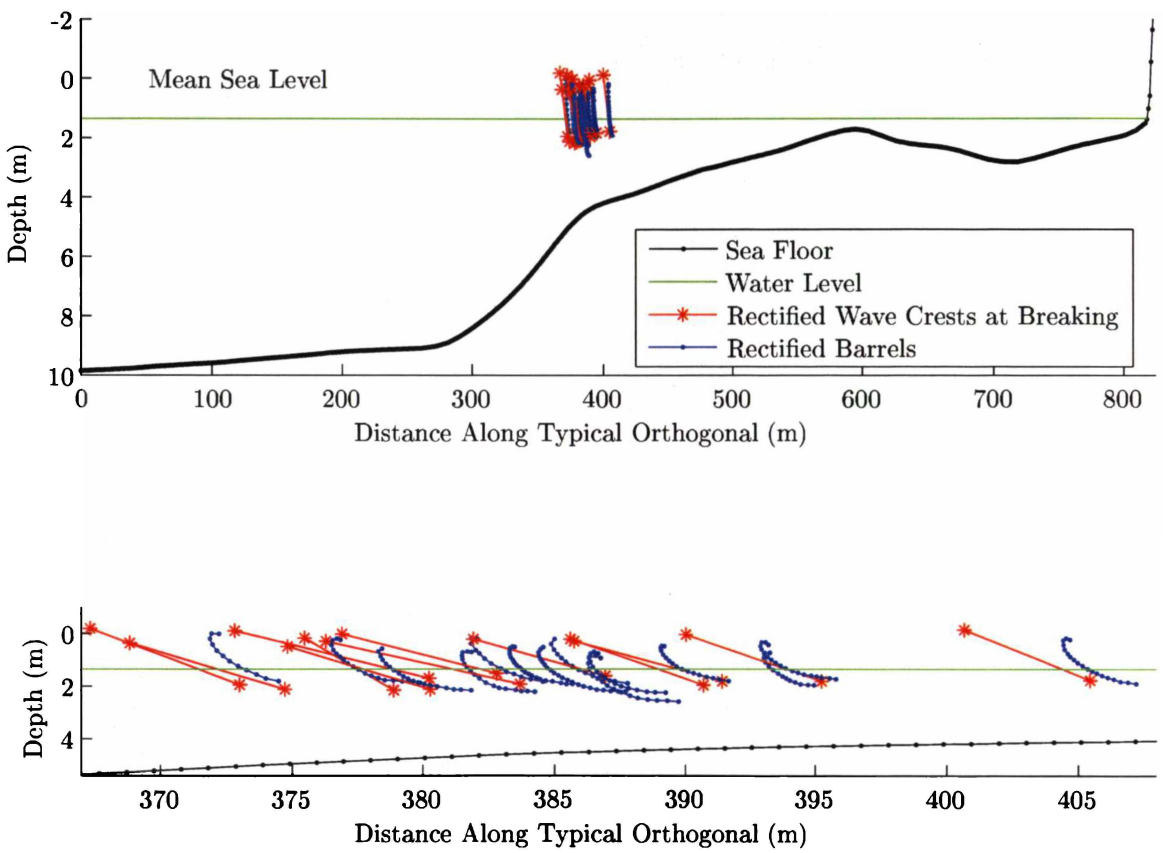

Figure 6.16: *Barrels and wave heights rectified using the estimated camera position, and orientations calculated for each frame. The horizontal coordinate is distance along the typical orthogonal and the vertical is depth with respect to mean high water. The top plot shows the rectified barrels and waves heights on the same plot as the depths along the orthogonal. The lower plot shows details of the rectified objects. Each red line is drawn between two points, one at the crest and the other in the trough. The wave height is the vertical distance between the two.*

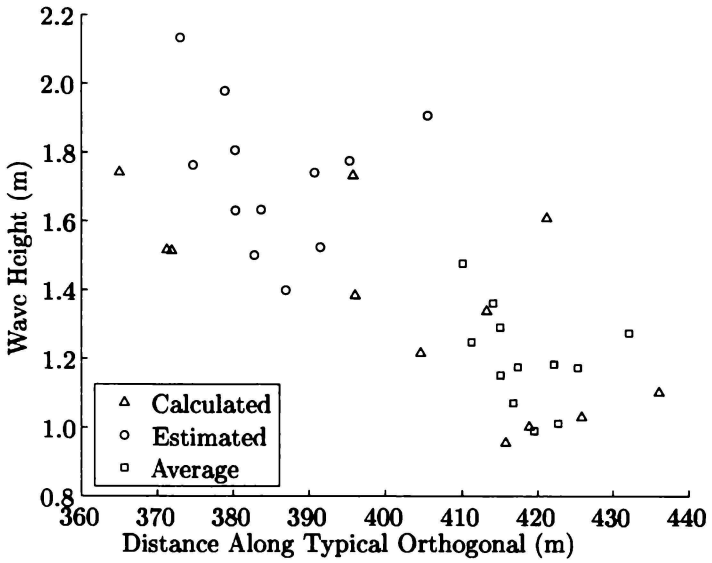

Figure 6.17: *Breaking points and wave heights for twelve waves, digitised from photographs and rectified using each of three camera position options (calculated, estimated and average).*

waves are predicted as being very close to mean sea level, and wave heights are in agreement with estimates.

Longuet-Higgins Cubic and Barrels at Whangamata Bar

Figure 6.18 shows a photograph of a breaking wave, with digitised points around the barrel, and the cubic curve that was fitted. The points in black were also digitised, but it was found that including these when the curve was fitted meant the cubic did not match the barrel so well. For some of the photographs, the fitted cubics did not match the barrels well. It seems that it is necessary to have at least some points in the lip of the barrel. Also, too many points in the trough cause poor curves.

Rather like that in Figure 2.5, Mead and Black (2001) chose photos for which points could be digitised right around the barrel. In all of the photographs here, the barrels were obscured to some extent, by either the lip or by the shoulder of the waves. Many of the photographs used by Mead and Black (2001) will have been taken by photographers floating in the water, not very far from the barreling section of the wave, so they did not have the same problem. Using this approach at Whangamata would be an improvement, but would require that the control points be better distributed. This could be achieved by placing clearly visible markers on the cliff face, so taking photos looking approximately North.

Given the somewhat subjective approach used here, of requiring that the fitted curve *look correct*, data extracted here has high uncertainties associated with it. For those waves where the fitted cubics match the barrels well, the orientation and dimensions can be identified. These are presented in Table 6.2.

The wave heights predicted in Table 6.2 seem reasonable, given that the waves photographed were in the region of head-high in comparison to the surfers. The orientations cover a range of 22° and the results of Mead and Black (2001) are close. Given the uncertainties in the values found here, this represents good agreement.

The single value presented by Mead and Black (2001) for the length to height ratio (l/H) is much smaller than those found here, which might suggest that the waves photographed here are smaller, however the width to height ratios (w/H) are lower here. So no conclusions can be

Figure 6.18: *One of the better examples of a barrel at Whangamata bar, the associated digitised points, and the cubic curve. The red points were used in fitting the cubic while the black points were not, since including them generated a poor curve. The cubic shown here is* unrectified *(image coordinates are generated from real-world coordinates) from the cubic fitted to the real-world coordinates.*

Wave	H (m)	w (m)	l (m)	θ (°)	l/w	l/H	w/H
1	1.98	0.58	1.74	46	2.98	0.88	0.29
2	1.50	0.42	1.71	44	4.05	1.14	0.28
3	1.40	0.58	1.47	55	2.54	1.05	0.41
4	1.81	0.47	1.58	48	3.36	0.87	0.26
5	1.91	0.52	1.99	43	3.81	1.04	0.27
6	1.63	0.54	1.90	45	3.52	1.16	0.33
7	1.63	0.53	1.51	51	2.85	0.92	0.32
8	1.76	0.62	2.32	36	3.74	1.31	0.35
9	1.78	0.30	0.89	58	2.94	0.50	0.17
Mean	1.71	0.51	1.68	47	3.31	0.99	0.30
Mead and	-	-	-	33	2.95	0.18	0.53
Black (2001)	-	-	-	43	2.90	-	-

Table 6.2: *Barrel dimensions for waves at Whangamata bar. See Figure 2.5 which shows definitions for the symbols. Results for Whangamata bar from Mead and Black (2001) are shown in the bottom two rows. Figure 6.18 shows wave four.*

drawn.

The length to width ratios here cover a large range, from 2.54 to 4.05, with the results of Mead and Black (2001) near the centre, representing surprisingly good agreement. Mead and Black (2001) described waves at Whangamata bar as being steep faced, but rarely tubing. Here the waves had definite barrels, but were not large, and occurred primarily at the point where the peak first broke. Most surfers would probably not describe these waves as barreling, since their interest lies more in the behaviour of the shoulder of the wave. Significantly, the photographs taken here all show waves very soon after the peak has plunged, but it is not clear at what time during the shoaling process the two photographs that Mead and Black (2001) used were taken.

6.2.6 Conclusions Regarding Breaking Waves at Whangamata Bar

Here waves breaking at Whangamata bar were examined experimentally and details of the barrels quantified. Breakers were photographed and rectified, then information on the barrels extracted. It was not possible to identify exactly important parameters for the camera used (namely the principal point) because of the lack of fiducial marks. This, with the limitations of the control points here, meant that the rectification process engendered significant uncertainties.

The details of the swell observed at Whangamata could only be estimated, because wave meters were not deployed, representing an obvious means by which this experiment could be improved. A camera that could record time would greatly increase the amount of information in the photographs. It was assumed that digitised points on the breaking waves also lay on the *typical orthogonal*, an imaginary construct generated from SWAN simulations. For each wave there exists a real peak orthogonal, but it will only be approximated by the typical orthogonal.

It was found that the waves observed here broke either over a section of the seabed which had a slope of approximately 1:77, or over a section with slope 1:22. Parameters that describe the barrels have been established, having orientations of between 36° and 58°, and length to width ratios of between 2.6 and 4.0.

To summarise, in this section, parameters describing how and where waves break at Whangamata bar were established. The reason for collecting is to provide a comparison for SPH simulations. In the following section such simulations will be presented, and comparisons made.

6.3 MARIAN Simulations of Waves Breaking at Whangamata Bar

The aim of this thesis is to investigate the applicability of a particle-based Lagrangian model for simulating processes in the coastal zone. In the preceding chapters, a model created to fit this description (MARIAN) was described, and details of its behaviour examined. In the coastal zone waves break over real bathymetries, and if MARIAN is to be considered a good tool, it must be demonstrated that it can simulate such processes. In the preceding section of this chapter breaking waves at a real beach were examined experimentally. In this section MARIAN is used to simulate these breaking waves.

It is found that MARIAN does not predict breaking waves; instead the waves continually lose height as they propagate. The behaviour of breaking waves observed at Whangamata bar, and indeed, seen in general, is not replicated by MARIAN. The reasons for this are investigated, and more simulations are conducted with the aim of identifying why this failure has occurred.

6.3.1 A First Attempt at Creating Breaking Waves

In this section attempts are made to simulate breaking waves at Whangamata bar using MARIAN. Details on how the initial state is created are given, as is information on the waves input to the

simulation. The results are analysed using similar techniques as have been used in the previous chapters.

Generating the Bathymetry

The bathymetry used in this simulation is created from depths along the typical orthogonal, and distance along the typical orthogonal. However these vectors are taken from SWAN output and have a relatively coarse resolution (around 10 m). The *interp1* function in MATLAB is used to interpolate values at the higher resolutions required by MARIAN. The bathymetry (shown by the green curve in Figure 6.10) has its deepest point at the seaward boundary, where it is 8.5 m deep.

Input Waves

A cnoidal wave with a period of 13.3 s, a 140 m wave length, and a height of 2.1 m is used as input. This wave is a permanent wave in water of 8.5 m, which is the depth at the seaward boundary. This wave is similar to those observed at Whangamata bar.

A wave train with four peaks is specified by the initial state method, with the first peak being half a wavelength from the seaward boundary. It is anticipated that the central two peaks will behave well, that the leading and trailing waves will be poorer approximations to breaking cnoidal waves.

Particles are initially placed on a rectangular grid with a regular spacing of 1.8 points per metre, meaning a total of 10 878 particles, each 308.6 kg, since the density is 1000 kg/m^3. The rectangular grid means curved surfaces cannot be resolved well early in the simulation. This is a problem at the free surface and also at sloping sections of the bathymetry. Here both the free surface, and the sea floor are sloping and curved. At the free surface the poor resolution affects how well the wave is simulated, while at the sea floor it causes the addition of spurious boundary potential and gravitational energies.

A step-size of 1 ms is used for 28 000 steps, with every 280 th step written to file. This is time enough for both the first and second peaks to pass the anticipated break point, but avoids the overly large output file if more steps are written to file, and the low temporal resolution caused by writing to file less frequently. If desired, new simulations can be hot-started to show the third and fourth peaks breaking. The smoothing length is 1.0 m (so particles will have in the region of 37 neighbours) and the speed of sound is 125 m/s (all flows should be slower than Mach 0.1). XSPH is used and the viscosity is chosen to be 0.0005. Ghost point boundaries are used in the hope that the corrugated boundaries potential will inhibit the growth of toothpaste jets.

Results

Figure 6.19 shows free surface profiles during the simulation, with depths also shown. The first three peaks experience at least part of the shoaling process. None of the peaks break as expected; at most they steepen, but not very much. The wave faces in Figure 6.19 are deceptively steep because of vertical exaggeration due to the differing scales on the vertical and horizontal axes.

For previous simulations, plots of velocities were examined for evidence of breaking, and similar plots are produced here in Figure 6.20 for the same reason. In none of these plots however, is clear indication of breaking present. The top plot shows something approximating expected behaviour (however it is difficult to see), with the beach-ward velocities at the peak and lower velocities in front of and behind the peak. However one the more obvious features is the amount of noise, which may be the result of spurious energy in the initial state. Various energies are plotted in Figure 6.21.

Once again, MARIAN has failed to simulate expected behaviour in breaking waves and there are a number of reasons why this failure may have occurred. Clearly the lack of toothpaste jets

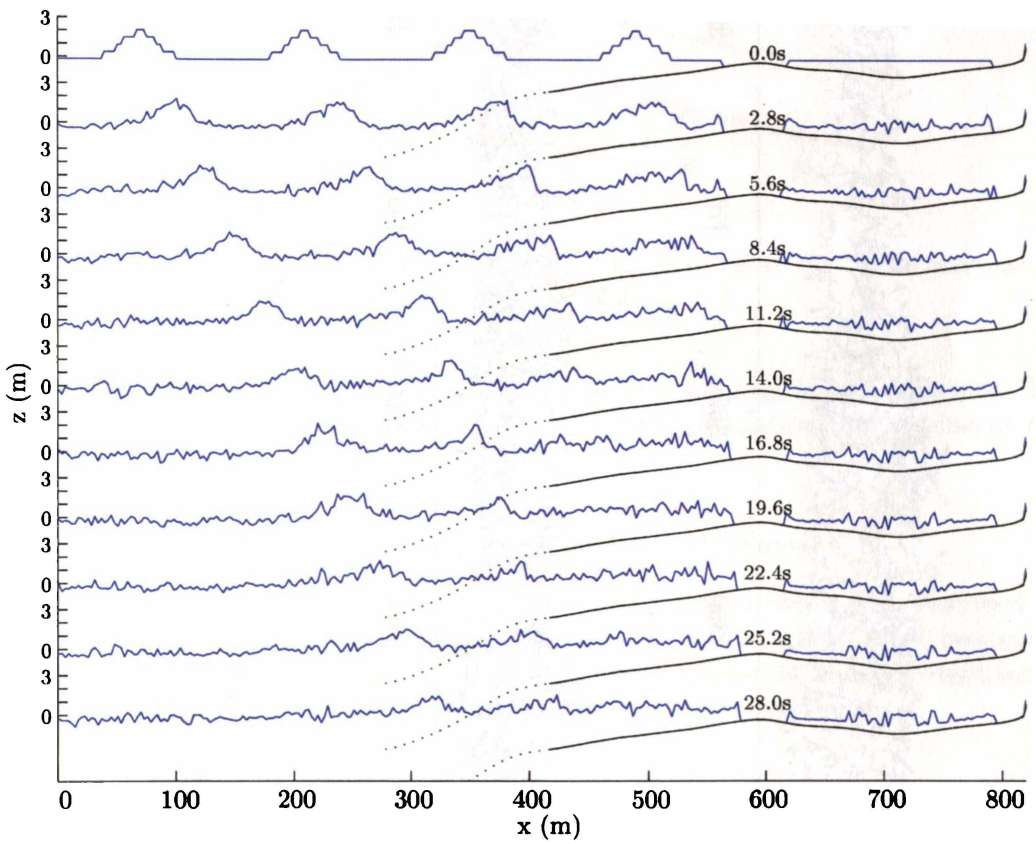

Figure 6.19: *Free surface profiles for the first attempt at simulating breaking waves at Whangamata bar. Note that the different scales used on the vertical and horizontal axes. The black curves (both dotted and full) are plots of the bathymetry shown at each time. The dotted section is used to avoid creating confusion in the following profiles.*

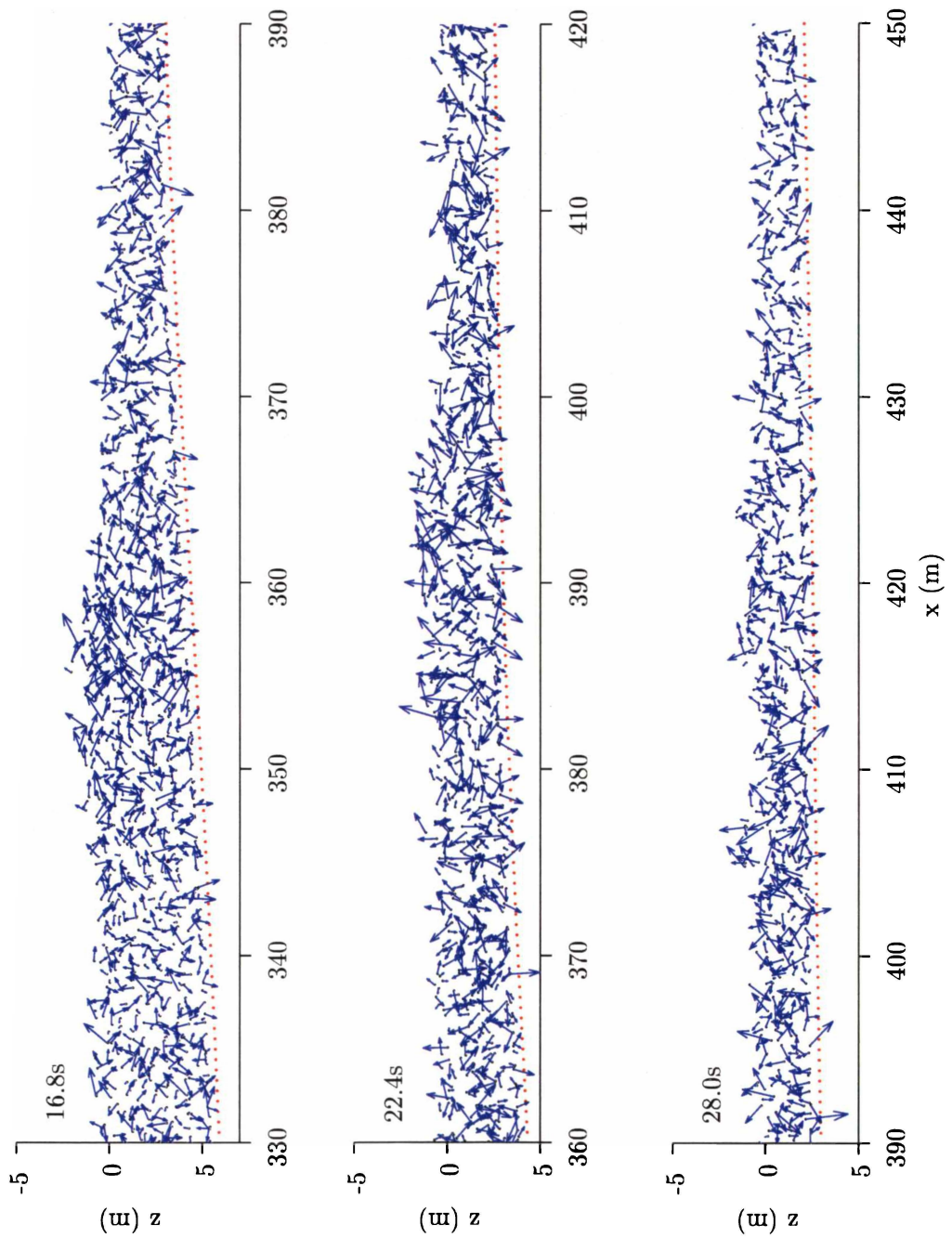

Figure 6.20: *Particle velocities for the Whangamata bar simulation for selected times. At 16.8 s the crest is at 357 m, at 22.4 s it is at 395 m, while at 28.0 s it is at 425 m.*

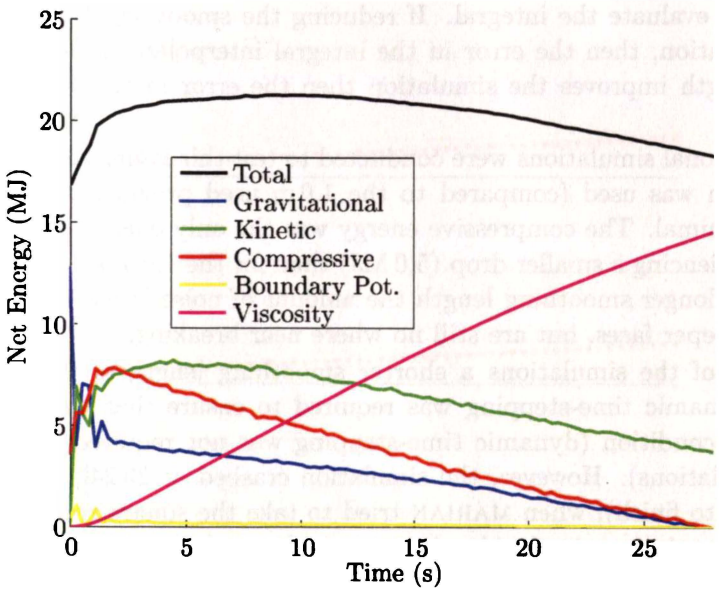

Figure 6.21: *Energies for the first Whangamata bar simulation. Note the initial increase in the compressive energy which is followed by a more gradual decay.*

(see Figure 6.20) here means that this explanation for non-breaking is not applicable. In the following section other reasons are identified, and additional simulations are conducted to test these suggestions.

6.3.2 Testing Hypotheses for the Non-Breaking of Waves

In the preceding sections an initial attempt at creating breaking waves was described. The attempt failed. Reasons for this failure are suggested in the following sections. The simulation discussed in the previous section is described as the ***standard simulation***, and provides a standard against which other simulations can be measured, since they are variations on it.

Time-Step

Shorter time-steps improve the accuracy of time-stepping methods, but at the cost of increased run-times if the simulation duration is to remain the same. Here the time-step was reduced to 0.5 ms, and the main effect is the change to the compressive energy. The change is not significant and the energies are not shown here. As in Figure 6.21, which shows the energies for the standard simulation, the compressive energy for the short time-step simulation still experiences an increase soon after the start of the simulation, but the loss in compressive energy occurs at a lower rate. However the noise observed in the free surface is of the same order in both simulations, as are the gradients of the wave faces. Errors due to time-stepping appear not to be the cause for the failure of waves to break, but may be a small contributor.

Smoothing Length

In deriving the SPH equations, two approximations are required: firstly approximating the value of a property by the integral interpolant, and secondly approximating the integral interpolant by a summation interpolant. The first approximation has an error that scales with h^2 (neglecting, for now, the lack of consistency). The second approximation has an error that reduces when more

points are used to evaluate the integral. If reducing the smoothing length causes an improvement to the simulation, then the error in the integral interpolant is dominating. If increasing the smoothing length improves the simulation then the error in the summation interpolant is dominant.

Here two additional simulations were conducted to test this issue. In the first case, a smoothing length of 1.2 m was used (compared to the 1.0 m used previously), but the effect on the simulation was minimal. The compressive energy was the only energy component that changed significantly, experiencing a smaller drop (5.0 MJ) than for the 1.0 m simulation (7.5 MJ, see Figure 6.21). For the longer smoothing length the amount of noise is marginally larger, and waves achieve slightly steeper faces, but are still no where near breaking.

In the second of the simulations a shorter smoothing length of 0.8 m was employed. In this simulation dynamic time-stepping was required to ensure that the time-step was smaller than the Courant condition (dynamic time-stepping was not required in either the $h = 1.0$ m or $h = 1.2$ m simulations). However, the simulation crashed at 23 240 time-steps (compared to 28 000 timestamps to finish), when MARIAN tried to take the square root of a negative number. This occurs when variations in the density are too large, beyond the limits within which the equation of state can be used. A shorter time-step would have avoided this problem. In the short time-step simulation, noise grows at a greater rate, and the change in compressive energy is a massive 60 MJ.

Evidently errors in the integral interpolant are not the cause of the failure, and while errors in the summation interpolant are possibly contributing, they are not the sole cause.

CSPH

A related issue is the lack of consistency. An additional simulation was run in which CSPH was turned on, the aim being to ensure that zeroth order consistency is achieved. However this simulation failed to complete even two time-steps, since it crashed. The crash was forced by MARIAN when dynamic time-stepping attempted to reduce the time-step to 1/16 its specified value, a preset limit.

This crash is clearly not due to the accumulation of errors, so it must be that for this initial state the CSPH equations are grossly unstable. This is an interesting occurrence, but it will not be investigated further.

Artificial Clustering

Artificial clustering causes problems in many of the simulations described in this work. Here a simulation is conducted in which clustering is avoided by using the modification to the cubic spline kernel given by Thomas and Couchman (1992), shown in Equation 3.45. In addition, a new kernel is used, a kernel which has a monotonically decreasing gradient. The effects of using the modified cubic spline and the new kernel on artificial clustering, and on the Whangamata bar simulation are examined.

The Modified Cubic Spline

Here the cubic spline kernel is modified according to the method of Thomas and Couchman (1992). The energies for this simulation are very similar to those shown in Figure 6.21. It is surprising that this change to the kernel does not have a greater effect on the compressive energy. The variation in densities are the same here as for the simulation with the unmodified kernel, with all particle densities being within 10 % of the standard density, for all time-steps, for each case.

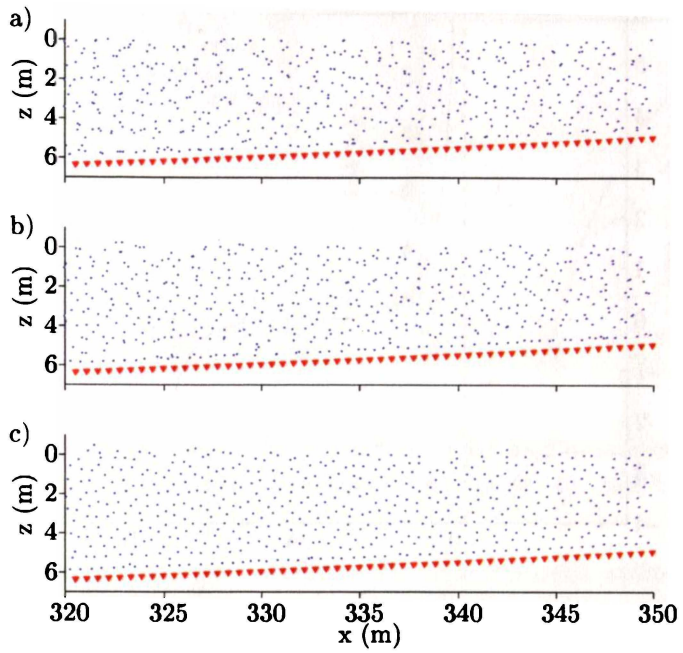

Figure 6.22: *Particle positions from three simulations showing artificial clustering.* (a) *The standard Whangamata bar simulation using the unmodified cubic spline;* (b) *A simulation using the modified cubic spline (Thomas and Couchman, 1992);* (c) *A simulation using a new kernel which has a monotonic gradient. All plots are for the same time-step. Artificial clustering is severe in* (a), *present in* (b) *and virtually non-existent in* (c).

The free surface profiles are similar to those in Figure 6.19, but with marginally steeper wave faces. Still, breaking is not observed. While the degree of artificial clustering is decreased, it can be seen in Figure 6.22 that clusters still occur.

A Kernel With a Monotonically Decreasing Gradient

In this section a new kernel which has a monotonically decreasing gradient is described.

Consider the function

$$y(x) = \frac{(A-x)^3}{x\,(B-x)} \tag{6.4}$$

which is plotted in Figure 6.23. The exponent 3 is present in Equation 6.4 to ensure that the second derivative is zero at $x = A$, as well as the first, important for stability in SPH (Morris *et al.*, 1997). By increasing the exponent to higher integers, it can be ensured that higher order derivatives are also zero at $x = A$. The kernel requires a transformation of variables (to the new variable r, for example), so that $x = \xi$ corresponds $r = 0$, and $x = A$ correspond to the radius of the support.

Following a transformation of variables the kernel becomes

$$H(r,h,n,\xi) = \frac{(n - r/h)^3}{\left(\frac{n\xi}{(n-\xi)} + r/h\right)\left(\frac{n(2n-\xi)}{(n-\xi)} - r/h\right)}, \tag{6.5}$$

which is not yet normalised. The support for this kernel has radius nh.

The normalisation constant is dependent on the desired dimension, and are calculated from: in one dimension

$$\frac{1}{C} = 2\int_{r=0}^{\infty} H(r,h,n,\xi)dr, \tag{6.6}$$

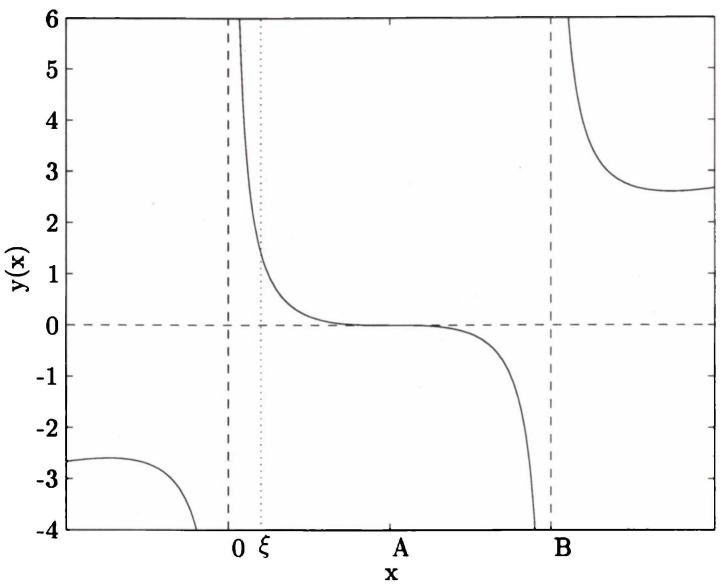

Figure 6.23: *A plot of the source function (Equation 6.4) for a kernel with a monotonically decreasing gradient. The section between ξ and A is used for the kernel, where ξ is an offset from the origin that ensures the kernel is integrable.*

from

$$\frac{1}{C} = 2\pi \int_{r=0}^{\infty} H(r,h,n,\xi) r dr \tag{6.7}$$

in two dimensions, and in three dimensions from:

$$\frac{1}{C} = 2\pi \int_{r=0}^{\infty} H(r,h,n,\xi) r^2 dr. \tag{6.8}$$

These integrals can be evaluated exactly and while the solutions are rather long and cumbersome, they simplify greatly when n and ξ are specified. Choosing $n = 2$ and $\xi = 1/2$ means the kernel has a gradient that matches that of the cubic spline well in the outer section. In one dimension the normalisation constant becomes

$$\frac{1}{C} = \frac{4}{9} h \left(-9 + 16\left(4\ln(2) - \ln(7)\right)\right) \tag{6.9}$$

while in two dimensions

$$\frac{1}{C} = \frac{8}{9}\pi h^2 \left(29 + \frac{16}{3}\left(12\ln(2) - 7\ln(7)\right)\right) \tag{6.10}$$

and in three dimensions

$$\frac{1}{C} = \frac{8}{9}\pi h^3 \left(93 + \frac{32}{9}\left(100\ln(2) - 49\ln(7)\right)\right). \tag{6.11}$$

The final form of the kernel is

$$H(r,h) = C \frac{(2 - r/h)^3}{\left(\frac{2}{3} + r/h\right)\left(\frac{14}{3} - r/h\right)} \tag{6.12}$$

and its gradient is given by

$$\nabla H(r,h) = \frac{-C}{h}(2 - r/h)^2 \times \frac{(2 - r/h)^2 + \frac{64}{9}}{\left(\frac{2}{3} + r/h\right)^2 \left(\frac{14}{3} - r/h\right)^2} \hat{\boldsymbol{r}} \tag{6.13}$$

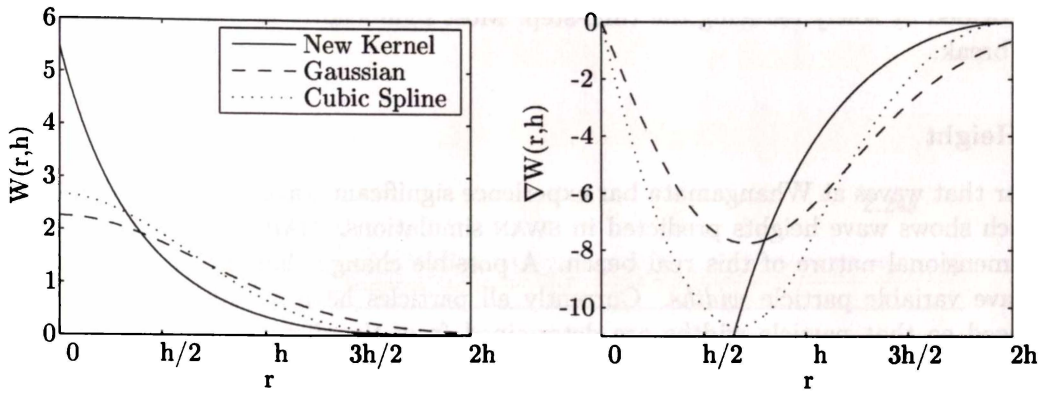

Figure 6.24: *The plot on the left shows the Gaussian, cubic and new kernels, and on the right are their gradients. The new kernel has a minimum gradient of -61, which occurs at zero.*

The new kernel is plotted with the Gaussian, and with the cubic spline, and their gradients (in one dimension) in Figure 6.24. The new kernel described here has a monotonic gradient and a compact support. It does not require a piecewise domain and is computationally cheap. The kernel is continuous everywhere within its domain and the 1st and 2nd derivatives are continuous everywhere in the domain except at the origin.

The Whangamata bar simulation performed with this kernel experienced much less clustering than simulations using either the cubic spline or the modified cubic spline. However, this simulation crashed at the 22 680 timestamps when MARIAN forced the simulation to end because the square of a velocity of sound was negative. Surprisingly, dynamic stepping was never used. It may be that this kernel requires a different Courant condition to that used for the cubic spline. It may also be that the energy previously tied up in clusters now remains free in the system, and a higher viscosity may be necessary to damp it out.

The compressive energy here experienced a marked drop, being 12 MJ, (in comparison to 7.5 MJ for the standard Whangamata bar simulation, see Figure 6.21). The free surfaces show the steepest wave faces observed yet, but these occur late in the simulation, and given the crash, it would be unwise to attribute too much importance to these. Also, the magnitude of noise in the free surface is marginally lower here, than in the standard simulation.

Returning to Artificial Clustering

Artificial clustering is reduced by the use of the modified cubic spline (Thomas and Couchman, 1992), and further reduced by the use of the new kernel developed to have a monotonically decreasing gradient. However while clustering has been reduced, breaking is still not observed, so the occurrence of artificial clustering does not provide the sole reason for the failure.

Speed of Sound

A marked difference in SPH in comparison to other modelling methods is the techniques with which compressibility is accounted for. It may be that the compressible nature of the fluid is causing problems. This hypothesis is tested by the creation of a simulation in which the velocity of sound is chosen to be 600 m/s (in comparison to 125 m/s for the standard simulation), requiring a reduction of the time-step from 1 ms to 0.4 ms in order to ensure that the Courant condition is still met.

This simulation took just over eleven days to complete, a consequence of the short time-step. The results were not markedly better than the standard simulation, with differences similar to

those provided by solely reducing the time-step. Most significantly in this simulation; the waves did not break.

Wave Height

It is clear that waves at Whangamata bar experience significant wave focusing, evident in Figure 6.7, which shows wave heights predicted in SWAN simulations. MARIAN cannot account for the three dimensional nature of this real beach. A possible change that might address this would be to have variable particle *widths*. Currently all particles have a unit width, but this could be changed so that particle widths are determined from the distance between adjacent wave orthogonals (see Figure 6.8 for an example of wave orthogonals).

An alternative, and much simpler approach is to use a bigger wave. Here a simulation is created in which the initial state contains a cnoidal wave with $\alpha = 0.6$. Of course the solution for the cnoidal wave that is used here is inaccurate for waves this high, but this is not of too much concern here. The plunging waves simulated by Koshizuka *et al.* (1998) are similar in that they have $\alpha = 0.5$.

Waves in this simulation did break, and plots of particle positions for the first wave in the train are shown in Figures 6.25 and 6.26. Similar plots for the second peak in the train are shown in Figures 6.27 and 6.28. The first peak breaks in very shallow water, very soon after the start of the simulation. Given that this peak is so high in comparison to the depth of water it is in, little importance can be attributed to it. It does offer some relief though, verifying that MARIAN can simulate a breaking wave, even if it is quite unphysical.

In Figures 6.27 and 6.28 it can be seen that the second peak in the train breaks part way up the 1:22 sloped section of the bathymetry, while waves observed during the field work at Whangamata are estimated to have broken over the 1:77 sloped section (or at the top of the 1:22 section). However, it must be recalled that the waves in this simulation are very high, being around 5 m high at the break point, so it is no surprise that they break in much deeper water.

The leftmost peak does not break, instead it loses significant height as it propagates, suggesting it is falling foul of the problems experienced in the numerical wave tank simulations of Chapter 5. Also problematic, is that the noise in the free surface grows to considerable magnitudes late in the simulation.

In Figure 6.29 plots of the particle velocities are presented. The velocities in these plots are very much as expected, at least to the eye. These plots serve mainly to verify earlier conclusions: that plots of velocities for previous simulations did not show behaviour consistent with having broken.

Barrels are clearly evident in Figures 6.27 and 6.28. For the second peak, points around the barrel are selected and the cubic that best fit them is found, shown in Figure 6.30. The process of selecting points around the barrel is very subjective. Notice the free surface behind the peak, where significant spaces between particles can be seen, indicating that the uncertainty in the choice of barrel points is high.

The parameters for the fitted cubic are presented in Table 6.3, together with some values from the field work. With the exception of the value for the orientation (which, for the simulation here, is low), the parameters in Table 6.3 are in good agreement. It must be remembered that there are high uncertainties in all of the data presented here, and significantly, that the simulated wave is very large in comparison to those photographed.

This represents success: waves can be simulated with MARIAN that break in a manner similar to what is expected. It seems that there is some limit as to the wave heights that can be simulated though, since smaller waves do not break as anticipated. It was noted earlier that waves at Whangamata bar experience significant refraction, which acts to increase wave heights, but does not affect period. The peaks of these waves are higher, but not thicker and they are no

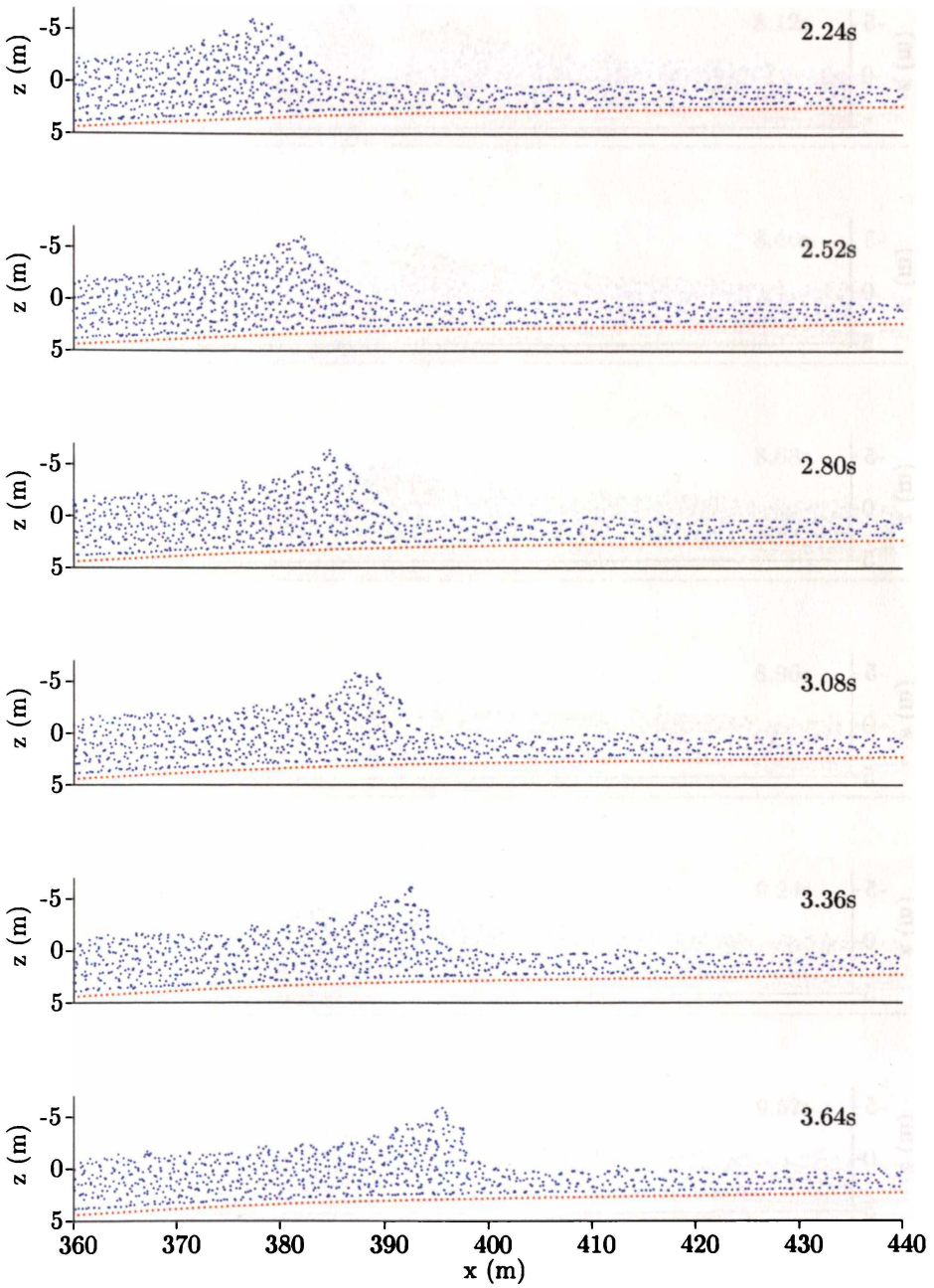

Figure 6.25: *Particle positions during the Whangamata bar simulation. The wave initially has* $\alpha = 0.6$ *and the peak shown breaking here is the second from the right in the wave train. See the following figure for later frames.*

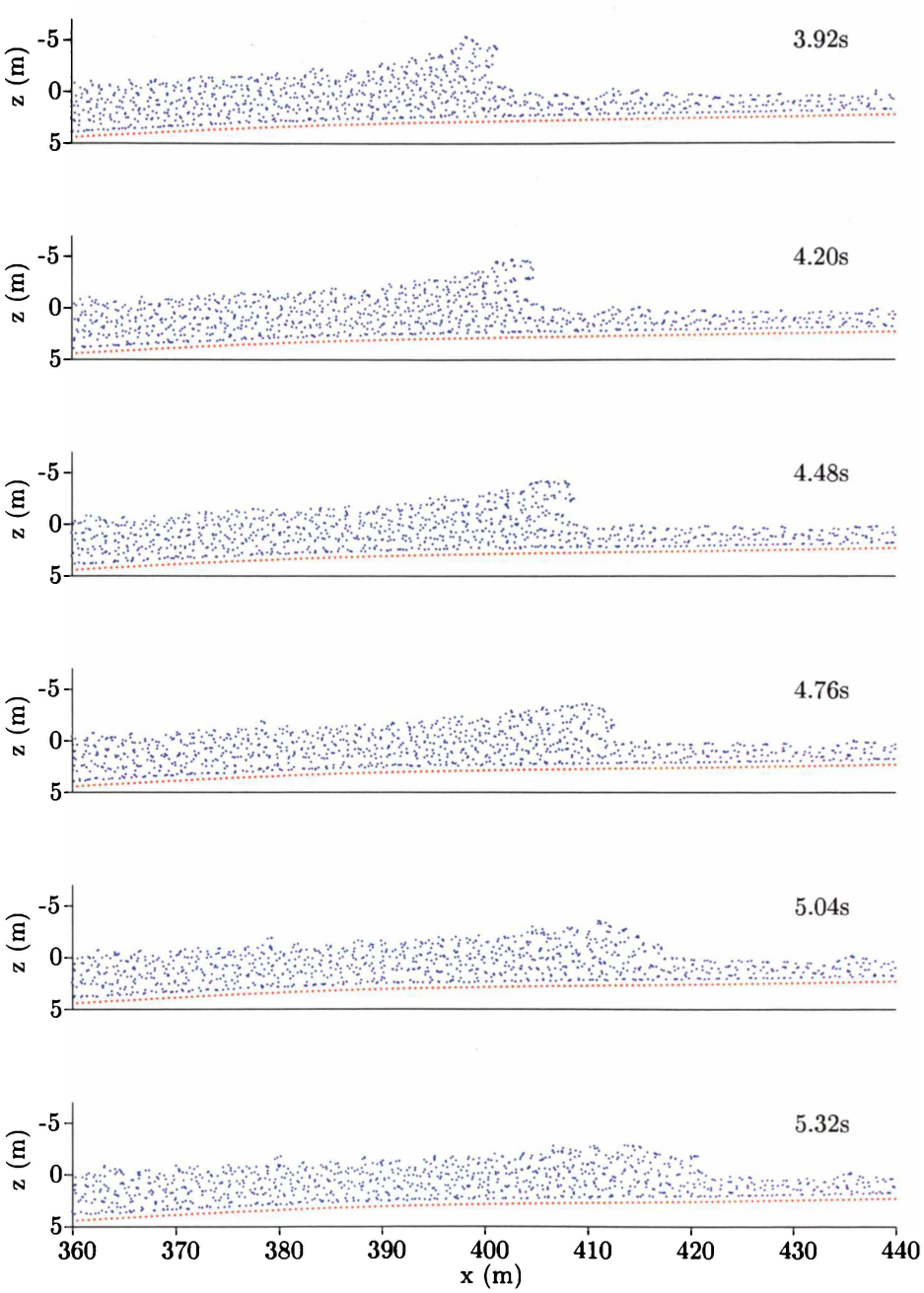

Figure 6.26: *Particle positions during the Whangamata bar simulation. The wave initially has* $\alpha = 0.6$ *and the peak shown breaking here is the second from the right in the wave train. See the previous figure for earlier frames.*

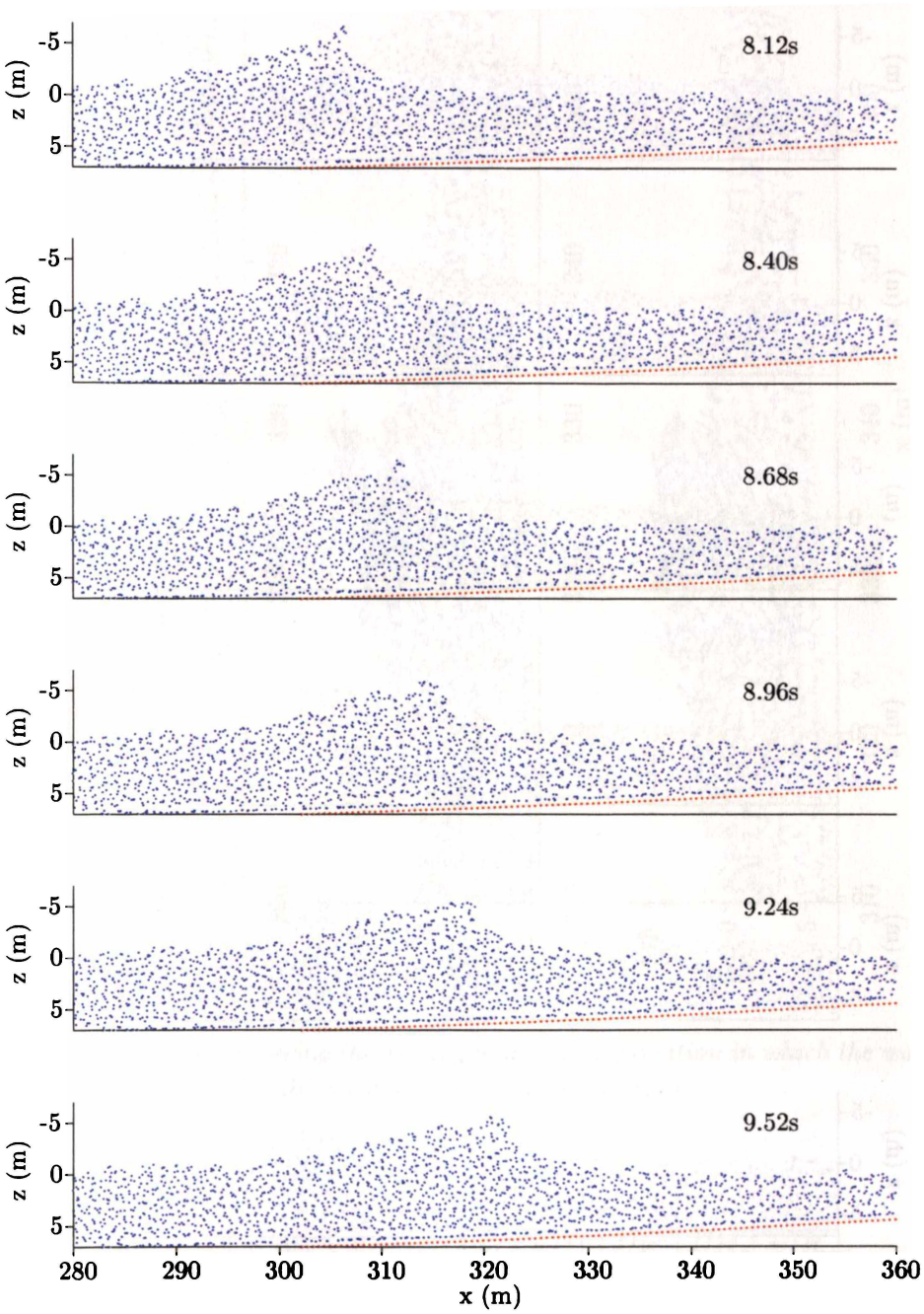

Figure 6.27: *Particle positions during the Whangamata bar simulation. The wave initially has* $\alpha = 0.6$ *and the peak shown breaking here is the third from the right in the wave train. See the following figure for later frames.*

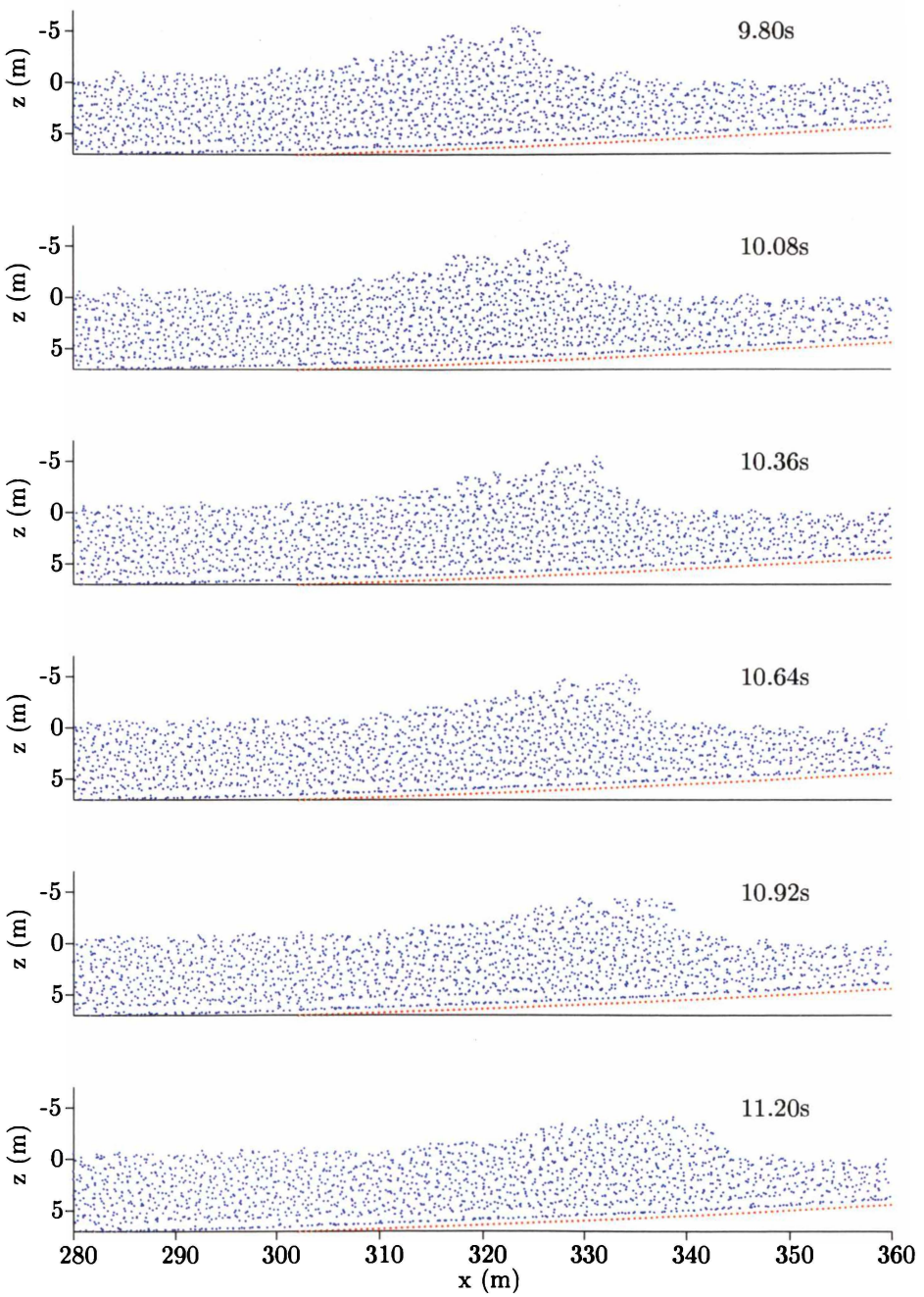

Figure 6.28: *Particle positions during the Whangamata bar simulation. The wave initially has* $\alpha = 0.6$ *and the peak shown breaking here is the third from the right in the wave train. See the previous figure for earlier frames.*

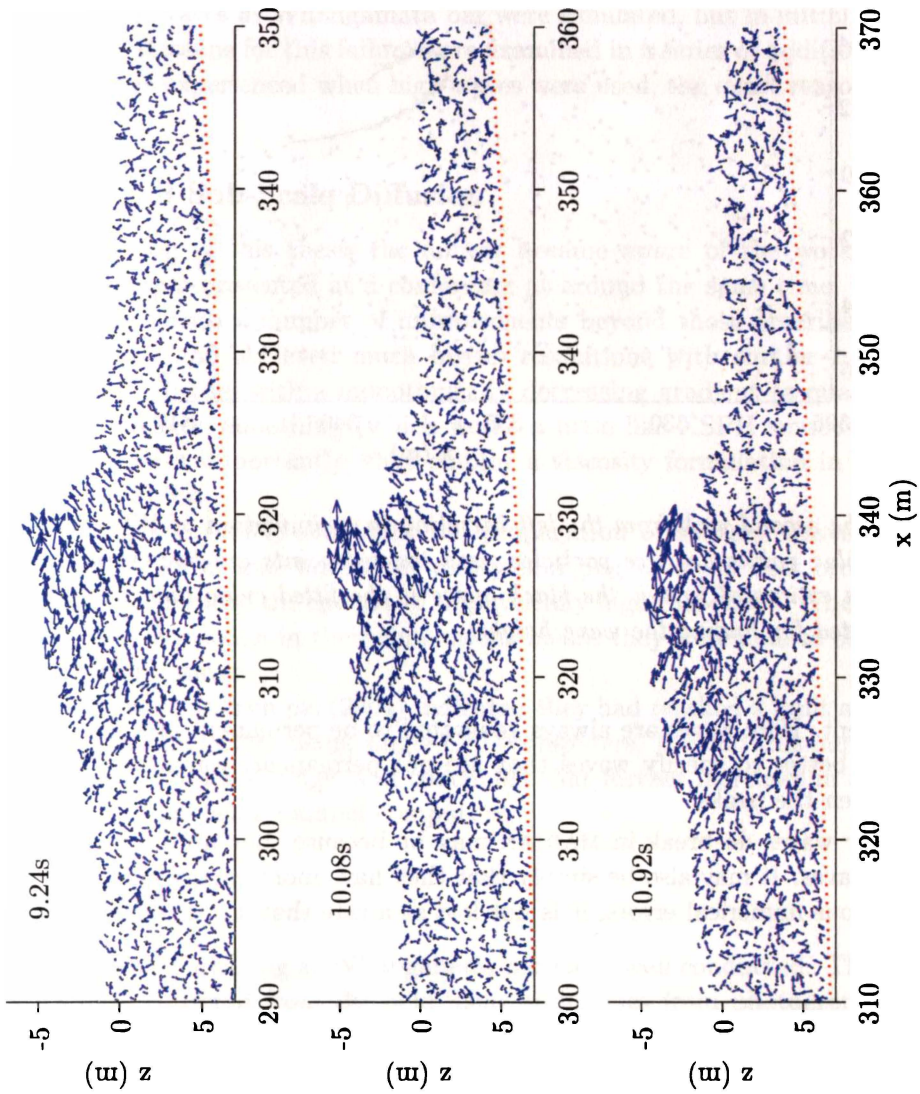

Figure 6.29: *Particle velocities during the Whangamata bar simulation in which the wave initially has* $\alpha = 0.6$. *These plots are for the second peak in the wave train.*

Wave	H (m)	w (m)	l (m)	θ (°)	l/w	l/H	w/H
Simulation	4.34	1.37	4.73	31	3.45	1.09	0.32
Maximum	1.98	0.62	2.32	58	4.05	1.31	0.53
Minimum	1.40	0.30	0.89	33	2.54	0.18	0.17
Mean	1.71	0.51	1.68	47	3.31	0.99	0.30

Table 6.3: *Barrel dimensions from the simulated wave at Whangamata bar. See Figure 2.5, which shows definitions for the symbols. The top line contains the data for the simulation here, and the following lines contain data compiled from Table 6.2.*

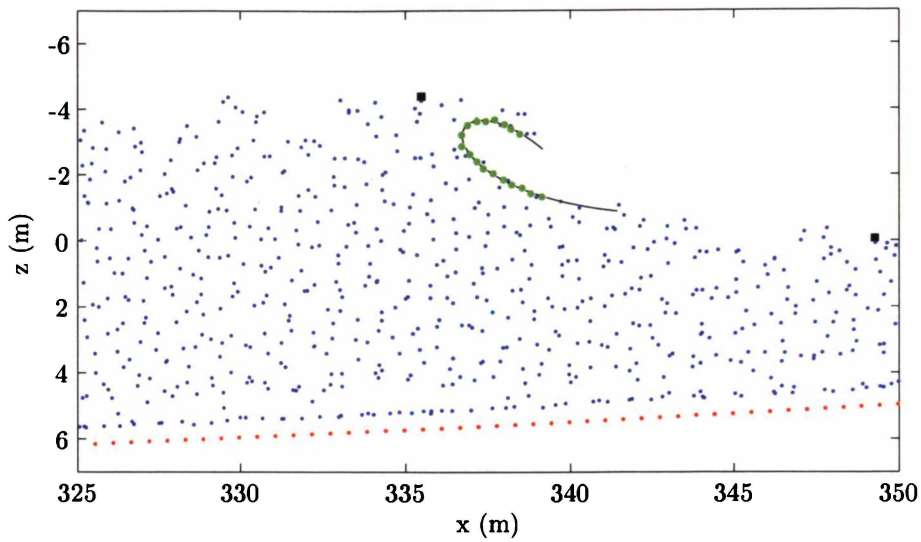

Figure 6.30: *The second peak from the left barreling in a simulation of Whangamata bar, where, as before, the blue points are free particles, and the red points are boundary points. The green points show the estimated barrel, the black curve is the fitted cubic and the black squares show the points selected for finding the wave height.*

longer permanent. Here waves are always generated to be permanent in a specified water depth, but it may be better to specify waves that are not permanent, waves in which refraction has acted to heighten the peaks.

That higher waves do break in MARIAN may be because they are better approximations to the refracted waves. It may also be simply that they have more energy and while some of this is lost to the various numerical errors, it is not lost at a rate that stops the wave from breaking.

Resolution

In all of the simulations discussed here, in the region of 10 000 particles are used. Compare this to the 200 000 grid cells used by Guignard *et al.* (2001) in their VOF model. So here the resolution is relatively low. It is anticipated that results will improve if higher resolutions are used, but this will not be tested here. While it is possible that there is a single reason (or perhaps a number of contributing factors) that explain why MARIAN does not always correctly simulate breaking, it is also possible that the cause is the low resolutions employed here, and that SPH needs high resolutions to achieve acceptable results.

The Missing Boundary Integrals

It was noted on Page 53, that some boundary integrals are neglected in the SPH equations used here (and in many other works, for example, Monaghan, 1994; Dalrymple and Knio, 2001), and that while this assumption is justified when dealing with unbounded simulations, it is not, in general, for fluid mechanics. The presence of the boundaries is significant here, and these missing integrals may be very important contributors to the failure of the simulations here. This topic is one that is deserving of attention: the author has seen only one reference to this topic (cited in Benz, 1990).

6.3.3 Concluding Remarks on the Whangamata Bar Simulations

In this section breaking waves at Whangamata bar were simulated, but in initial simulations the waves failed to break. Reasons for this failure were examined in a series of additional simulations. While some success was experienced when high waves were used, the exact reason for the failure was not established.

6.3.4 Viscosity and Sub-Scale Diffusion

Following the submission of this thesis the author became aware of the work of Rogers and Dalrymple (2004), a paper presented at a conference at around the same time. In this work an SPH model is discussed with a number of improvements beyond those described here. Firstly, they employ link-listing and achieved much higher resolutions with similar runtimes than in this work. They used a kernel with a monotonically decreasing gradient (a quadratic) to avoid clustering and used density smoothing (which works a little like XSPH smooths the velocity). Finally, and possibly most importantly, they employ a viscosity formulation in which sub-scale turbulence is accounted for.

Rogers and Dalrymple (2004) simulated the propagation of solitary waves which did not experience a change in form, and simulated weakly plunging breakers that broke as expected. So the changes they use remove the spectre of unphysically high dispersion. The high resolution and lack of unphysical dispersion in their simulations meant they were able to observe details of the flow within breaking waves.

In discussing this work Dalrymple (2004) indicated they had reached a limit as to the number of particles (96 000) in simulations using single-CPU computers. Beyond this number they found simulations required too much storage, and concluded that further increases in resolution could most easily be achieved by using parallel computers.

6.4 Conclusion

Here simulations of waves breaking at Whangamata bar have been conducted. These simulations are compared to data collected from the field using techniques from photogrammetry, namely rectifying data from photographs. The rectified data contained high uncertainties but results were consistent with data in the literature, and with observations.

The first simulation of a wave at Whangamata bar did not demonstrate breaking waves, and other simulations were run to establish why. Small improvements were achieved by changing various model parameters, but it was discovered higher waves did break as expected. This may be because the higher waves are better approximations to the refracted waves that occur on Whangamata bar, or because the higher energy is not lost at a rate great enough to stop breaking.

It is found that shorter timestamps and longer smoothing lengths cause slight improvements in simulations, but on their own they are not the cause for the failure to breaking. Two methods were employed for avoiding artificial clustering, and while they did achieve this, it appears that artificial clustering is not the cause for the failure to break. The relatively low resolutions used here may be the reason for the failure, but if so, high resolutions (and very long run-times) are required to get good results. Terms missing in the SPH equations are suggested as being a source for the failure, a suggestion that would require further research to verify.

Chapter 7

Conclusion: SPH for Simulating the Surf-Zone

The aim of this thesis is to establish whether Lagrangian particle-based methods are suitable for simulating processes in the coastal zone. This chapter contains the major findings of this thesis, from which a conclusion is drawn, responding to the aim. Conclusions based solely on the results of the work done here could not be anything but negative. However recent research has solved some of the problems encountered here and enables a more optimistic outlook. This recent research and the results of the work done here are the basis for conclusions drawn in this chapter. Also in this chapter, suggestions for further research stemming from this thesis are made.

Major Findings

For this research a model (called MARIAN) was created using SPH, a particle-based Lagrangian method. To establish that the model operated well a number of tests were performed, tests commonly used for such models. The evolution of an elliptic drop (Monaghan, 1994) involves finding the solution to a system of ordinary differential equations. The solution to this system of equations can be found accurately using standard numerical techniques, and also by using MARIAN. A comparison of results shows MARIAN behaves similarly to existing models described in the literature. Another common test is the bursting dam, a simulation for which good experimental data is available in the literature (Martin and Moyce, 1952). MARIAN also performed this test well, and is comparable to existing models.

But while initial tests of MARIAN demonstrate that it performs well in some simulations, other tests reveal that there are limits to what it can do well. In simulating a seiching basin the results were an approximation to expected behaviour, but the seiche attenuated too rapidly. Notably though, the period of the seiche in simulations was close to the period predicted by Merian's formula. These tests demonstrate undesirable traits in MARIAN: namely particle stacking, artificial clustering, poor initial system state, and growth of net energy. These tests also provided an opportunity to develop optimal operating procedure.

In order to simulate breaking waves, propagating waves must first be generated, and this process was tested in a number of simulations in which solitary and cnoidal waves were created in a numerical wave tank. Solitary waves appear to experience dispersion, losing height and increasing in width as they propagate, undesirable effects since these waves should have permanent form. The cnoidal waves, surprisingly, are much better and reasonable approximations to permanence are observed. For both solitary and cnoidal waves the wave celerities in simulations are slower than predicted by theory, the cause of which could only be suggested here. It

is possible that part of the cause lies in the model but is likely that the second order solution used for the comparison is poor also. In the numerical wave tank simulations noise at the free surface, artificial clustering, neglected boundary integrals, an inadequate viscosity formulation and less-than-perfect initial states are identified as contributing to poor performance.

Simulations of solitary waves approaching plane-sloped beaches showed very poor behaviour, in that the shoaling expected did not occur. It was not definitively established why this failure occurred, but significant contributors to the problem may include the formation of *toothpaste jets* (unphysical jets occurring in the fluid at the boundaries), artificial clustering and particle stacking. Cnoidal waves over a real bathymetry were also simulated, but only the very largest of these waves broke as expected; the smaller waves failed to break at all. Attempts to determine the reason for the failure established that either the accumulation of many errors caused the problem, or that resolutions used here are simply too low. Here, once again, the poor performance is caused by noise at the free surface, artificial clustering, neglected boundary integrals, an inadequate viscosity formulation and poor initial states.

Fieldwork was conducted to provide real data for comparison to MARIAN simulations. This involved obtaining profiles of real breaking waves using photogrammetric techniques. Parameters describing barrels and wave heights were found that are clearly a good approximation to the observed waves. However uncertainties in these values are high, and improvements to the experimental method necessary to reduce these were identified. It seems that laboratory experiments may be a better source of data for comparison at this stage.

So MARIAN is great for a few simulations, poor for others, and atrocious for some. For simulating breaking waves, only very big waves can be used, meaning there is a limit to what MARIAN is useful for. But simulations in the surf-zone are always going to be expensive to run, irrespective of the model employed, since they demand long, narrow model domains with high resolutions and long simulation durations. It is difficult to compare simulation run-times for different models since they are often conducted on completely different types of computers, but it seems that MARIAN requires relatively long run-times. Also adding to the cost of using MARIAN, is that a single good simulation is usually the culmination of a series of many abortive or only partially successful simulations. This is all about "bang for bucks", that is, whether MARIAN can provide more accurate simulations at lower cost than other models using other methods. The answer to this question is no, it cannot, at least not yet. It seems that with certain improvements MARIAN would become at least competitive though.

Potential theory models (more specifically those using the boundary element method, BEM) are by far the cheapest and fastest option for simulations up until the lip has hit the water (Svensden and Grilli, 1990). In the rationale for this thesis, the importance of being able to simulate breaker shape for the design of artificial surf reefs was noted, and for this task, a BEM model appears to be the best option (Svensden and Grilli, 1990). But modellers may desire simulations of waves after the lip has hit the water, and for this class of problem Volume of Fluid models are the present standard. Such models have been run at high resolutions on powerful computers and have provided good results, but do not compare in speed nor accuracy to BEM (Chen *et al.*, 1999; Guignard *et al.*, 2001). In comparison to MARIAN however, VOF models are faster and more accurate.

SPH models can be constructed from simple and concise source code, and do not suffer from numerical diffusion. These points, together with the fact that no special techniques are required for dealing with the free surface, are often quoted as merits of SPH (Benz, 1990; Monaghan and Kos, 1999). But these matter little if the model cannot simulate a process to even first order accuracy. The main reasons for the poor (and sometimes atrocious) performances of MARIAN are the low resolutions used, and the magnitude of the various errors.

The failure to achieve even zeroth order consistency, neglected boundary integrals, poor viscosity formulation, artificial clustering, unphysical boundary layers and the techniques used

to create initial states all contribute to poor performance. So if a tool is desired for simulating processes in the surf-zone, MARIAN is not the best option. But future improvements to algorithms and methods used in the model and to computer hardware mean this conclusion may be short-lived.

There is some flexibility in the choice for the governing equations for an SPH model, but all choices derive from the fundamental principles of physics. The exception to this is the use of the summation interpolant used as a replacement for the continuity equation. The justification for doing this (Gingold and Monaghan, 1977; Lucy, 1977) is incorrect, and doing so results in the non-conservation of mass (net mass is still conserved though).

The errors in SPH estimates have been examined before, but here they were written in the form $O\left(h^k/\sqrt{Nh^d}\right)$, where d is the number of dimensions and k is determined by the order of consistency (the ability to approximate a low-order polynomial exactly) achieved. In this form the error indicates clearly how the total number of points (N) must be scaled in order that a reasonable rate of convergence be achieved. Methods for achieving some order of consistency have been established but here it was established that they do so at the cost of ensuring the fundamental principles of physics are satisfied. The non-conservation of mass, energy and momentum is caused by kernel gradients (modified to ensure consistency) no longer having anti-symmetric gradients.

Following submission of this thesis, the author became aware of research in which some of these problems have been addressed. Rogers and Dalrymple (2004) developed an SPH model in which clustering and unphysical dispersion were not present, and used their model to simulate breaking waves with some success. So while MARIAN is incapable of performing these simulations satisfactorily, it can be done.

Future Research

At many times in the course of this research, limits were placed on what would be included, and often interesting research was left undone as a result. Here a number of directions for further research are suggested. Firstly, the success of Rogers and Dalrymple (2004) in simulating breaking waves with SPH represent improvements that are quite achievable for MARIAN.

The successes in simulating breaking waves that other modellers have experienced (in particular those using VOF and BEM models), have sometimes been achieved by using parallel CPU computers. Parallelisation of source code, enabling the program to run on multiple CPUs, is an obvious step forward in improving the performance of MARIAN. A super-computer would also make a difference. Some researchers have improved run-times for VOF models by coupling them to BEM models, an avenue that would reduce run-times for SPH models as well.

But the poor simulations seen here are only partly caused by the low resolutions resulting from limits on run-times: high errors are a significant problem in SPH. The implementation of SPH used in MARIAN has not accounted for the lack of even zeroth consistency, and this is a major source of error. This is high in priority on the list of improvements to MARIAN.

It is not entirely clear, but boundary integrals neglected when gradients of properties are found using SPH formulations may also be important sources of errors, a topic that is deserving of more attention. If boundary integrals are to be evaluated, then the free surface, which is also a boundary, needs to be identified more accurately. Algorithms for doing this exist in the literature, where they have been used to account for surface tension (Morris, 2000). Being able to identify the free surface as a sharp interface is also a priority for improving MARIAN, so including these boundary integrals is useful in two ways.

On the topic of boundaries, it should be noted that the methods for creating boundaries in SPH are less than perfect. Some methods generate unavoidable corrugations that cause problems

(ghost point boundaries, Monaghan, 1994), while for others the boundary potential energy is not defined (normal/tangential boundaries, Monaghan and Kos, 1999). Better, more controllable methods for generating boundaries are desirable. Also significant in some simulations performed here, are boundary jets, strange layers of particles along boundaries that move in unphysical ways and contribute to poor performance. A better understanding of what causes these jets would be an important step forward.

Particle stacking and crystallisation (Lombardi *et al.*, 1999) are two similar features of SPH simulations that are not well understood. It is most likely they are symptoms of the same discretisation error. This topic requires more research to establish the cause for the effects, methods for identifying when it is occurring in a simulation, and how to avoid it.

A conclusion that occurs frequently in the discussion of simulations run for this thesis, is that poor initial states may contribute significantly to poor results. Improvements to this aspect of SPH modeling would be of some value, hopefully reducing the addition of spurious energy to systems, and reducing the magnitude of noise.

In the final chapter, in an attempt to avoid the consequences of artificial clustering, a new kernel was proposed that has a monotonically decreasing gradient. This kernel shows some promise, since results demonstrate little clustering, but questions remain as to the stability of SPH when it is used. This kernel merits more examination in order to establish how it best be employed. Rogers and Dalrymple (2004) used a kernel with a monotonically decreasing gradient (they used a quadratic) with some success. The quality of their simulations is partly reflected by the lack of clustering.

It seems that there are some questions as to how the governing equations should be derived in order that estimates achieve some order of consistency but still enforce the conservation laws. The equations that are derived for standard SPH are lacking terms required by Corrected SPH. Using corrected estimates with uncorrected equations means errors are introduced. More research is required on this topic, in particular it needs to be established that these errors converge to zero at a suitable rate. If this proves to be impossible better governing equations need to be derived for use with Corrected SPH. Sacrificing the conservation principles for supposed improvements in accuracy results in a model that does not simulate the real world, which is surely the goal of any model.

The aim of this thesis was to examine Lagrangian particle-based methods, and SPH is just one example from the family of meshfree methods which could have been used. It seems that SPH always converges and can do so at a good rate, but this may require long runtimes to achieve. As such, SPH represents an entry level option, offering low accuracy but fast simulations. If higher accuracy is desired it may necessitate an alternative meshfree method.

References

Aagard, T. and G. Masselink (1999). The surf zone. In: A. D. Short (editor), *Handbook of Beach and Shoreface Morphodynamics*, pp. 72–118. John Wiley & Sons, Chichester.

Adams, R. A. (1999). *Calculus of Several Variables.* Addison-Wesley, Ontario, fourth Edition, 625 p.

Anderson, J. J. J. (1995). *Computational Fluid Dynamics. The Basics With Applications.* McGraw-Hill, inc., 545 p.

Ashgriz, N. and J. Y. Poo (1991). FLAIR: Flux and line-segment model for advection and interface reconstruction. *Journal of Computational Physics*, **93**, 449–468.

Babuška, I. and B. Q. Guo (1996). Approximation properties of the *h-p* version of the finite element method. *Computer Methods in Applied Mechanics and Engineering*, **133**, 319–346.

Baker, J. G., A. A. Magill, W. P. Tayman, K. Pestrecov, and F. E. Washer (1980). *Manual of Photogrammetry*, Chapter 3: Elements of Photogrammetric Optics, pp. 103–185. American Society of Photogrammetry, Fourth Edition.

Batchelor, G. K. (1967). *An Introduction to Fluid Dynamics.* Cambridge University Press, Cambridge, UK, 615 p.

Battjes, J. A. (1974). Surf similarity. *Proceedings of the 14th International Conference on Coastal Engineering*, Coppenhagen, Denmark, pp. 466–480. ASCE.

Belytschko, T., Y. Krongauz, D. Organ, M. Fleming, and P. Krysl (1996). Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering, Special Issue on Meshless Methods*, **139**, 3–47.

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

Benz, W. (1990). Smooth particle hydrodynamics: A review. In: J. R. Buchler (editor), *The Numerical Modelling of Nonlinear Stellar Pulsations*, pp. 269–288. Kluwer Academic Publishers.

Black, K. P. and C. J. Andrews (2001). Sandy shoreline response to offshore obstacles, Part 1: Salient and tombolo geometry. *Journal of Coastal Research*, **SI29**, 82–93.

Bonet, J. and S. Kulasegaram (2000). Correction and stabilisation of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, **47**, 1189–1214.

Bonet, J. and S. Kulasegaram (2001). Remarks on tension instability of Eulerian and Lagrangian corrected smooth particle hydrodynamics (CSPH) methods. *International Journal for Numerical Methods in Engineering*, **52**, 1203–1220.

Bonet, J., S. Kulasegaram, M. X. Rodriguez-Paz, and M. Profit (2004). Variational formulation for the smooth particle hydrodynamics (SPH) simulation of fluid and solid problems. *Computer Methods in Applied Mechanics and Engineering*, **193**, 1245–1256.

Bonet, J. and T. S. L. Lok (1999). Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, **180**, 97–115.

Booij, N., L. H. Holthuijsen, and R. C. Ris (1996). The SWAN model for shallow water. *Proceedings of the 25th International Conference on Coastal Engineering*, Orlando, Florida, Volume 1, pp. 668–676. ASCE.

Bradford, S. F. (2000). Numerical simulation of surf zone dynamics. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **126**, 1–13.

Carrington, C. G. (1994). *Basic Thermodynamics*. Oxford University Press, New York, 385 p.

Chang, K.-A. and P. L.-F. Liu (1997). Velocity, acceleration and vorticity under a breaking wave. *Physics of Fluids*, **10(1)**, 327–329.

Chen, G., C. Kharif, S. Zaleski, and J. Li (1999). Two-dimensional Navier-Stokes simulation of breaking waves. *Physics of Fluids*, **11**, 121–133.

Churchill, R. V. and J. W. Brown (1990). *Complex Variables and Applications*. McGraw-Hill Book Company, Singapore, Fifth Edition, 361 p.

Cleary, P. W. and J. J. Monaghan (1993). Boundary interactions and transition to turbulence for standard CFD problems using SPH. In: D. E. Stewart, D. Singelton, and H. Gardner (editor), *Proceedings of the 6th International Computational Techniques and Applications Conference: CTAC93*, pp. 157–165. World Scientific, Singapore.

Cummins, S. J. and M. Rudman (1999). An SPH projection method. *Journal of Computational Physics*, **152**, 584–607.

Dalrymple, R. A. Personal communication (2004).

Dalrymple, R. A. and O. M. Knio (2001). SPH modelling of water waves. *Proceedings of Coastal Dynamics 2001*, Lund, Sweden, pp. 779–787. ASCE.

Dalrymple, R. A., O. M. Knio, D. T. Cox, M. Gesteira, and S. Zou (2001). Using a Lagrangian particle method for deck overtopping. *Proceedings of the Fourth International Symposium on Ocean Wave Measurement and Analysis*, San Francisco, U.S.A., pp. 1082–1091. ASCE.

Dean, R. G. and R. A. Dalrymple (1991). *Water Wave Mechanics for Engineers and Scientists*. Advanced Series on Ocean Engineering - Volume 2. World Scientific Co. Pte. Ltd., Singapore, 353 p.

Douglass, S. L. (1990). Influence of wind on breaking waves. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **116(6)**, 651–663.

Duarte, C. A. (1995). A review of some meshless methods to solve partial differential equations. Technical Report 95-06, TICAM, The University of Texas at Austin. 37 p.

Duarte, C. A. and J. T. Oden (1995). Hp clouds - a meshless method to solve boundary value problems. Technical Report 95-05, TICAM, The University of Texas at Austin. 66 p.

Fenton, J. (1972). A ninth-order solution for the solitary wave. *Journal of Fluid Mechanics*, **53**, 257–271.

Ferziger, J. H. and M. Perić (1996). *Computational Methods for Fluid Dynamics*. Springer, Berlin, 364 p.

Flow Science Inc. `www.flow3d.com` (2003).

Fries, T.-P. and H. G. Matthies (2003). Classification and overview of meshfree methods. Technical Report Informatikbericht Nr.: 2003-03, Institute of Scientific Computing, Technical University of Braunschweig. 85 p.

Galvin, C. J. (1968). Breaker type classification on three laboratory beaches. *Journal of Geophysical Research*, **73(12)**, 3651–3659.

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

Gingold, R. A. and J. J. Monaghan (1982). Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, **46**, 429–453.

Golden Software Inc. `www.goldensoftware.com` (2004).

Goring, D. G. (1978). Tsunamis–the propagation of long waves onto a shelf. Technical Report KH-R-38, California Institute of Technology. 337 p.

Goring, D. G. and F. Raichlen (1980). The generation of long waves in the laboratory. *Proceedings of the 17th International Conference on Coastal Engineering*, Sydney, Australia, pp. 763–783. ASCE.

Gotoh, H. and T. Sakai (1999). Lagrangian simulation of breaking waves using particle method. *Coastal Engineering Journal*, **41**, 303–326.

Grilli, S. T., P. Guyenne, and F. Dias (2001). A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom. *International Journal for Numerical Methods in Fluids*, **35**, 829–867.

Grilli, S. T. and J. Horrillo (1997). Numerical generation and absorption of fully nonlinear periodic waves. *Journal of Engineering Mechanics*, **123(10)**, 1060–1069.

Grilli, S. T., I. Skourup, and I. A. Svendsen (1989). An efficient boundary element method for nonlinear water waves. *Engineering Analysis with Boundary Elements*, **6(2)**, 97–107.

Grilli, S. T., R. Subramanya, I. A. Svensden, and J. Veeramony (1997a). Shoaling of solitary waves on plane beaches. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **120(6)**, 609–628.

Grilli, S. T. and I. A. Svendsen (1990). Long wave interaction with steeply sloping structures. *Proceedings of the 22nd International Conference on Coastal Engineering*, Delft, The Netherlands, pp. 1200–1213. ASCE.

Grilli, S. T., I. A. Svensden, and R. Subramanya (1997b). Breaking criterion and characteristics for solitary waves on slopes. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **123(3)**, 102–112.

Grimshaw, R. (1971). The solitary wave in water of variable depth. Part 2. *Journal of Fluid Mechanics*, **46**, 611–622.

Guignard, S., S. T. Grilli, R. Marcer, and V. Rey (1999). Computation of shoaling and breaking waves in nearshore areas by the coupling of BEM and VOF methods. *Proceedings of The Ninth (1999) International Offshore and Polar Engineering Conferen ce*, Brest, France, Volume 3, pp. 304–309. ISOPE.

Guignard, S., R. Marcer, V. Rey, C. Kharif, and P. Fraunie (2001). Solitary wave breaking on sloping beaches: 2d two phase flow numerical simulation by SL-VOF method. *European Journal of Mechanics B/Fluids*, **20**, 57–74.

Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. In: B. Alder, S. FernBach, and M. RotenBerg (editor), *Methods in Computational Physics*, pp. 319–343. Academic Press, New York.

Harlow, F. H. and B. D. Meixner (1961). The particle-and-force computing method for fluid dynamics. Technical Report LAMS-2567, Los Alamos Scientific Laboratory. 49 p.

Harlow, F. H. and J. E. Welch (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, **8(12)**, 2182–2189.

Hernquist, L. and N. Katz (1989). TREESPH: A unification of SPH with the hierarchial tree method. *Astrophysical Journal Supplement Series*, **70(2)**, 419–446.

Hirt, C. W. (1970). An arbitrary Lagrangian-Eulerian computing technique. *Proceedings of The Second International Conference on Numerical Methods in Fluid Dynamics*, Berkley, California, pp. 350–355. Springer-Verlag.

Hirt, C. W., J. L. Cook, and T. D. Butler (1970). A Lagrangian method for calculating the dynamics of an incompressible fluid with a free surface. *Journal of Computational Physics*, **5**, 103–124.

Hirt, C. W. and B. D. Nichols (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**, 201–225.

Isobe, M. (1988). Beach morphologies and their change. In: K. Horikawa (editor), *Nearshore Dynamics and Coastal processes*, pp. 136–166. University of Tokyo Press, Tokyo.

Katell, G. and B. Eric (2002). Accuracy of solitary wave generation by a piston wave maker. *Journal of Hydraulic Research*, **40(3)**, 321–331.

Kinsman, B. (1984). *Wind Waves. Their Generation and Propagation on the Ocean Surface.* Dover Publications, Inc., New York, 676 p.

Komar, P. D. (1998). *Beach Processes and Sedimentation.* Prentice-Hall, Inc., New Jersey, Second Edition, 544 p.

Koshizuka, S., A. Nobe, and Y. Oka (1998). Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, **26**, 751–769.

Koshizuka, S. and Y. Oka (1995). Moving Particle Semi-implicit method: a gridless approach based on particle interactions for incompressible flow simulation. *Proceedings of the Third Workshop on Supersimulators for Nuclear Power Plants*, Tokyo, pp. 43–49.

Koshizuka, S. and Y. Oka (1996). Moving Particle Semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, **123**, 421–434.

Koshizuka, S. and Y. Oka (2000). Moving Particle Semi-implicit method: fully Lagrangian analysis of incompressible flows. *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering 2000*, Barcelona, 16 p. ECCOMAS.

Koshizuka, S., H. Tamako, and Y. Oka (1995). A particle method for incompressible viscous flow with fluid fragmentation. *Computational Fluid Dynamics Journal*, **4**, 29–46.

Lambert, J. D. (1973). *Computational Methods in Ordinary Differential Equations.* John Wiley and Sons, Chichester, 278 p.

Lattanzio, J. C., J. J. Monaghan, H. Pongracic, and M. P. Schwarz (1986). Controlling penetration. *SIAM Journal on Scientific and Statistical Computing*, **7(2)**, 591–598.

Li, Y. and F. Raichlen (1998). Discussion: Breaking criterion and characteristics for solitary waves on slopes. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **124**, 329–334.

Lin, P. and P. L.-F. Liu (1998a). A numerical study of breaking waves in the surf zone. *Journal of Fluid Mechanics*, **359**, 239–264.

Lin, P. and P. L.-F. Liu (1998b). Turbulence transport, vorticity dynamics, and solute mixing under plunging breaking waves in surf zone. *Journal of Geophysical Research*, **103(c8)**, 15677–15694.

Liszka, T., C. A. Duarte, and W. W. Tworzydlo (1996). Hp-meshless cloud method. Technical report, The Computational Mechanics Company. 41 p.

Liszka, T. and J. Orkisz (1977). Finite difference method at arbitrary irregular meshes in non-linear problems of applied mechanics. *Proceedings of the 4th International Conference on Structural Mechanics in Reactor Technology*, San Francisco, pp. 1–10.

Liu, W. K., S. Jun, and Y. F. Zhang (1995). Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, **20**, 1081–1106.

Lombardi, J. C., A. Sills, F. A. Rasio, and S. L. Shaprio (1999). Tests of spurious transport in smoothed particle hydrodynamics. *Journal of Computational Physics*, **152**, 687–735.

Longuet-Higgins, M. S. (1982). Parametric solutions for breaking waves. *Journal of Fluid Mechanics*, **121**, 403–424.

Longuet-Higgins, M. S. and E. D. Cokelet (1976). The deformation of steep surface waves on water. Part I: A numerical method of computation. *Proceedings of the Royal Society of London, Series A*, **350**, 1–26.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, **82**, 1013–1024.

Macky, G. H. and G. J. Latimer (1995). Wave climate in the Western Bay of Plenty, New Zealand, 1991-1993. *New Zealand Journal of Marine and Freshwater Research*, **29**, 311–327.

Martin, J. C. and W. J. Moyce (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*, **244**, 312–324.

McComb, W. D. (1990). *The physics of fluid turbulence.* Clarendon Press, Oxford, 572 p.

Mead, S. and K. P. Black (2001). Predicting the breaking intensity of surfing waves. *Journal of Coastal Research*, **SI29**, 51–65.

Mead, S. T. *Incorporating High-Quality Surfing Breaks into Multi-Purpose Reefs.* Phd, The University of Waikato (2000).

Mei, C. C. (1989). *The Applied Dynamics of Ocean Surface Waves.* World Scientific, Singapore, 740 p.

Monaghan, J. J. (1982). Why particle methods work. *SIAM Journal on Scientific and Statistical Computing*, **3**, 422–433.

Monaghan, J. J. (1988). An introduction to SPH. *Computer Physics Communications*, **48**, 89–96.

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. *Annual Review of Astronomy and Astrophysics*, **30**, 543–574.

Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, **110**, 399–406.

Monaghan, J. J. Personal communication (2003).

Monaghan, J. J., R. A. F. Cas, A. M. Kos, and M. Hallworth (1999). Gravity currents descending a ramp in a stratified tank. *Journal of Fluid Mechanics*, **379**, 39–70.

Monaghan, J. J. and A. Kos (1999). Solitary waves on a Cretan beach. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **125(3)**, 145–154.

Monaghan, J. J. and J. C. Lattanzio (1985). A refined particle method for astrophysical problems. *Astronomy and Astrophysics*, **149**, 135–143.

Morris, J. P. A. (2000). Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, **33**, 333–53.

Morris, J. P. A., P. J. Fox, and Y. Zhu (1997). Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, **136**, 214–226.

New, A. L. (1983). A class of elliptic free-surface flows. *Journal of Fluid Mechanics*, **130**, 219–239.

Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods.* Society for Industrial and Applied Mathematics, Montpelier, Vermont, 241 p.

Oran, E. S. and J. P. Boris (1987). *Numerical Simulation of Reactive Flow.* Elsevier, New York, 601 p.

Rabczuk, T. and T. B. S. P. Xiao (2004). Stable particle methods based on Lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering*, **193**, 1035–1063.

Radovitsky, R. and M. Ortiz (1998). Lagrangian finite element analysis of Newtonian fluid flow. *International Journal for Numerical Methods in Fluids*, **43**, 607–619.

Ranasinghe, R., N. Hacking, and P. Evans (2001). Multi-functional artificial surf breaks: A review. Technical Report CNR 2001.015, NSW Department of Land and Water Conservation, NSW Government. 53 p.

Reif, F. (1985). *Statistical and Thermal Physics.* McGraw-Hill Book Company, Singapore, international Edition, 651 p.

Rogers, B. D. and R. A. Dalrymple (2004). Sph modelling of breaking waves. *Proceedings of the 29th International Conference on Coastal Engineering, ICCE 2004*, Lisbon, portugal.

Sakai, T., T. Mizutani, H. Tanaka, and Y. Tada (1986). Vortex formation in plunging breaker. *Proceedings of the 21st International Conference on Coastal Engineering*, Taiwan, pp. 711–723. ASCE.

Sawaragi, T. (1995). *Coastal Engineering - Waves, Beaches, Wave-Structure Interactions.* Elsevier Science B.V., Amsterdam, The Netherlands, 479 p.

Schlichting, H. (1955). *Boundary-Layer Theory.* McGraw-Hill, New York, 747 p.

Schussler, M. and D. Schmitt (1981). Comments on smoothed particle hydrodynamics. *Astronomy and Astrophysics*, pp. 373–379.

Shuto, N. (1988). Part I. Chapter 1. Introduction. In: K. Horikawa (editor), *Nearshore Dynamics and Coastal Processes*, pp. 11–15. University of Tokyo Press, Japan.

Springel, V., N. Yoshida, and S. D. White (2001). GADGET: A code for collisionless and gasdynamical cosmological simulations. *New Astronomy*, **6**, 79–117.

Steinmetz, M. (1996). GRAPESPH: Cosmological smoothed particle hydrodynamics simulations with the special-purpose hardware GRAPE. *Monthly Notices of the Royal Astronomical Society*, **278**, 1005–1017.

Svensden, I. A. and S. T. Grilli (1990). Nonlinear waves on steep slopes. *Journal of Coastal Research*, **SI7**, 185–202. Special issue on rational design of mound structures.

Svensden, I. A. and U. Putrevu (1995). Surf-zone hydrodynamics. In: P. L.-F. Li (editor), *Advances in Coastal and Ocean Engineering*, Volume 2, pp. 1–78. World Scientific Publishing Co. Pte. Ltd., Singapore.

Swegle, J. W., D. L. Hicks, and S. W. Attaway (1995). Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, **116**, 123–134.

Synolakis, C. E. (1987). The runup of solitary waves. *Journal of Fluid Mechanics*, **185**, 523–545.

Synolakis, C. E. and J. E. Skjelbreia (1993). Evolution of maximum amplitude of solitary waves on plane beaches. *Journal of Waterway, Port, Coastal and Ocean Engineering*, **119(3)**, 323–342.

Takikawa, K., F. Yamada, and K. Matsumoto (1997). Internal characteristics and numerical analysis of plunging breaker on a slope. *Coastal Engineering*, **31**, 143–161.

Thomas, P. A. and H. M. P. Couchman (1992). Simulating the formation of a cluster of galaxies. *Monthly Notices of the Royal Astronomical Society*, **257**, 11–31.

Tolman, H. L. (2002). User manual and system documentation of WAVEWATCH-III version 2.22. Technical Report 222, NOAA/NWS/NCEP/MMAB. 133 p.

Tworzydlo, W. W. (1987). Analysis of large deformations of membrane shells by the generalized finite difference method. *Computers and Structures*, **27(1)**, 39–59.

Tworzydlo, W. W. (1989). The FDM in arbitrary curvilinear co-ordinates–formulation, numerical approach and applications. *International Journal for Numerical Methods in Engineering*, **28**, 261–177.

Vinje, T. and P. Brevig (1981). Numerical simulation of breaking waves. *Advances in Water Resources*, **4**, 77–82.

Weisstein, E. W. Pathwise-connected. `http://mathworld.wolfram.com/Pathwise-Connected.html` (1999). From MathWorld–A Wolfram Web Resource.

Welton, W. C. (1998). Two dimensional PDF/SPH simulations of compressible turbulent flows. *Journal of Computational Physics*, **139**, 410–443.

Wong, K. W. (1980). *Manual of Photogrammetry*, Chapter 2: Basic Mathematics of Photogrammetry, pp. 37–102. American Society of Photogrammetry, Fourth Edition.