The integration of two stand-alone codes to simulate fluid–structure interaction in breakwaters

Jan Hendrik Grobler
23817658

Dissertation submitted in fulfillment of the requirements for the degree *Magister Ingeneriae* in Mechanical Engineering at the Potchefstroom Campus of the North-West University

Supervisor: Prof L Liebenberg

November 2013

Abstract

Harbours play a vital role in the economies of most countries since a significant amount of international trade is conducted through them. Ships rely on harbours for the safe loading and unloading of cargo and the harbour infrastructure relies on breakwaters for protection. As a result, the design and analysis of breakwaters receives keen interest from the engineering community.

Coastal engineers need an easy-to-use tool that can model the way in which waves interact with large numbers of interlocking armour units. Although the study of fluid–structure interaction generates a lot of research activity, none of the reviewed literature describes a suitable method of analysis. The goal of the research was to develop a simulation algorithm that meets all the criteria by allowing CFD software and physics middleware to work in unison.

The proposed simulation algorithm used Linux “shell scripts” to coordinate the actions of commercial CFD software (Star-CCM+) and freely available physics middleware (PhysX). The CFD software modelled the two-phase fluid and provided force and moment data to the physics middleware so that the movement of the armour units could be determined.

The simulation algorithm was verified numerically and experimentally. The numerical verification exercise was of limited value due to unresolved issues with the CFD software chosen for the analysis, but it was shown that PhysX responds appropriately given the correct force data as input.

Experiments were conducted in a hydraulics laboratory to study the interaction of a solitary wave and cubes stacked on a platform. Fiducial markers were used to track the movement of the cubes. The phenomenon of interest was the transfer of momentum from the wave to the rigid bodies, and the results confirmed that the effect was captured adequately. The study concludes with suggestions for further study.

Keywords: breakwater, fiducial markers, fluid–structure interaction, PhysX, Star-CCM+

Acknowledgements

I wish to thank:

- The Council for Scientific and Industrial Research (CSIR) for the opportunity to further my studies and for financial assistance.

- The members of the "Ports SRP"-team for their assistance.

- Prof Leon Liebenberg for his invaluable guidance, always provided in a positive spirit.

- My parents for their support.

Table of contents

List of figures

List of tables

List of symbols

A	Area [m^2]
a	Acceleration [m/s^2]
c	Damping coefficient [Ns/m]
c_p	Specific heat [J/kgK]
c_x	Displacement of rigid body in x-direction in a Cartesian coordinate system [m]
c_z	Displacement of rigid body in z-direction in a Cartesian coordinate system [m]
e_0	Total energy [J]
F	Force [N]
$F_{calculated}$	Calculated force [N]
$F_{external}$	External force [N]
F_{ff}	Fluid force [N]
F_g	Gravitational force [N]
$F_f^{pressure}$	Pressure force vector [N]
F_{rbf}	Rigid body force [N]
F_f^{shear}	Shear force vector [N]
F_x	Force component in direction of x-axis in a Cartesian coordinate system [N]
F_z	Force component in direction of z-axis in a Cartesian coordinate system [N]
f	Face or surface element
f_r	Ramp factor
i, j	Indices for an axis in a Cartesian coordinate system
k	Spring stiffness [N/m]
M_y	Moment around the y-axis of a Cartesian coordinate system [Nm]
$M_{y(rbm)}$	Moment around the y-axis of a Cartesian coordinate system fixed to a rigid body [Nm]
m	Mass [kg]
n	Natural number; normal component (if used as subscript)
n_f	Direction vector
n_i *or* n_j	Component of an outward-pointing unit-normal vector on the boundary of a volume along an axis in a Cartesian coordinate system
p	Pressure [Pa]

q_j	Heat flux in direction of an axis in a Cartesian coordinate system [W/m^2]
r	Radius of fiducial marker [pixel units]
t	Time [s]; tangential component (if used as subscript)
t_r	Ramp time [s]
t_s	Release time [s]
u_i or u_j	Velocity in direction of an axis in a Cartesian coordinate system [m/s]
V	Volume [m^3]
v	Velocity [m/s]
v_0	Initial velocity [m/s]
x_i, x_j *or* x	Displacement in direction of an axis in a Cartesian coordinate system [m]
x_0	Initial displacement [m]
α_i	Volume fraction of the i^{th} phase of a multiphase fluid
Δt	Time step [s]
δ_{ij}	Kronecker delta (for definition, see Appendix A)
μ	Friction coefficient
ρ	Density [kg/m^3]
τ_{ij}	Stress tensor [Pa]

Glossary

A4: A surface measuring 210 mm by 297 mm.

Angularity (geology): Description of the corners on a particle. Although it can be numerically quantified, a simple visual chart with up to six categories of angularity is typically used.

Armour unit: Large quarried stone or specially shaped concrete block used as primary protection against wave action.

Bias error: Error that is not due to chance alone, for example when measurements are taken using instruments that have not been calibrated properly.

Constitutive equations: Relates physical quantities that are specific to a material or substance and approximates the response of that material to external stimuli.

Control volume: The smallest division of a fluid domain, also known as a “cell” or “element”. The variables of interest are located at the centroid of a control volume.

Dispersion (statistics): The amount of variability or spread in a data sample.

Dot product: An algebraic operation that takes two sequences of numbers with equal length as input and returns a single number. The dot product of two three-dimensional vectors equals the product of their magnitudes and the cosine of the angle between them.

Elastic deformation: Deformation that is reversible. Once the forces are no longer applied, the object returns to its original shape.

Fiducial marker: A marker placed in the field of view of an imaging system for use as a point of reference or a measure.

Fluke: The triangular blade at the end of an arm of an anchor, designed to catch in the ground. In the study it refers to the two identical parts of a dolos, offset by 90° with respect to each other and connected by the shaft.

Free surface: The surface of a fluid that is subjected to constant perpendicular normal stress and zero parallel shear stress, such as the boundary between two homogenous fluids, for example liquid water and the air.

Histogram: A graphical representation of statistical dispersion.

Index of refraction: The ratio between the speed of light in a vacuum and its speed in another medium. Light bends when moving to a medium with a different index of refraction.

Macro: A single program statement that expands into a larger sequence of computing instructions. It makes interactions with software less tedious and reduces the likelihood of errors.

Middleware: Computer software that provides services to software applications beyond those available from the operating system. Middleware makes it easier for software developers to perform communication and input/output, so they can focus on the specific purpose of their application.

Monotonic: A function *f*, defined on a subset of the real numbers with real values, is called monotonic if for all x and y, $f(x) \leq f(y)$ if $x < y$.

Operating system: A collection of software that manages computer hardware resources and provides common services for computer programs. Examples include Android, Linux, OS X and Microsoft Windows.

Parallel mode: Traditionally flow solutions were obtained by employing a single computer equipped with a single CPU. In recent years simulation times have reduced significantly by spreading the computational load across a number of CPUs working in parallel. This method of operation is known as performing a simulation in “parallel mode”.

Partial differential equations: A differential equation is an equation that relates the derivatives of a (scalar) function. It is called ordinary if the function depends on only a single variable and partial if it depends on more than one variable.

Precision error: Precision (also called reproducibility or repeatability) is the degree to which repeated measurements show the same results. Precision errors are caused by uncontrollable fluctuations in variables that affect experimental results.

Principal axis: In a body fixed reference frame, the inertia matrix of an object can be decomposed into a rotation matrix and a diagonal matrix. The columns of the rotation matrix define the principal axes of the body. If a body has a constant density, the principal axes are the axes of rotational symmetry.

Principal moment of inertia: In a body fixed reference frame, the inertia matrix of an object can be decomposed into a rotation matrix and a diagonal matrix. The diagonal values of the latter are called the principal moments of inertia.

Reynolds-averaged Navier-Stokes equations: Turbulent flow manifests itself as fluctuations around a mean value. Reynolds-averaging of the Navier-Stokes equations separates the terms into mean and fluctuating components. The latter lend themselves to treatment with a turbulence model.

Solution mapping: The transfer of a solution obtained with a previous mesh, to a new mesh. Values for the control volumes of the new mesh are obtained by interpolation.

Standard deviation: Shows how much dispersion exists from the mean value in a data sample. Appendix A contains a mathematical definition of the term.

Steady-state analysis: A simulation where the partial derivatives with respect to time are zero. It is the opposite of a transient analysis.

Surface normal: A vector that is perpendicular to a plane that is tangent to a surface at a given point.

Tessellated: Created through the repetition of a geometric shape with no overlaps and no gaps.

Transient analysis: A simulation where the partial derivatives with respect to time are not zero. It is the opposite of a steady-state analysis.

Ubiquitous computing: Also known as “pervasive computing”, “ambient intelligence”, or “everyware” and describes a situation where machines fit the human environment instead of forcing humans to enter theirs. In such an environment people may engage many computational devices and systems simultaneously, without even being aware of the fact.

Validation: This term may have a variety of definitions depending on the context. For purposes of the study it refers to activities aimed at determining the suitability of an algorithm developed to model a large number of armour units interacting with waves.

Verification: This term may have a variety of definitions depending on the context. For purposes of the study it relates to activities aimed at determining if the algorithm that was developed produced results that were accurate enough for the intended application.

Volume fraction: The volume of a given fluid present in a cell divided by the total volume of that cell.

Abbreviations

2D	Two-dimensional / two dimensions
3D	Three-dimensional / three dimensions
6DOF	Six degrees of freedom
CD-adapco	Computational Dynamics Limited and Analysis & Design Application Co. Ltd.
CFD	Computational fluid dynamics
CICSAM	Compressive interface capturing scheme for arbitrary meshes
CLI	Command line interface
CPU(s)	Central processing unit(s)
CSIR	Council for Scientific and Industrial Research
DDA	Discontinuous deformation analysis
DEM	Discrete element modelling
DFBI	Dynamic fluid body interaction
DVD	Digital video disc
FEA	Finite element analysis
FEM	Finite element modelling
FSI	Fluid–structure interaction
GUI	Graphical user interface
HRIC	High resolution interface capturing scheme
JPEG	Joint Photographic Experts Group
LCP	Linear complementary problem
NSE	Navier-Stokes equations
NVD	Normalized variable diagram
OpenFOAM	Open-source field operation and manipulation
OS X	Unix-based graphical interface operating systems developed by Apple Inc.
RANSE	Reynolds-averaged Navier-Stokes equations
RSRU	Remote Sensing Research Unit
SPH	Smoothed-particle hydrodynamics
SWE	Shallow water equations
VOF	Volume of fluid
MS	Microsoft

1 Introduction

1.1 Background

International trade is vital to the economies of most countries, and a significant amount of this trade is conducted through sea ports. Ships rely on the integrity of harbours to provide shelter and to ensure the safe loading and unloading of cargo. Harbours in turn rely on breakwaters to protect ships and infrastructure during severe storms by absorbing the impact of violent seas and reducing overtopping. Overtopping occurs when the sea penetrates harbour defences and is often associated with damage to ships and infrastructure as well as the disruption of normal operations.

Coastal engineers have a need to analyse existing harbours but also need tools to assist in the design of new ones. They make extensive use of three-dimensional (3D) physical models by which armour units, waves and the topography of the seabed are recreated on a small scale in a hydraulics laboratory. Relatively few of these laboratories are available worldwide, with the Council for Scientific and Industrial Research (CSIR) possessing the only significant facility of this kind in Africa. In addition to increasing demands placed on these facilities to assist with the maintenance

and improvement of existing harbours and the design of new ones, such scale models are generally complex, expensive and time-consuming to build (Cooper, et al., 2008).

Flume and basin experiments have produced a large number of very successful semi-empirical formulations that parameterize key variables. Semi-empirical formulations are very useful but constrained because they cover a limited number of configurations and are accurate only if applied within a certain range. Factors that have an important influence on the stability of breakwater structures are not taken into account, such as non-linear wave–structure interaction and contact forces between blocks (Kaidi, et al., 2012). As an alternative, a number of numerical techniques have been developed in recent years, supported by impressive increases in computer hardware capability.

Numerical approaches can generally be grouped into three categories: those based on the shallow water equations (SWE); particle methods like smoothed-particle hydrodynamics (SPH); and those solving the Reynolds-averaged Navier-Stokes equations (RANSE).

The shallow water equations are obtained by integrating the Navier-Stokes equations over depth, assuming that the horizontal length scale is much greater than the vertical length scale. Under these conditions the conservation of mass implies a small vertical velocity. The conservation of momentum implies vertical pressure gradients that are nearly hydrostatic, and horizontal velocity components that are constant with depth. Integration eliminates the vertical velocity components to yield the shallow water equations. Although computationally very efficient, they find limited application in practice due to the need to satisfy the shallow water assumption.

The SPH method divides the fluid into a set of discrete elements, referred to as particles. A kernel function and characteristic length are introduced and the properties of the particles are smoothed over this length using the kernel function. Any property at a particle of interest is obtained by adding the contributions of surrounding particles. The contribution of each surrounding particle is weighed by its distance from the particle of interest as well as its density. SPH offers the advantages of guaranteeing the conservation of mass and producing a free surface directly when modelling two-

phase flow, but has several disadvantages: a high number of particles is usually required; the use of fixed particle spacing; and very low computational efficiency (Lara, et al., 2008).

Methods based on the RANSE are not constrained by the limitations of the above-mentioned approaches. Unlike the SWE approach, it can accurately simulate wave conditions at any relative water depth and does not have to deal with the inherent discontinuity between individual particles introduced by SPH. As a result, its role as a tool for modelling coastal engineering processes is growing in importance (Higuera, et al., 2013). It requires relatively few assumptions and produces pressure and velocity profiles in three dimensions. Its main disadvantage is its high computational demand, an issue partly addressed by solvers developed to run in parallel on sets of central processing units (CPUs).

RANSE–based methods have proved so versatile that they have branched out to the field of multiphysics – the analysis of processes involving more than one physical effect. Fluid–structure interaction, or FSI, is one such process and may be defined as "interactions of some movable or deformable elastic structure with an internal or surrounding fluid flow" (Bungartz & Schäfer, 2006). Examples of engineering problems involving FSI are abundant and vary from the infamous and large scale such as the Tacoma Narrows bridge collapse (1940); the North Sea flood (1953) and the Indian Ocean tsunami (2004) to the tiny, such as micro-pump design, and the potentially life-saving, such as the simulation of blood flow in arteries.

The interaction between waves and coastal structures is an important FSI research area because it has a significant influence on human activities and was a major area of interest in coastal engineering even before the havoc wreaked by recent tsunamis and hurricanes (Ai & Jin, 2010).

Latham et al. identified hydraulic instability as the main failure mode of breakwaters during severe storms. The lift and drag forces result in rocking, displacement and collisions significant enough to break the concrete units. They focussed their work on numerical simulations of such systems, modelling randomly packed armour units with discrete element codes and coupling a computational fluid dynamics (CFD) code to resolve the wave dynamics (Latham, et al., 2008).

1.2 Motivation for the study

Competition among commercial CFD code developers, supported by the constant growth in computer hardware capability, has resulted in software packages that are versatile, easy to use and computationally efficient. The laborious process of building a solution domain by defining blocks of cells in a command line and then waiting for long periods while the equations are solved on a single powerful CPU, has been replaced by automated meshing, parallel solvers and an intuitive graphical user interface (GUI) which allows the user to watch the solution as it develops. As a result, a number of researchers have employed general-purpose CFD codes with success in the analysis of breakwater structures (Higuera, et al., 2013); (Finnegan & Goggins, 2012). Surprisingly, research on breakwater analysis rarely mentions the two main commercial general-purpose RANSE CFD codes, Star-CCM+ and ANSYS FLUENT.

Coastal engineers looking for commercial CFD codes with an FSI capability find their options severely limited. They are generally looking for an FSI capability that offers the following features:

- Movement in all six degrees of freedom (6DOF)
- Modelling of complex structures that are able to interlock
- Large numbers of units allowed (typically several hundred)
- Ease of use

Many codes that allow 6DOF movement have been developed for store release simulation and prohibit objects to touch (CFD Research Corporation, 1997). Their solvers have also not been developed to accommodate hundreds of independent objects, which places a practical limit on their application in this field.

Discrete element modelling (DEM) has been integrated in some commercial CFD codes and accommodates larger numbers of identical elements. In practice, the number is still limited and the

elements have a simple shape (usually spheres). More complex objects may be assembled using the elements but remain a rough approximation if the desired shape is complex. The interlocking of armour units is often dependent on relatively minor geometric features. Accuracy will be compromised if these geometric features are approximated with simple elements like spheres.

Research codes can be considered as an alternative, but are often difficult to use by outsiders since they were not developed for that purpose. If such codes are made available for general use, significant resources are required to provide technical support, and such resources are usually simply not available.

A need therefore exists for a method that uses commercially available codes to produce a tool for coastal engineers involved in the analysis of breakwater structures.

1.3 Research goals

The goal of this research is to determine if commercial CFD software and physics middleware can work in unison to produce a method for simulating interactions between a relatively large number of rigid bodies and a two-phase fluid. The method must be suitable for the analysis of breakwater structures.

1.4 Scope of the study

The study involved the following:

1. Study the features of Star-CCM+ and PhysX to determine if the desired simulation method is possible in concept.
2. If the concept is feasible, to formulate the simulation method.

3. Investigate the practical feasibility of each element of the simulation method and adapt where necessary.
4. Automate the method so that an entire simulation can be performed without human intervention.
5. Design and evaluate a simple test case to demonstrate the method: two light-weight cubes swept away by an onrush of fluid.
6. Simulate benchmark test cases available in the literature to verify the method.
7. Design and conduct additional experiments, then post-process and interpret the data.
8. Document the results.

The simulation method developed in the study may be utilised in applications other than the analysis of breakwater structures. Any application where one or more rigid bodies interact with one or more fluids, with each other and/or with solid boundaries would be a candidate, provided that the bodies are identical and that the effect of the bodies' movement on the fluid is negligible. The suitability of the method for other applications falls outside the scope of the study.

The requirement that bodies be identical may be relaxed somewhat to allow for a limited number of groups of identical bodies. This would require some changes to the method and although such changes fall outside the scope of the study, it would allow the analysis of breakwater structures consisting of more than one type of armour unit, as shown in Figure 1-1.

The method assumes that forces and moments acting on the bodies as a result of fluid motion, gravity and contact with neighbouring bodies do not change their shape. They remain perfectly rigid and do not break, erode, melt, expand or contract. The method proposed in the study locates areas of high stress in the breakwater where breakages are likely to occur, but does not model the mechanism of breakage.

Figure 1-1 Breakwater with different types of armour units (Photo: Dave Phelp)

No theoretical limit is placed on either the geometric complexity of the bodies or their number, but a sensible mesh is required to capture the flow patterns around the bodies. This places a practical limit on what is feasible. Ultimately users must work within their computer hardware limits and solution time constraints to find the correct balance of geometric complexity, number of bodies and mesh resolution. The method assumes access to an unstructured meshing capability but allows for meshes containing any type of polyhedral control volume.

1.5 Assumptions

The following assumptions are made:

1. The method will be practical to implement using present-day computer hardware.

2. At least one suitable case study is available in the literature for the quantitative assessment of the method.
3. A suitable flume or flow channel will be available for the experiments in the hydraulics laboratory of the CSIR in Stellenbosch.
4. The pictures taken during the experiments will be of high enough quality to enable effective post-processing.

2 Important FSI concepts and fiducial marker technology

2.1 Introduction

The motion of a fluid is described mathematically by the Navier-Stokes equations (NSE). The NSE are based on the principle of the conservation of momentum (equation (2)), although the inclusion of the continuity equation (conservation of mass: equation (1)) and the energy equation (conservation of energy: equation (3)) is usually implied. These equations are shown below in index notation, where $i \in \{1, 2, 3\}$; $j \in \{1, 2, 3\}$ (Löhner, 2008). Appendix A explains the mathematical notation used.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_j\right] = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}\left[\rho u_i u_j + p\delta_{ij} - \tau_{ij}\right] = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho e_0) + \frac{\partial}{\partial x_j}\left[\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}\right] = 0 \tag{3}$$

These equations can be solved analytically in some cases, but have no known general analytical solution. They do however lend themselves to numerical solution after a number of assumptions are made, particularly regarding the stress tensor (τ_{ij}). Constitutive equations are introduced to close the set.

The finite volume method provides a means to evaluate partial differential equations as algebraic equations and is widely used in CFD to obtain numerical solutions for the NSE. The term "finite volume" refers to the volumes surrounding a number of discrete node points where values are calculated. The volumes are chosen such that they completely fill the flow domain without any overlap. The partial differential equations are integrated across the set of volumes using the divergence theorem to convert the volume integrals to surface integrals. For example, if the divergence theorem (as defined in Appendix A) is applied to equation (2), it takes the following form:

$$\frac{\partial}{\partial t}\int_V (\rho u_i)dV + \oint_A (\rho u_i u_j)n_j dA + \oint_A p\delta_{ij}n_j dA - \oint_A \tau_{ji}n_j dA = 0 \qquad (4)$$

This formulation implies that changes in the flow property of interest over time in any given volume are now determined by evaluating properties at the boundary surface only. Information is also required regarding the surface normal and the surface area. The task is made significantly easier if the volumes take the shape of polyhedrons since the surface normal is then constant across each face and the face surface area is easily calculated. If the volumes are small compared to variations in the flow properties, it is reasonable to use a single, approximated flow property value for the whole face. This assumption introduces a discretization error, but reduces the governing equations to algebraic expressions which lend themselves to numerical solution on a computer.

A myriad methods have been developed to make adaptions to this basic technique for specific purposes. These include different methods to approximate flow properties at the faces; performing Reynolds-averaging to enable the introduction of turbulence modelling; modifications to allow the modelling of multiphase flow; etc. (Löhner, 2008)

Subsequent sections in this chapter will focus on the modelling of two-phase flow with the NSE (2.2); modelling the effect of fluid motion on rigid body dynamics (2.3); attempts to model fluid–structure interaction with commercial CFD codes (2.4); and the experimental validation of such methods.

2.2 Two-phase flow simulation

Liquids and gases represent two different phases of matter and may appear to be very different to a casual observer dealing with them in his/her everyday life. They are, however, both fluids and their behaviour is described by the same governing equations. In spite of this fact, special care is required when modelling two-phase flow, especially in the region where the fluids meet.

Two-phase flow is often modelled with the volume of fluid (VOF) method. The VOF method is a numerical technique for tracking and locating the free surface between immiscible fluids. It is not a stand-alone flow solving algorithm and is therefore usually employed in conjunction with the NSE. The VOF method uses an additional transport equation to determine how much of the background fluid is present in any given control volume. The control volumes that are partially filled with the background fluid are deemed to contain the free surface.

Two difficulties are encountered when discretizing the additional transport equation: limiting artificial diffusion of the interface profile and assuring monotonic changes in the values of the variables. Geometric interface reconstruction was introduced to address these difficulties but requires a considerable increase in computational effort. An alternative approach was developed in which the aforementioned difficulties are handled by properly choosing the discretization scheme. They are known as “high resolution schemes” and two popular versions have emerged: the “Compressive Interface Capturing Scheme for Arbitrary Meshes” (CICSAM) (Ubbink & Issa, 1999) and the “High Resolution Interface Capturing scheme” (HRIC) (Muzaferija, et al., 1999). In general, both of these high-resolution schemes show good agreement with experimental data (Wacławczyk & Koronowicz, 2008).

A commercial CFD code called Star-CCM+, developed by Computational Dynamics Limited and Analysis & Design Application Co. Ltd. (CD-adapco), was used in the study. It uses the VOF

method to model mixtures of immiscible fluids and therefore requires a numerical mesh that is fine enough to capture the interface between the phases.
It assumes that all phases share the same velocity, pressure and temperature fields so the governing equations are solved as they would be for a single phase fluid, except that the physical properties of the fluid are changed to account for the contributions of the constituent phases.

For example the density (ρ) and specific heat (c_p) are calculated as follows, for a given volume fraction (α_i), where i refers to the i^{th} phase (CD-adapco, 2012):

$$\rho = \sum_{i=1}^{n} \rho_i \alpha_i \tag{5}$$

$$c_p = \sum_{i=1}^{n} \frac{(c_p)_i \rho_i}{\rho} \alpha_i \tag{6}$$

Other physical properties are treated in a similar way.

The normalized variable diagram (NVD) is used to ensure that the convective transport of properties remain bounded. The value of a property in any given cell must be bound by (i.e. lie between) the value of that property in the upwind cell and the value in the downwind cell, provided that no sources or sinks are present. Certain properties also have physical limits, such as the volume fraction, which has to remain between 0 and 1. Star-CCM+ uses the HRIC scheme to ensure a sharp interface between phases.

2.3 Fluid–structure interaction

FSI was broadly defined in the previous chapter as “interactions of some movable or deformable elastic structure with an internal or surrounding fluid flow” (Bungartz & Schäfer, 2006). The solid structures may therefore be rigid or elastically deformable and the fluid may be gaseous, liquid or a mixture of the two. In addition, the fluid may surround the solid structure or be located within it and the effect of the structure’s deformation on the fluid may or may not be significant.

Numerical analysts have adopted a number of approaches for dealing with FSI with the aim of narrowing down the above-mentioned definition to suit specific problems. In the simplest approach, the solids become part of the stationary geometric features of the domain boundary and affect the fluid only through their interaction with these wall boundaries.

In another approach, known as one-way coupling, only the effect of the fluid on the solid is modelled and solids are allowed to move. Two-way coupling allows feedback of the elastic motions of the structure into the fluid solver (Paik, et al., 2009). One-way and two-way coupling both form part of a group known as partitioned methods, where the fluid and solid domains are solved separately. In contrast, monolithic methods take both these domains into account at once. A single, non-linear, discrete system of equations is solved as a whole (Ryzhakov, et al., 2010).

When simulating the interaction of waves with a breakwater, the free surface of the sea water is modelled and the armour units are regarded as rigid bodies since they are usually made of solid rock or concrete (Latham, et al., 2008). In the context of the study, "fluid–structure interaction" therefore refers to the analysis of two-phase flow around rigid bodies using partitioned methods, unless otherwise indicated.

As stated earlier, a flow domain has to be divided into a number of control volumes in order to solve the flow field using CFD. The collection of control volumes is referred to as the mesh and it has to change in some way to accommodate rigid body motion. Researchers have developed a number of strategies to deal with this issue (Hadžić, et al., 2005).

Strategy 1: Move the entire mesh without deforming the control volumes. This strategy can be used for modelling only a single body moving in an infinite domain and require special treatment of the far-field boundaries.

Strategy 2: Divide the flow domain into two regions. One region is fixed to the body and the other to the far-field boundaries. The mesh is allowed to deform in the vicinity where the two regions meet, as shown in Figure 2-1. Excessive mesh deformation will result if the body motion is not moderate.

Strategy 3: Divide the flow domain into two regions. One region is fixed to the body and the other to the far-field boundaries. Regenerate the mesh in the vicinity where the two regions meet instead of deforming the mesh. This strategy requires an automatic meshing capability as well as solution mapping.

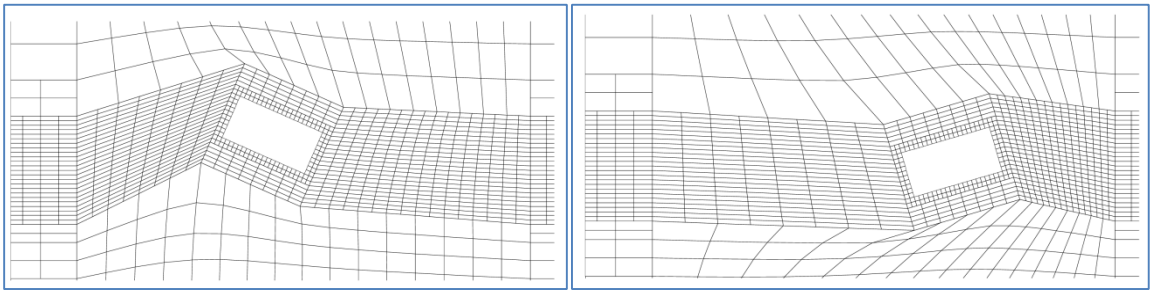


Figure 2-1 Two-region strategy with mesh adaption (Hadžić, 2005)

Strategy 4: Divide the flow domain into three regions. The first region contains blocks attached to the rigid body. They are allowed to rotate, but not to translate. The second region contains blocks that do not rotate, but are allowed to translate with the rigid body. The third region is attached to the far-field boundaries and expands or contracts to accommodate the translation of the second region. The regions are shown in Figure 2-2: the first region is shown in red; the second region in green and the third region in yellow and blue. This strategy preserves the mesh quality, but does not allow for rigid bodies to touch each other.

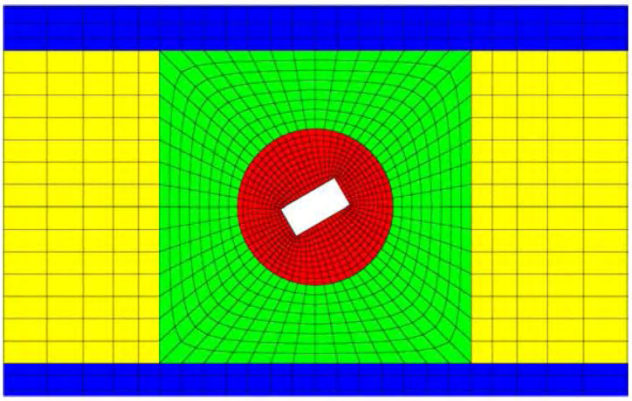


Figure 2-2 Three-region strategy: 1st (red); 2nd (green); and 3rd (yellow and blue)

Strategy 5: Use overlapping meshes. Divide the flow domain into two regions. One region is smaller and fixed to the body; the other is larger and covers the whole flow domain. In Figure 2-3 the larger, background mesh is shown in red and the smaller mesh surrounding the body is shown in black. The meshes are allowed to occupy the same space at the same time and information is exchanged between the regions using sophisticated interpolation algorithms. This strategy offers great flexibility, but conserving properties such as mass across the regions is difficult when using arbitrary unstructured meshes.

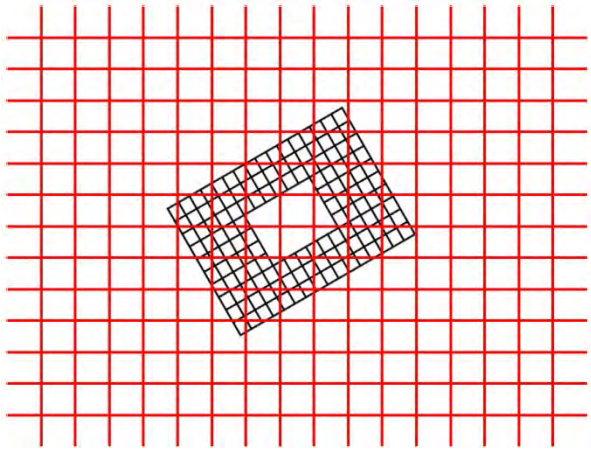


Figure 2-3 Overset meshes: background mesh in red; body-fixed mesh in black

Strategy 6: Re-mesh the entire domain after every time step. This strategy is computationally expensive but allows for arbitrary movement of the rigid body in the flow domain. This strategy requires an automatic meshing capability as well as solution mapping.

All of the above-mentioned strategies can be used to model the moderate motion of a single rigid body such as a ship floating in the sea. A typical breakwater consists of a large number of interlocking bodies where individual armour units may dislodge and tumble down the structure. Most of the mesh adaption strategies will be unsuitable for such an analysis. Strategy 1 prohibits relative movement of units outright while strategy 2 does not allow the mesh to be deformed enough to model a tumbling armour unit. Strategy 4 allows rotation in excess of 360° but only for bodies separated by a significant distance. Commercial CFD codes that allow overlapping meshes (strategy 5) prohibit wall boundaries from touching at present (CFD Research Corporation, 1997); (CD-adapco, 2012).

Only strategies 3 and 6 offer a method flexible enough to handle a large number of interlocking armour units while allowing individual units to tumble down. The study employed strategy 6 but the computational demand could have been reduced by employing strategy 3: regenerate the mesh only in the vicinity of the breakwater while the rest of the mesh remains fixed.

Solutions obtained from a two-phase flow solver will yield force and moment data for each armour unit and a suitable mesh adaption strategy will allow them to move, but exactly how they will move still has to be determined. Approaches to deal with this issue are discussed below.

A general method for modelling the behaviour of a large number of (relatively small) particles was originally developed to study rock mechanics (Cundall & Strack, 1979). A number of related numerical approaches based on this method have since been developed and they are generally referred to as the discrete element method or DEM.

The solid phase in a DEM analysis is assumed to consist of a relatively large number of elements, or particles. The forces between the particles are calculated and these forces are used to compute their motion over a relatively small time interval using Newton's laws of motion. The positions of the particles are then updated and the process is repeated for the next time increment.

The forces between the particles are typically calculated by assuming spring, dashpot and slider components at the contact points between the particles, as shown in Figure 2-4, where μ and c are the friction and damping coefficients, respectively, and k the spring stiffness at the contact point between two arbitrary particles i and j. The subscript n refers to normal components to yield the normal force, F_n, and t refers to the tangential components to yield the tangential force, F_t.

The original DEM has seen many improvements over the years and has found application in the fields of physics, chemistry and engineering. CFD and the DEM have also been combined with success by Tsuji et al. in a method that has become known as CFD-DEM (Tsuji, et al., 1992).

Research showed that the assumptions made in the calculation of the inter-particle forces have a significant influence on the method’s accuracy. For example, the accuracy obtained with non-linear

spring and damping models was found to be much higher compared to the accuracy obtained with linear spring and damping models (Zhang & Whiten, 1996).

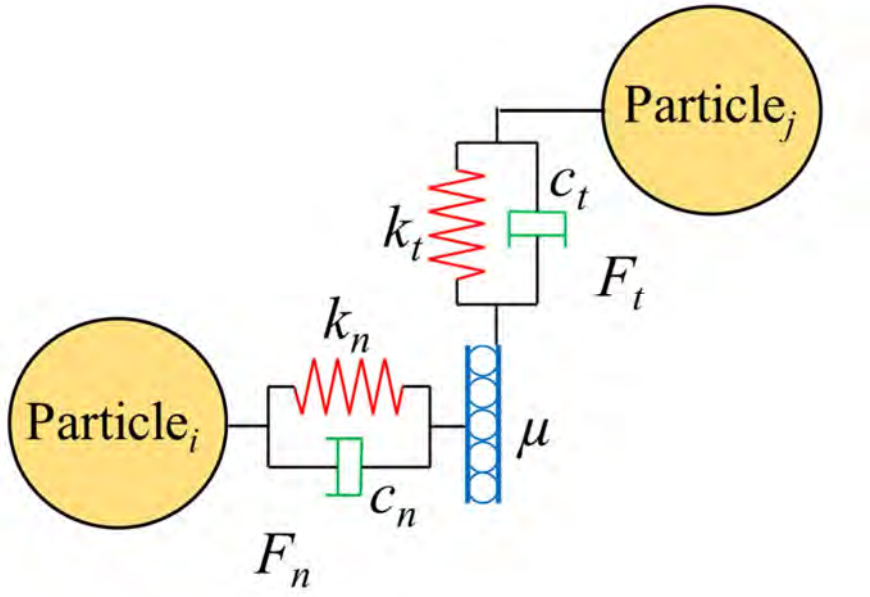


Figure 2-4 Spring, dashpot and slider components (Hongchang, et al., 2012)

The assumption of spherical particles, or circular particles in two dimensions (2D), also has certain implications for the method's accuracy. The internal friction angle and shearing resistance of spherical particles are less than those of non-spherical particles. The direction of contact normal forces on a spherical particle is always toward its centre and therefore makes no contribution to the moment – rotation is completely dependent on tangential forces. As a result, accuracy is compromised if distinct particles with various, non-spherical shapes are modelled as spheres. A DEM analysis of four groups of grains proved that the angularity of the grains significantly affected the mechanical behaviour of the granular material (Mollanouri Shamsi & Mirghasemi, 2012).

This limitation of the DEM can be mitigated by allowing groups of particles to be rigidly clumped together as shown in Figure 2-5. Here a single sunflower pith is optically scanned in 3D and the resulting surface is processed to produce a simplified surface. The simplified surface is used to generate an object homogeneously filled with spherical particles and a DEM simulation containing 460 such objects can then be performed (Pennec, et al., 2013).

DEM particles with shapes that deviate from a sphere have been suggested as an alternative to the use of aggregate particles when modelling granular media (Hosseininia, 2012). The irregular convex-polygonal shaped particles used by Hosseininia are shown in Figure 2-6.

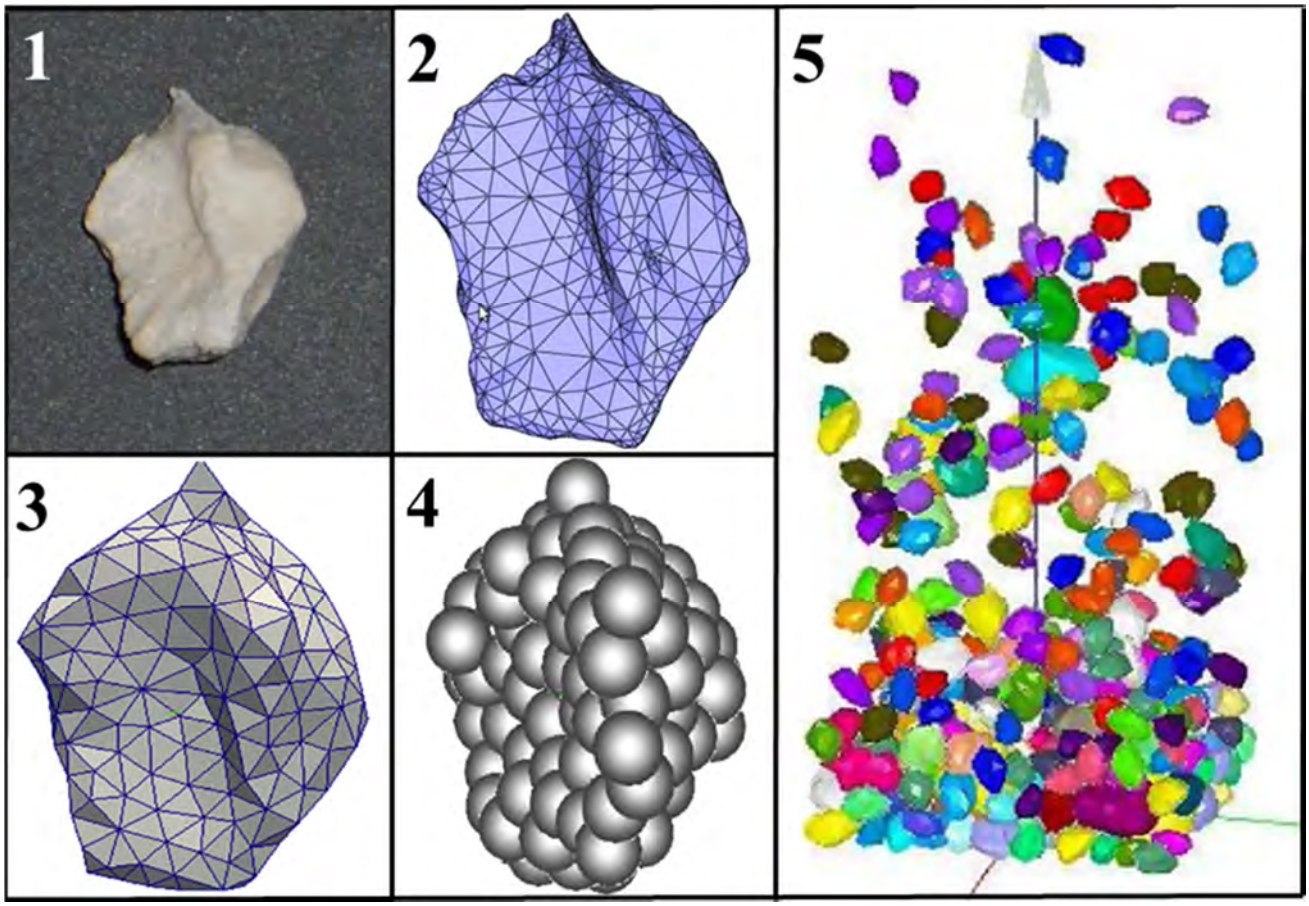


Figure 2-5 Generation of aggregate particles in five steps (Pennec, et al., 2013)

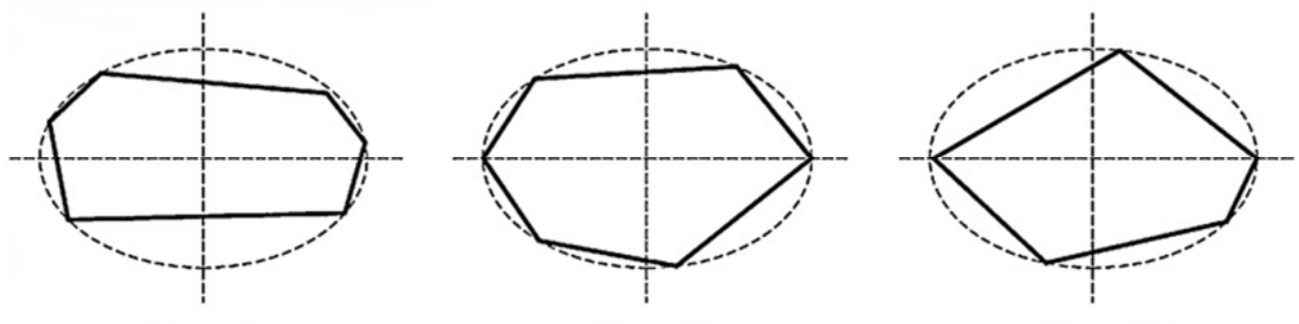


Figure 2-6 DEM particles with irregular shapes (Hosseininia, 2012)

Latham et al. have modelled armour units successfully as aggregate particles consisting of spheres but regarded FEMDEM as a more realistic approach (Latham, et al., 2008). FEMDEM is the combination of the finite element method (FEM) and the DEM and allows a far more accurate representation of angular bodies.

Kaidi et al. used discontinuous deformation analysis (DDA) for the simulation of rigid body motion (Kaidi, et al., 2012). DDA parallels the FEM and was originally proposed by Shi (Shi, 1988). It uses the principle of minimum potential energy to solve the equations of motion for a number of independent blocks. DDA is considered to be a main branch of the DEM.

In DDA, the contacts between the blocks are modelled using one of three methods: the penalty method; the method of Lagrange multipliers; or the augmented Lagrangian formulation (a combination of the first two methods). Solutions employing the method of Lagrange multipliers become computationally prohibitive when large numbers of contacts are modelled (Kaidi, et al., 2012). Likewise, the CPU time required for solutions using the penalty method increases non-linearly with an increase in the number of colliding bodies, placing an upper limit on the number that can reasonably be modelled.

Methods involving the solution of a linear complementary problem (LCP) have also been extensively used to model resting contact forces in rigid body simulations. The LCP-method determines the resting contact forces analytically to prevent interpenetration while the penalty method achieves the same goal by applying virtual springs to surfaces that are in contact (Drumwright, 2008). In contrast to the penalty method, LCP-methods seem to show a linear relationship between the required CPU time and the number of colliding bodies (Madsen, et al., 2007). Simulations involving one million rigid bodies have been successfully demonstrated using the LCP-method (Negrut, et al., 2011).

PhysX is software that simulates rigid body motion using the LCP-method. It is technically more accurate to refer to it as middleware, since it fulfils its role between the operating system on the one hand, and the software developer on the other. By providing additional services and applications in the background, it extends the operating system and allows software developers to focus on the

specific purpose of their application. The purpose of PhysX is to handle the complex physics interactions required in modern computer games, saving game developers the effort of writing their own codes to handle the physics. PhysX was developed by Ageia Technologies, Inc. which was acquired by the Nvidia Corporation in February 2008. It is available for a wide range of operating systems and products: Microsoft Windows, OS X, Linux, PlayStation 3, Xbox 360 and the Wii.

A code such as PhysX has to balance the need for speed, accuracy and stability. Since it is required to solve physics interactions in real time, the perception might be created that speed, rather than accuracy is the primary goal (Negrut, et al., 2011). PhysX has however, been successfully used to model a vibration particle-screening machine (Ai-min, et al., 2008). Various parameters affecting screening efficiency were varied: vibration amplitude, frequency and direction; surface length and inclination; as well as initial particle distribution. Results showed that the virtual model correctly simulated the screening process and that there was basic agreement with theoretical predictions. Ai-min et al. concluded that the virtual model was useful for design and theoretical research.

PhysX allows the creation of bodies with arbitrary shapes which may be flexible or rigid, though only the latter will be discussed here. Bodies, or actors as they are called, may have concave shapes provided that they are concatenations of convex shapes. The surfaces are tessellated with polygons (usually triangles or quadrilaterals) constructed from previously defined vertices.

Each actor has properties comprised of linear and angular quantities. The linear quantities include their mass (a scalar) as well as linear velocity and position (both 3-component vectors). The position vector gives the actor’s centre of mass relative to its reference frame. The angular quantities include their principal moments of inertia, angular velocity (both 3-component vectors) and orientation (a 3x3-matrix). The orientation matrix gives the principal axes relative to the actor’s reference frame (4Front Technologies, 2010).

Collisions between actors are detected using a hierarchical method involving bounding boxes, and a contact graph is used to decide the order in which actors interact. Interactions with immovable objects are given priority (Hahn, 1988). If a part of one actor overlaps with another it signals a

collision. The movements of the actors are then “backed up” to the time that the actors first touched, and collision dynamics is used to calculate new positions and orientations.

An actor will be regarded as “asleep” if its velocity drops below a user-specified threshold value. This is done to eliminate relatively small, but persistent bouncing movements – also known as “chatter”. An actor will “wake up” if it receives energy in excess of a threshold level (Gledhill, et al., 2012).

2.4 Commercial computational fluid dynamics codes

A number of researchers have applied general-purpose commercial CFD codes to model FSI in the coastal engineering environment. The research most relevant to the study is outlined below.

The commercial CFD code COMET was successfully used to model the motion of a rigid body floating in a wave tank (Hadžić, et al., 2005). The flow solution was obtained using the RANSE and coupled to the solution of the rigid body motion with a user-coded interface. The focus of the research was on ship hydrodynamics and the meshing strategy was suitable for modelling a single rigid body, not a large number of interlocking units. Only 2D cases were analysed. The mesh was refined in the vicinity of the free surface and around the rigid body with a relatively coarse mesh in distant regions and in the air. The time step was kept small to keep the temporal discretization error smaller than spatial discretization error. The numerical and experimental results compared well.

Finnegan and Goggins used the commercial RANSE CFD code ANSYS CFX (Release 12.1) to model a floating truncated vertical cylinder under the influence of a linear wave. In one simulation, the position of the cylinder was fixed and in another it was able to move vertically under the influence of the hydrodynamic force. The results were compared to an analytical solution based on the method of separation of variables, and good correlation was achieved (Finnegan & Goggins, 2012).

The popular open source general-purpose CFD code OpenFOAM has also been used to model two-phase flow. Li and Lin claim to have improved the existing surface capturing scheme in OpenFOAM by introducing a new two-phase 3D code called “interFoam”. The code uses a k-ε turbulence model and is based on the RANSE, discretized using the finite volume method. The CICSAM method is used to capture the free surface. Regular and irregular waves were modelled as well as their impact on a fixed floating structure. The results were compared to experiments in a wave tank and showed good agreement (Li & Lin, 2012).

Other researchers also used “interFoam” to demonstrate the practical application of OpenFOAM to simulate coastal engineering processes (Higuera, et al., 2013). Five cases that are relevant in coastal engineering were investigated, including a solitary wave interacting with a vertical structure and run-up on a conical island. Comparisons with experimental benchmarks were made and showed good correlation as regards to wave breaking, run up and undertow currents.

Dentale et al. used the code FLOW-3D to analyse a relatively large number of armour units and could identify individual units where the hydrodynamic forces were similar to the unit's weight. Such units could then be flagged as being at risk of moving or even being dislodged, but their actual movements were not modelled (Dentale, et al., 2012).

It is interesting to note that no research was found during this literature study where one of the two main commercial general-purpose RANSE CFD codes, Star-CCM+ and ANSYS FLUENT, was used for breakwater analysis.

2.5 Fiducial marker technology

FSI implies movement and therefore experiments investigating such phenomena need to measure how the surfaces defining object boundaries change as a function of time. In the case of rigid bodies, the surfaces will only translate and rotate along with the bodies themselves, but when dealing with non-rigid bodies, surfaces may also move as a result of deformation.

A popular method to measure movement is by means of accelerometers, but the method has several disadvantages when analysing the interaction of waves with breakwaters. Scaled armour units are

usually less than 50 mm in size and placing accelerometers on them may significantly affect their inertial properties and interlocking characteristics. It may also restrict their movement and become prohibitively expensive in larger tests involving hundreds of units.

Image processing of photographs taken during the experiment offers a non-invasive alternative. Relatively small movements can be identified by using flicker animation. Photographs taken before and after the movement are displayed in rapid succession, making the differences between the images obvious to an observer. Although this method has been used with success in some applications (Berger, et al., 2000), it remains a 2D tool suitable for qualitative assessments only.

Quantitative data can be obtained using photographs of objects moving in 2D. An uncertainty of approximately 5% for translation and approximately 9% for rotation can be expected for moderate motion if the position of a rigid body is determined from digital images taken at an assumed frequency of 60 Hz (Hadžić, et al., 2005).

Fiducial markers have the ability to provide quantitative data of movement in 3D by revealing their orientation and position relative to an observer. They are defined as “special geometric patterns that are used as reference points in machine vision systems” and have traditionally been used to align printed circuit boards during automated optical inspection (Vieira, et al., 2008). Of late, they have found application in other fields, such as ubiquitous computing.

Fiducial markers may take many forms. Some designs are square or circular while others use highly irregular shapes (see Figure 2-7). Owen et al. formulated the following criteria for a good fiducial marker design (Owen, et al., 2002):

- An ideal fiducial image should support the unambiguous determination of position and orientation relative to a calibrated camera.
- The image should not favour some orientations over others.
- The image must be a member of a set of images that are unlikely to be confused so that a large space or set of objects can be uniquely marked.
- The image must be easy to locate and identify using fast and simple algorithms.
- Images must function over a wide camera capture range.

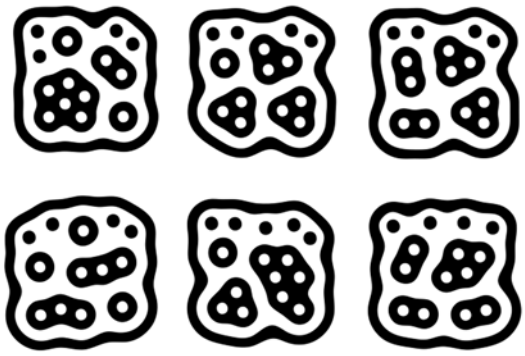


Figure 2-7 Fiducial markers with irregular shapes (Sourceforge, 2009)

Bose and Amir identified a circular shape as the one of the best due to its compactness, low maximum error of centroid location, and independence of camera orientation (Bose & Amir, 1990). Efrat and Gotsman determined that sub-pixel accuracy in the order of $r^{-0.5}$ is achievable with circular shaped fiducial markers, where r equals the fiducial marker's radius in pixel units (Efrat & Gotsman, 1994).

Vieira et al. achieved sub-millimetre accuracies from an A4-scale scene photographed with a 6-megapixel camera (Vieira, et al., 2008). Figure 2-8 shows two examples of the fiducial markers they used, which were based on an earlier design (López de Ipiña, et al., 2002).

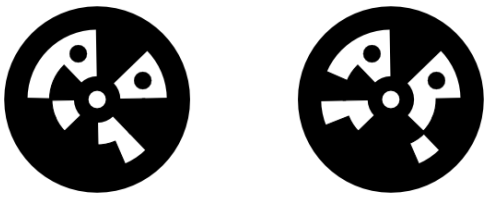


Figure 2-8 Fiducial markers with circular shapes (Vieira, et al., 2008)

The surfaces of their fiducial markers are divided by concentric circles into three bands and then separated into eight sectors. Three dots (two black and one white) are placed on the surface so that they form a 90° angle. The purpose of these dots is to define a local coordinate system.

The sector elements in the remaining 270° are coloured black or white to spell out a unique binary number used for identification. The design of a fiducial marker is usually a trade-off between the number of unique codes needed for identification and the positional accuracy required. The design shown in Figure 2-8 allows 1024 unique patterns while designs allowing 32768 distinct patterns have been proposed (Naimark & Foxlin, 2002).

A circular fiducial marker will appear as an ellipse to an observer viewing it at an angle, with the direction of the major and minor axes dependent on its orientation relative to the observer. Through sophisticated image processing techniques, the target is identified and its distance and orientation relative to the observer is determined (Shiu & Ahmad, 1989).

2.6 Conclusion

The study builds on the work of other researchers and will address aspects not previously investigated. Mesh adaption through regeneration is required to accommodate the complex movement of the armour units and to allow for their tendency to interlock with other units in the breakwater structure. The LCP-method is well suited for simulating the interaction of a large number of rigid bodies. Fiducial marker technology offers a method to track the movement of objects and would be particularly convenient when dealing with model sizes typically found in experiments conducted in a hydraulics laboratory.

3 Design of simulation algorithm

3.1 Introduction

A concept design was developed for a simulation algorithm that would allow two stand-alone codes to work in unison. The algorithm had to be suitable for modelling a large number of armour units interacting with waves. The requirements for such an algorithm are discussed and one is proposed. Since automation is important, a detailed discussion is included showing how each code, as well as the operating system, is managed to create a process that requires minimal user intervention. The successful validation of the algorithm is demonstrated with two simple test cases.

3.2 Concept design

At present, no single commercially available code can readily simulate a large number of geometrically complex rigid bodies interacting with a two-phase fluid and with each other. Although it is technically feasible to add such features to existing commercial CFD codes, it is unclear if (and when) this will happen. As a result, researchers have opted for an approach where two stand-alone

codes work in unison to obtain the desired effect, as many of the examples mentioned in Chapter 2 show.

It was decided to use two stand-alone commercial codes in the study since the development of suitable in-house codes was considered too expensive. Commercial CFD code developers with a significant market share can afford to employ specialists. These specialists ensure that each feature of the commercial software is properly designed, tested and maintained. As a result, such codes usually run efficiently regardless of the hardware or operating system used; the solvers are versatile and run properly in parallel mode; the GUI is well designed and easy to use; powerful post-processing features are available; and expert technical support is offered.

Flow simulations involving meshes of average size typically take several hours to produce a converged result and may even require several days. If such simulations use two stand-alone codes that frequently exchange information, it is highly desirable that the process takes place with minimal user input. Ideally, the user must only be required to start the simulation and to post-process the results. The simulation algorithm must be capable of producing a converged flow solution autonomously.

To achieve this goal the mesh must be regenerated automatically if the geometry changes. Either block structured or unstructured meshes can be generated automatically, with the former restricted to cases where either strategy 2 or strategy 4 (as described in section 2.3) is used to adapt the mesh. Block structured meshes were not required in the study and therefore the automatic mesh generation features offered by modern commercial CFD codes were available.

Other minimum requirements include the ability to model two-phase flow (preferably using the VOF method) and having some mechanism through which commands can be executed during the simulation process. The commands must trigger regeneration of the mesh; mapping of previous solutions to the new mesh; collection of force and moment data; and exporting of such data in a suitable format. Features that are desirable, but not essential, include an easy mechanism through which boundary values can be manipulated to simulate wave trains; the ability to perform

simulations in parallel mode; and the option to export graphical data while the simulation is in progress.

In addition to the transfer of solution data between the two stand-alone codes, some signalling mechanism is required to ensure proper coordination. Once the CFD code has finished simulating a time step, it must halt its own operations and signal the physics middleware to initiate a calculation using the latest data. The physics middleware must signal the CFD code when it is finished and wait for the next signal indicating that the motion of the rigid bodies for the next time step may be calculated.

The simulation algorithm should accommodate situations where each of the stand-alone codes is launched on a separate computer using a different operating system. This can be done by using a signalling and data transfer method that is platform independent. The use of simple text files was an attractive option.

The commercial CFD code Star-CCM+ conforms to all the above-mentioned requirements, including both the essential and the non-essential (but desirable) ones. Star-CCM+ is developed and distributed by CD-adapco, a multinational computer software company that is best known for its CFD products. Star-CCM+ emerged in 2004 from a complete rewrite of its predecessor, STAR-CD, and has grown to become the flagship product of the company.

From the perspective of the study, perhaps the best feature of Star-CCM+ is the ease with which a sequence of instructions can be recorded as a macro. The user presses a single button and all subsequent instructions are recorded in a file. The file contains statements written in the Java programming language and can be edited and made available for future simulations.

PhysX is implemented through a program written in C++ that includes selected PhysX header files. The header files give access to the required PhysX functions while those that are part of C++ remain available. This enables all the data exchange and automation requirements to be met.

All the above elements of the simulation algorithm are shown schematically in Figure 3-1

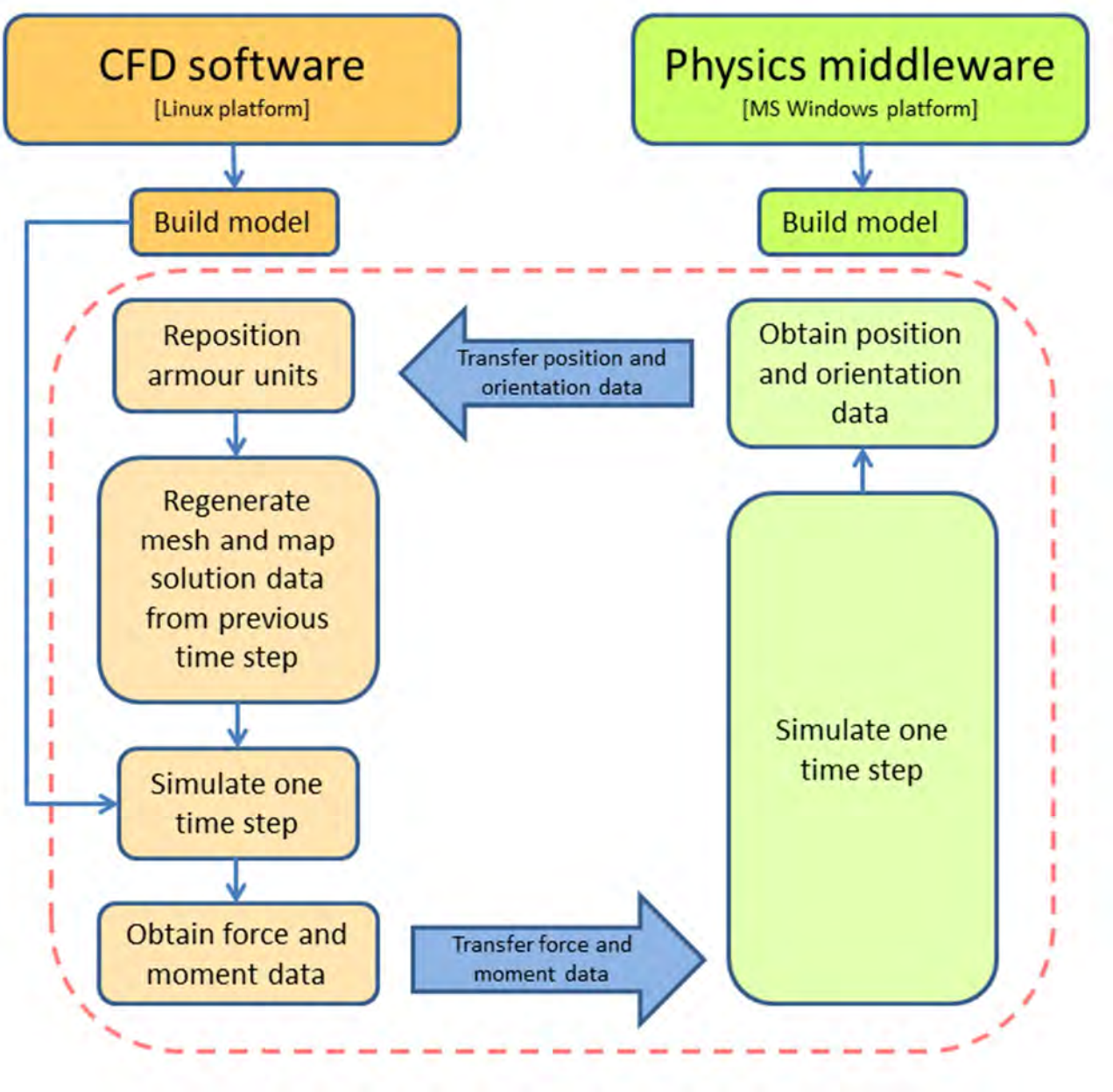


Figure 3-1 Concept design of simulation algorithm

The use of two stand-alone codes implies that the fluid motion calculation at any given time step is decoupled from the rigid body motion at the same time step, which implies that the method is

inherently partitioned. The coupling is also one-way: fluid forces affect the motion of the rigid bodies but their movement does not affect the flow, apart from taking up different positions in space.

3.3 PhysX automation

The models created using the physics middleware and the CFD software must be identical in certain key respects to ensure that the same flow problem is modelled in both codes. The geometry of the armour units and their initial position and orientation must be the same, as well as the direction and magnitude of gravity. In contrast, only the CFD model will contain a mesh and fluids.

The armour unit used in the study is called a dolos and was designed in the Republic of South Africa in 1963 (Bakker, et al., 2003). As mentioned previously, all shapes used in PhysX must be convex but they may be concatenated in ways that will result in an actor with a concave shape. The shape of a dolos is concave but can be represented by a combination of convex shapes, as shown in Figure 3-2. A similar approach should work for all the popular armour unit designs, although some might prove more challenging (see Figure 3-3).

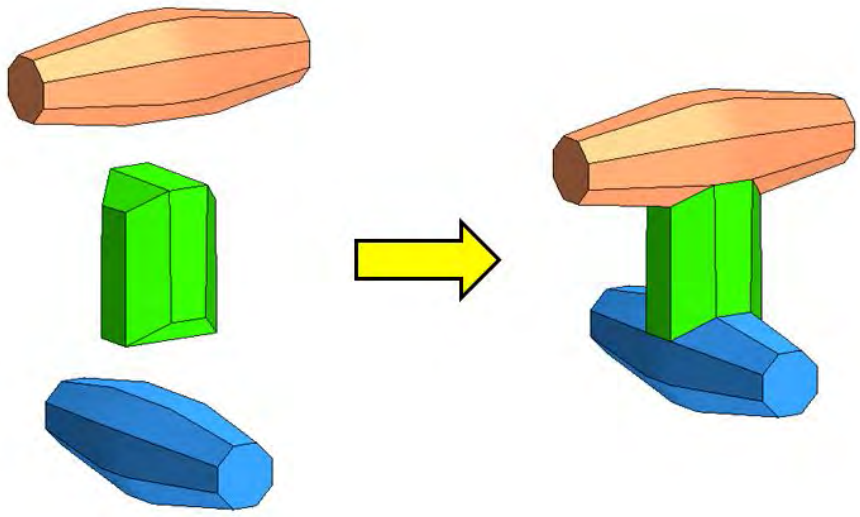


Figure 3-2 The dolos armour unit as a concatenation of convex shapes

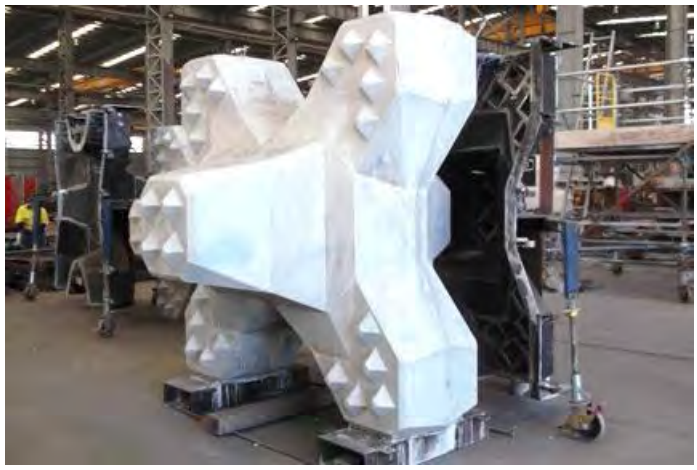


Figure 3-3 The Accropode II (Chevron Australia, 2013)

PhysX makes use of surfaces tessellated with polygons (usually triangles or quadrilaterals) that in turn are constructed from previously defined vertices. The generation of the critical vertices that define the desired shape is therefore the first step in the process. If provided with a suitable set of vertices, PhysX will create the polygons necessary for the convex shape through a process called “cooking”. The geometries used in the study consisted of two parts, the domain boundary and the armour units, and both were created using this process. A total of 84 vertices was required to model the dolos shown in Figure 3-2: each of the flukes required 34 vertices and the shaft 16.

The motions of the armour units are calculated by PhysX as part of a computer program written in C++. Once the C++ program is compiled into an executable format and run, it carries out the following tasks:

1. Read a number of user defined variables contained in a text file (“general.txt”) as well as the starting positions and orientations of the armour units (“pose.txt”)
2. Poll the signal files (“physx_execute.txt” and “simulating.txt”) to determine if PhysX must perform a calculation
3. Wait, continue with Task 4 or exit the program, depending on the content of the signal files
4. Read in the latest force and moment data (“force.txt”)

5. Obtain new positions and orientations with PhysX for each armour unit and write them to a text file (“pose.txt”)
6. Place a “0” in the file “physx_execute.txt” to prevent PhysX from performing a calculation
7. Signal Star-CCM+ to simulate the next time step by placing a “1” in the file “ccm_execute.txt”

The first task is performed only once and the program, called “Breakwater_Basic.exe”, loops through tasks 2 to 7 unless instructed to terminate by the file “simulating.txt”. Please note that the program makes no provision for offsets between the coordinate systems of the Star-CCM+ and PhysX models. Appendix B contains more details regarding the content of the files mentioned above.

3.4 Star-CCM+ automation

Commercial CFD software usually provides the user with a GUI to ease the process of building meshes, selecting physical models and specifying boundary conditions. It is sometimes desirable to use the software without employing the GUI, for example if a set of simulations has to be performed at a time when no users are available to start each subsequent simulation manually.

In such cases a command line interface (CLI) may be used, if available. When a CLI is used commands consisting of a sequence of key words are entered in a terminal window as lines of text. This is sometimes referred to as running the software in "batch mode".

It is possible to run Star-CCM+ in batch mode and the name of a Java script (or macro) may be specified as part of the batch command. If a macro is not specified, the simulation will be performed and the results will be saved using a default file naming convention. If a macro is specified, only the commands contained in the Java script will be executed, but the script may include commands to run the solver for a specified number of iterations and to save the simulation file.

Star-CCM+ automation was achieved by running the code in batch mode twice during each cycle and specifying a different macro in each case. The first macro places the armour units in the correct positions as specified by PhysX and the second macro calculates a new set of forces and moments for each armour unit and writes the data to a text file.

The first macro, contained in a file called “j00014.java”, performed the following tasks:

1. Provide the names of files where important information is stored and define key variables
2. Import the position and orientation data for each armour unit, as supplied by PhysX
3. Create the correct number of armour units and translate and rotate each one correctly
4. Generate a surface representing the armour units and the domain boundaries
5. Export this surface in a format that can be read by the second macro

The second macro, contained in a file called “j00015.java”, performed the following tasks:

1. Provide the names of files where important information is stored and define key variables
2. Replace the previous surface with the latest one available
3. Create a new mesh based on this surface
4. Map the solution data from the previous mesh to the new one
5. Simulate one time step
6. Extract the force moment data for each armour unit and write the information to a text file

3.5 Inter-code communication

The final step in automating the process was to develop a program that could manage the interaction of the CFD software and PhysX middleware. Both the Microsoft Windows operating system and the Linux operating system were available since the PhysX middleware ran on a machine operated by Microsoft Windows and Star-CCM+ was installed on another machine using Linux. Linux provides the option of using "shell scripts" which are programs consisting of a set of operating system

commands. These programs may include useful programming structures such as "if-then-else" and "while-do" loops and a decision was taken to perform the operating system control of the process on the Linux side.

The shell script was executed by entering a command in a terminal window. The name of the Star-CCM+ simulation file and the desired number of time steps were specified as part of this command.

The shell script, contained in a file called “h00005.sh”, performed the following tasks:

1. Check to confirm that the name of the Star-CCM+ simulation file and the desired number of time steps were specified and prompt the user to supply these if they were omitted
2. Place a “1” in the file “simulating.txt” to indicate that the simulation is in progress; a “1” in the file “ccm_execute.txt” to allow Star-CCM+ to simulate the first time step; and a “0” in the file “physx_execute.txt” to ensure that PhysX does not execute
3. Poll the file ccm_execute.txt to confirm that Star-CCM+ should execute
4. If not, wait 5 seconds and poll the file again
5. If so, run Star-CCM+ in batch mode, specifying that the macro “j00014.java” must be used
6. Run Star-CCM+ in batch mode again, specifying that the macro “j00015.java” must be used
7. Wait for Star-CCM+ to finish the simulation of a single time step
8. Place a “0” in the file “ccm_execute.txt” to prevent Star-CCM+ from executing and a “1” in the file “physx_execute.txt” to allow PhysX to calculate positions and orientations for the following time step
9. If the specified number of time steps have been simulated, terminate the program after placing a “0” in the file “simulating.txt”

The first two tasks are performed only once and the program loops through tasks 3 to 8 until the specified number of time steps have been simulated. After completion of all the simulations, task 9 is performed and the program then terminates.

3.6 Qualitative evaluation

Two simple test cases were devised to determine if all the elements of the algorithm worked as desired. The first test case involved two cubes placed apart on a flat surface in the path of a collapsing wall of water. The second test case was similar to the first but involved six stacked cubes.

Figure 3-4 shows the two-cube model 0.05 seconds after the start of the simulation. The mesh is visible as small squares coloured mostly in blue or red. The colours indicate the volume fraction: control volumes containing only water are shown in red and the blue control volumes contain only air. A small number of control volumes have a different colour because they contain a mixture of the two fluids. This indicates deformation of the original flat surfaces of the block of water due to gravity. The white block represents one of the cubes with the second one hidden from view behind the first. The geometry was symmetric around a vertical plane running through the middle of the flow domain along its length, but the mesh contained minor asymmetries. Appendix C contains additional technical information of the CFD model.

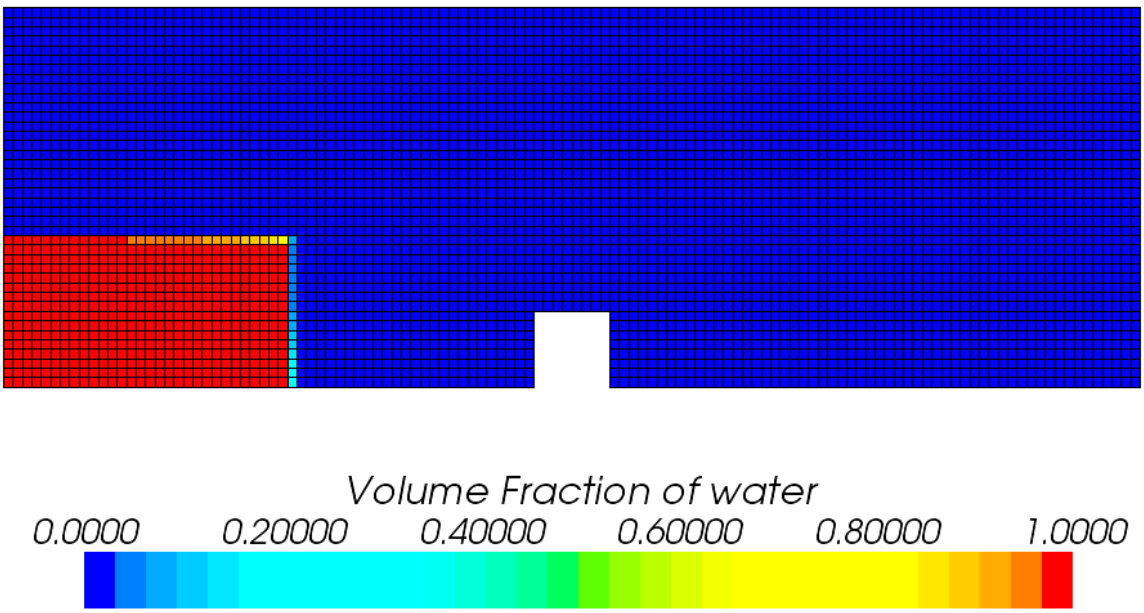


Figure 3-4 Two-cube model (side view)

A set of figures was compiled by storing a snapshot of the solution after each time step and it was used to produce a video clip. Figure 3-5 shows four of the snapshots from the set. The video clip revealed that the cubes remained at rest until they came in contact with the water, after which they moved with the water towards the downstream end of the flow domain. During their travel they appeared to rotate, translate and interact with the domain boundaries in a realistic way. Hardcopies of this document include a digital video disc (DVD) attached to the inside of the back cover page. The above-mentioned video clip is stored on the disc as “Video clip 01.avi”.

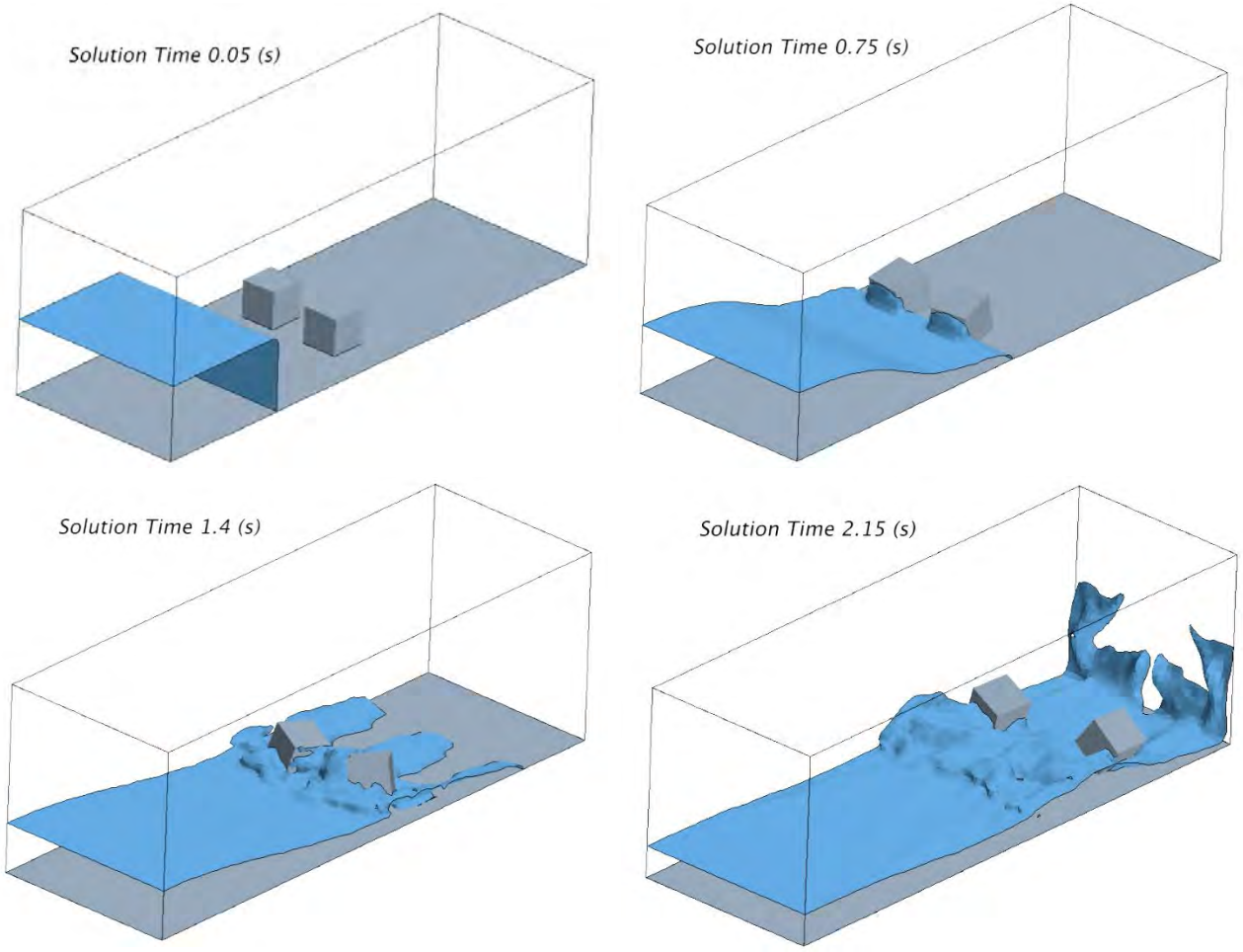


Figure 3-5 Two-cube model (bird’s-eye view)

Differences were observed between the movement of the blocks and these were ascribed to differences between the forces and moments acting on each block. The extent of the force and moment differences was revealed by plotting them on graphs during a simulation where the blocks

were not allowed to move. Figure 3-6 shows the force in the positive x, or downstream, direction as an example. In theory, forces and moments obtained from a CFD model based on a symmetric geometry with symmetric boundary and initial conditions are expected to also be symmetric.

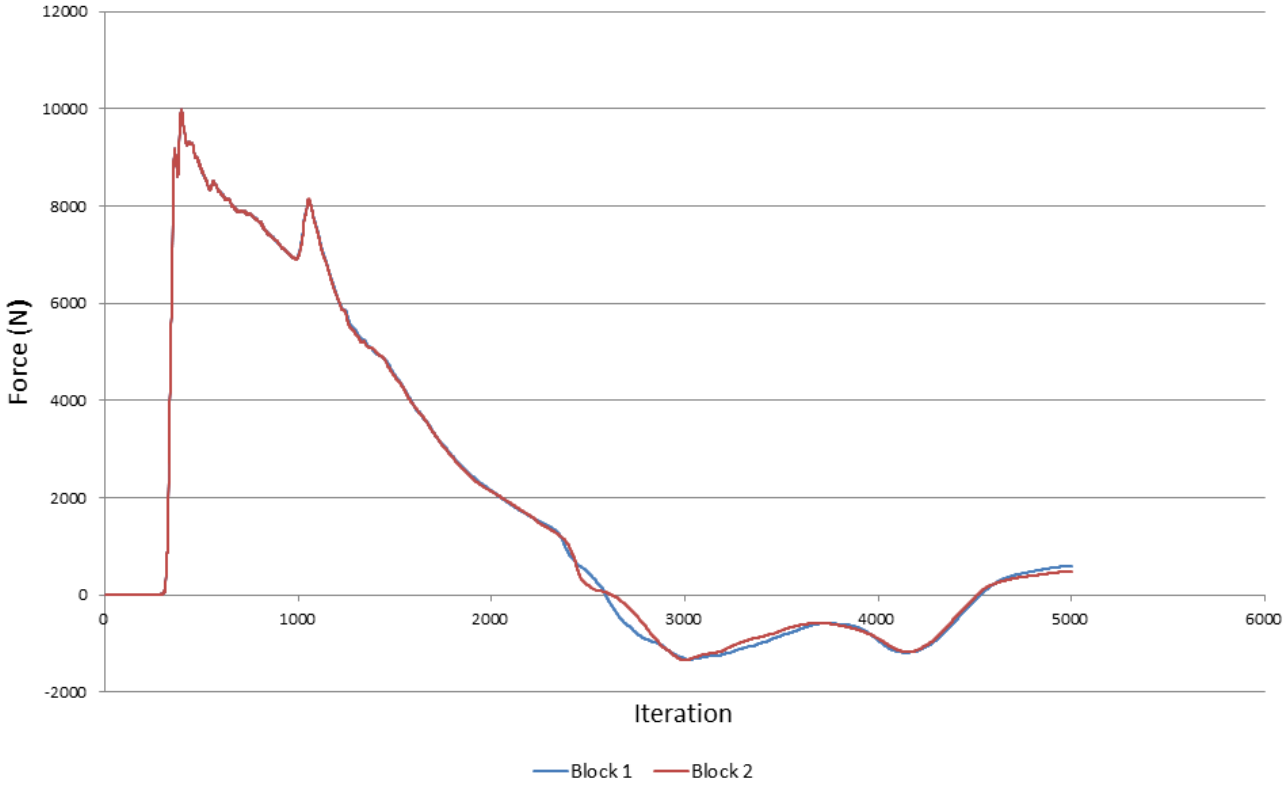


Figure 3-6 Difference in downstream force on stationary blocks in symmetric model

Bodies of revolution placed in a flow field at an ever increasing angle of attack may display a number of characteristic vortex flow patterns. At relatively low angles of attack no detectible vortex pattern will be present, but as the angle is increased, symmetric steady vortices will form followed by steady, asymmetric vortices and then unsteady asymmetric vortex shedding, also known as a von Kármán vortex sheet (Cummings, et al., 2003). The asymmetric steady vortices are believed to be the result of micro asymmetries in physical models. This implies that such vortices cannot be modelled with CFD unless some measure of asymmetry is also present in the CFD model. Mesh asymmetry was found to be a factor sufficient to introduce significant asymmetry in the flow solution (Liu, et al., 2007).

It is proposed that the relatively small differences observed in the forces and moments during the two-block simulation have their origin in the minor asymmetries detected in the mesh. It is further proposed that such asymmetric behaviour is desirable since it produces a more realistic result. As with bodies of revolution at an angle of attack, micro asymmetries will be impossible to avoid in a physical two-block experiment, making a perfectly symmetric result unobtainable in practice.

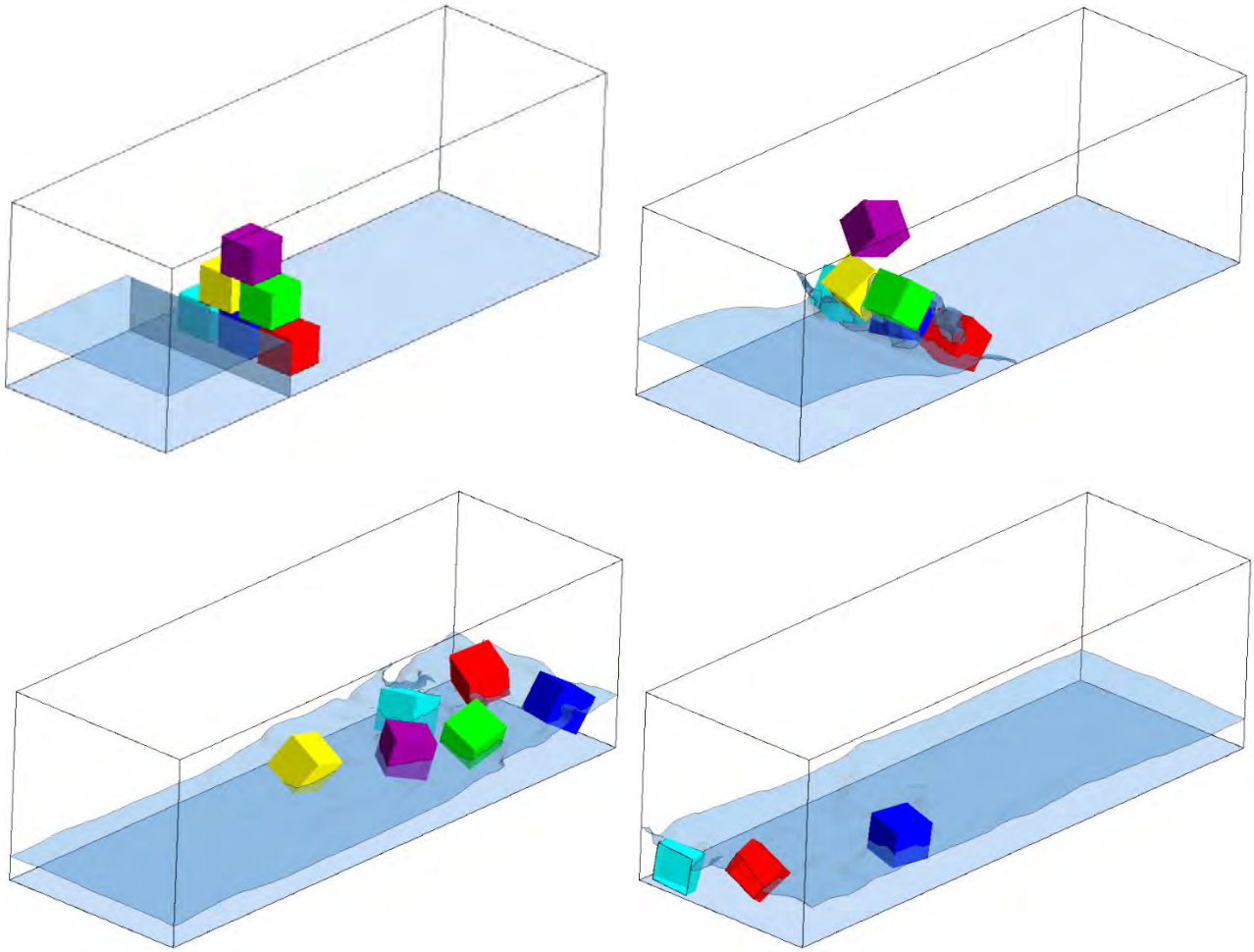


Figure 3-7 Six-cube model (bird's-eye view)

The second test case was similar to the first except for the number and placement of the blocks as well as the initial shape of the water column. Instead of two, there were six cubes stacked in a triangular pattern. A set of snapshots were again used to produce a video clip ("Video clip 02.avi" on the DVD). Figure 3-7 shows four of the snapshots from the set. The presence of six blocks and the

simulation of a longer time period allowed the blocks to interact with each other. The densities of the blocks were 60% lower than that of the water and they appeared to float, as expected. The blocks were allowed to leave the domain when reaching the upstream boundary. All behaviour observed in the video clip appeared realistic.

A breakwater consisting of 343 dolos armour units was simulated. The armour units were placed randomly on an inclined surface, as shown in Figure 3-8. A side view is shown in Figure 3-9 with a cross-section of the mesh in pink. The simulation did not indicate any significant movement of the armour units and none was expected, due to the weight of the armour units and the relatively moderate waves specified.

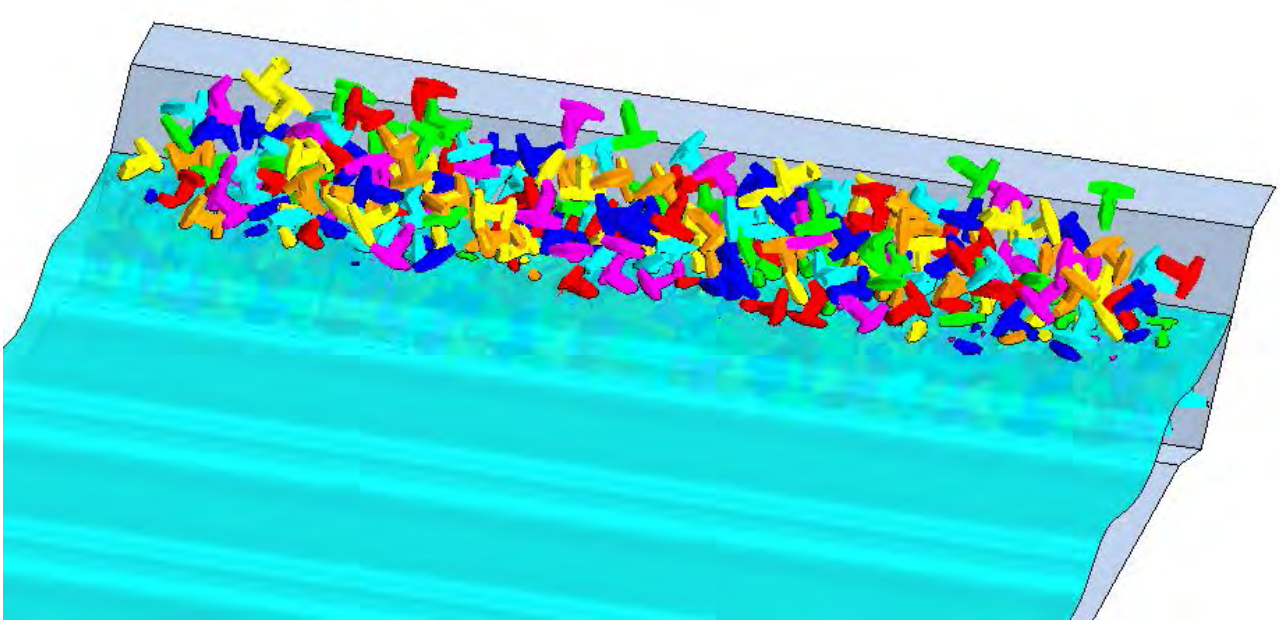


Figure 3-8 Armour units placed randomly on inclined surface (bird's-eye view)

3.7 Conclusion

An algorithm suitable for modelling a large number of armour units interacting with waves was developed. This was done by managing two stand-alone codes in a way that allowed them to work in unison with minimal user involvement. Two simple test cases demonstrated that each element of the

algorithm performed its role effectively. The simulations were stable and produced results that appeared qualitatively realistic. The algorithm was therefore regarded as properly validated and paved the way for its verification through quantitative assessment.

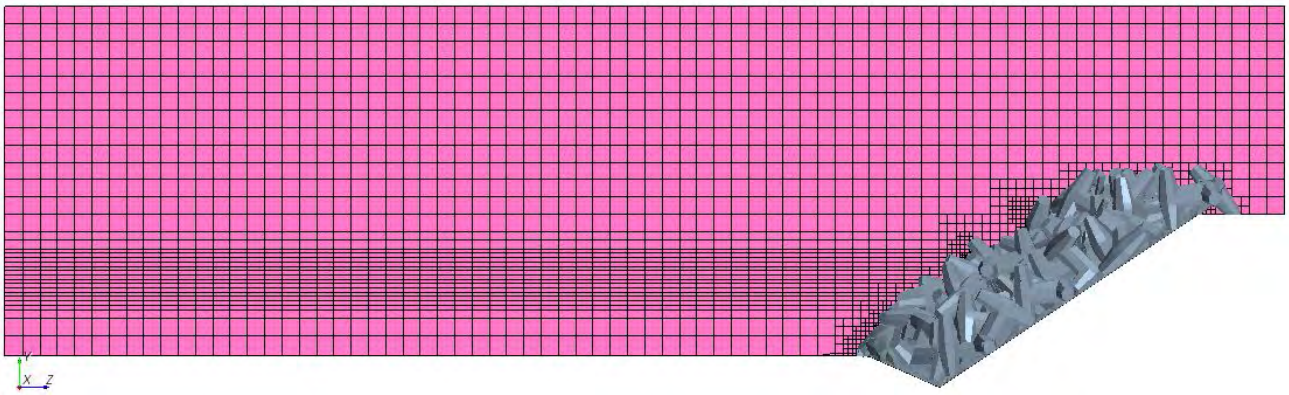


Figure 3-9 Armour units placed randomly on inclined surface (side view)

4 Numerical verification

4.1 Introduction

Chapter 3 described the successful validation of the proposed simulation algorithm. A CFD simulation of a boat hull encountering head waves was used to verify the method numerically. The initial simulation used a standard method to model the rotation and vertical translation of the hull. The CFD simulation was then repeated using the proposed simulation algorithm and the results were compared.

4.2 Boat floating in head waves (Star-CCM+ tutorial)

Star-CCM+ is a general-purpose CFD code and is therefore required to handle a wide range of flow scenarios including those that involve moving objects. The motion models in Star-CCM+ are divided into three broad categories (CD-adapco, 2012).

The first category includes methods that displace the mesh during a simulation. This is done by changing the positions of the vertices defining the mesh and may be used only during a transient analysis.

The second category contains methods that simulate motion during a steady-state analysis by using a moving frame of reference. An unsteady phenomenon is therefore modelled with a static mesh. Both rigid rotation and/or translation can be modelled with this method.

The third category focuses on the modelling of blade flutter and includes only the harmonic balance method. The harmonic balance method is used to solve non-linear differential equations with respect to frequency (as opposed to time). It assumes that the solution can be represented by a linear combination of sinusoidal waves.

The Star-CCM+ user manual includes a chapter with tutorials (CD-adapco, 2012). The tutorials assist users by providing stepwise explanations of how to use the software. One such tutorial simulates the motion of a boat facing head waves to demonstrate the use of motion models from the first category. It uses a combination of the 6DOF and multiphase solvers but does not require an overset mesh. Motion is simulated by displacing the whole mesh relative to a fixed frame of reference.

The literature survey yielded a number of studies containing analyses similar to the one in the above-mentioned tutorial. Unfortunately, none contained all the information required to replicate their simulations. A list of essential inputs would include – but is not limited to – the geometry of the objects; their mass, moments of inertia and initial positions; as well as the wave properties. The Star-CCM+ tutorial contains all the required information, and the software is considered a market leader (Progressive Media Group Limited, 2013). It was therefore decided to use the tutorial as the benchmark against which to test the accuracy of the proposed simulation algorithm.

The technical data of the tutorial are shown in Appendix G. The main results are shown in Figure 4-1 (vertical translation) and Figure 4-2 (rotation) and Figure 4-3 shows the hull geometry. Movement of the hull was constrained: only rotation around the y-axis and translation in the z-direction were allowed.

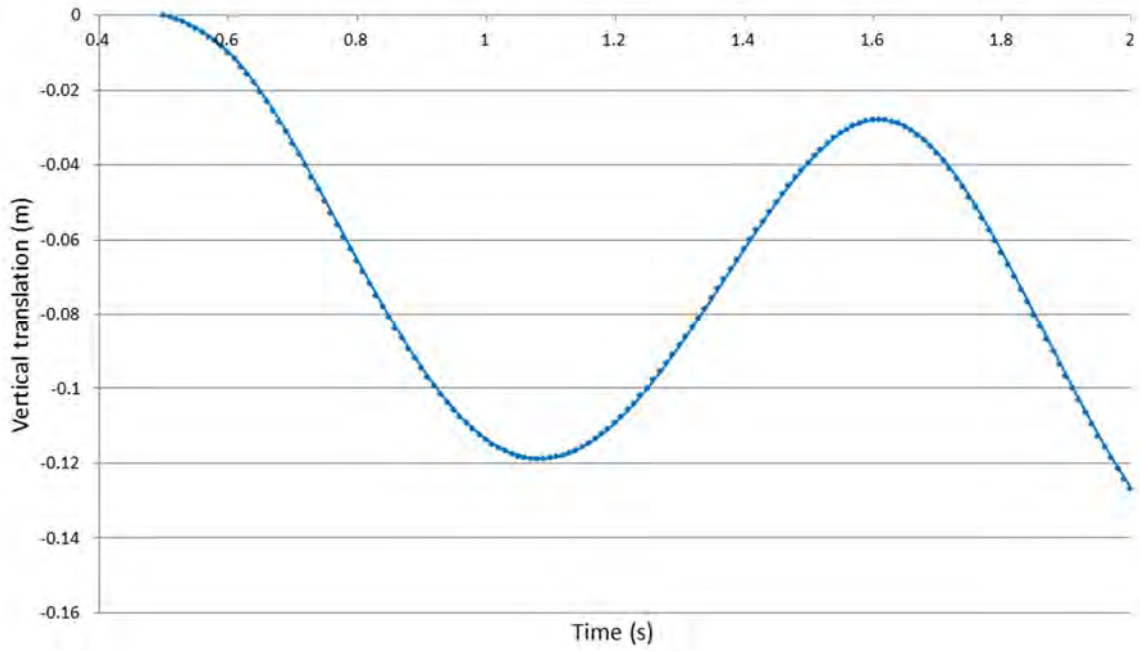


Figure 4-1 Vertical translation along z-axis (Star-CCM+ tutorial)

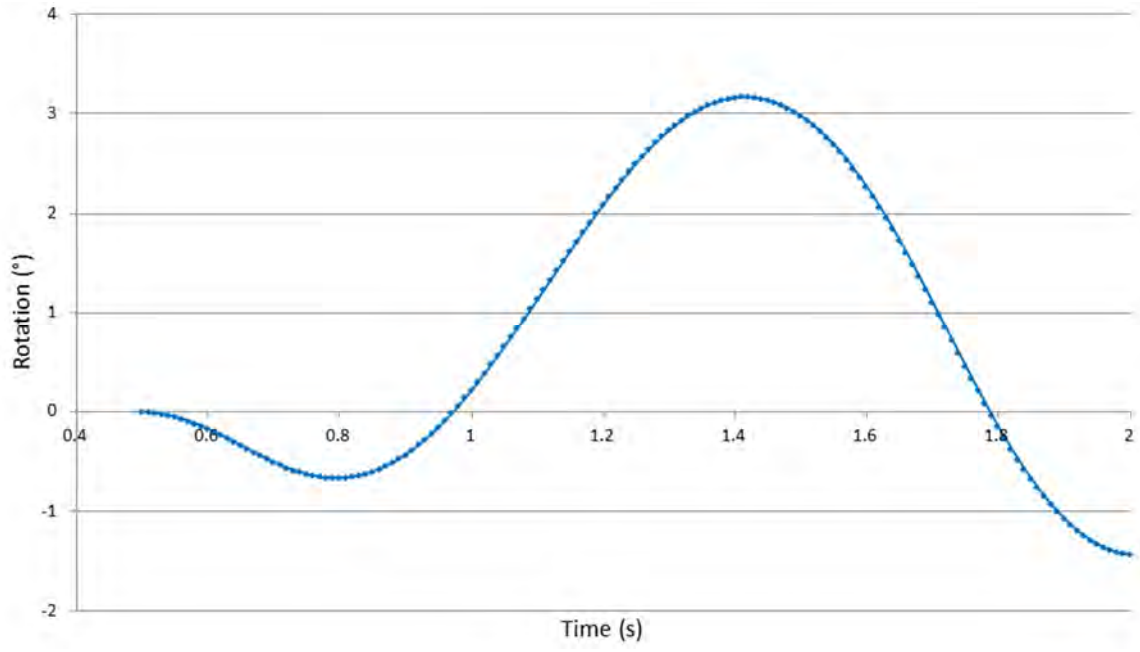


Figure 4-2 Rotation around y-axis (Star-CCM+ tutorial)

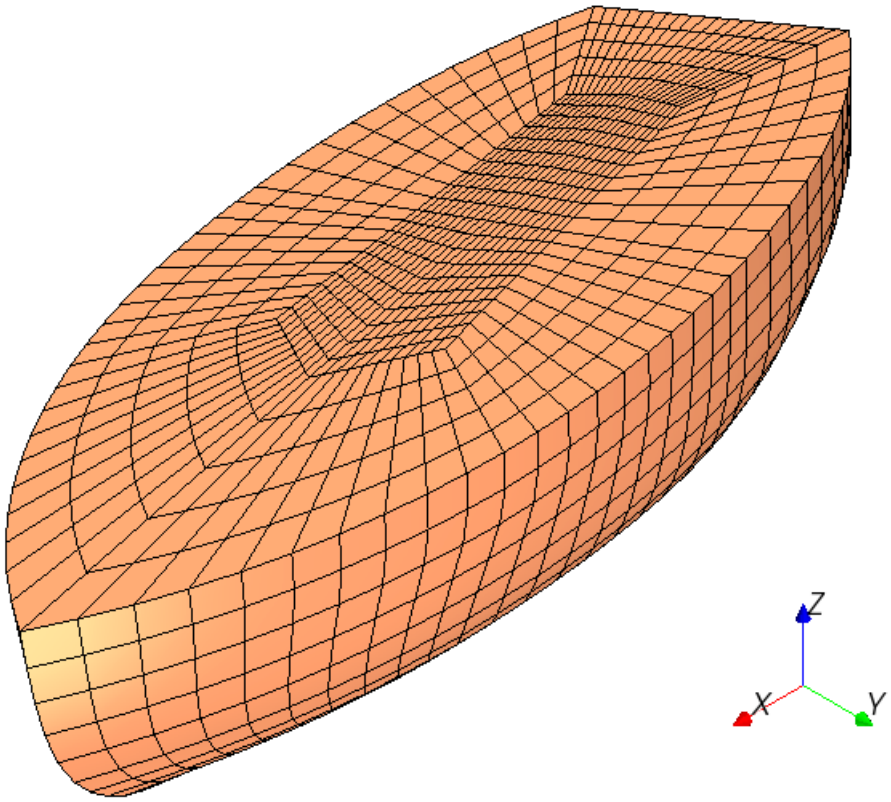


Figure 4-3 Hull geometry (Star-CCM+)

Various types of reports are available to extract data from any Star-CCM+ simulation (CD-adapco, 2012). The "Fluid Force Report" is one of the standard types and yields the total force acting on a set of faces, as shown below:

$$F_{ff} = \sum_{f} \left(F_f^{pressure} + F_f^{shear} \right) \cdot n_f \tag{7}$$

The set of faces includes every face that belongs to a group of surfaces chosen by the user. The user also specifies the direction in which the force is computed by providing the direction vector (n_f). The pressure and shear force vectors ($F_f{}^{pressure}$ and $F_f{}^{shear}$, respectively) acting on a given face (f) are added before the dot product is taken with the direction vector, yielding the force on that face in the desired direction. All the forces calculated in this way are summed to give the total fluid force (F_{ff}).

For example, the force exerted on the hull in the positive x-direction is obtained by selecting all the surfaces that comprise the hull and choosing the direction vector as (1, 0, 0).

Additional types of reports become available if the 6DOF motion model is employed in a simulation. One of the additional reports is called a "Rigid Body Force Report" and it is calculated as follows:

$$F_{rbf} = \sum_{f} \left(F_g + F_{ff} + F_{external}\right) f_r \tag{8}$$

The sum of three forces acting on the rigid body is first calculated. These three forces are the gravitational force (F_g), the fluid force (F_{ff}), and any external force specified by the user ($F_{external}$). The result is multiplied by a ramp factor (f_r) to yield the rigid body force (F_{rbf}). The user may exclude the gravitational force term from the calculation.

The ramp factor is defined as follows:

$$f_r = \begin{cases} 0 & : \quad t < t_s \\ \dfrac{t - t_s}{t_r} & : \quad t_s \leq t < t_s + t_r \quad and \quad t_r \neq 0 \\ 1 & : \quad t \geq t_s + t_r \end{cases} \tag{9}$$

The release time (t_s) is specified by the user and no motion is allowed before it is reached. The ramp time (t_r) is also specified by the user and allows for a gradual transfer of force to the rigid body.

If the gravitational force is excluded from equation (8) and the external force and ramp time are set to zero, equations (7) and (8) should yield the same result for any given direction vector, provided that the release time is exceeded.

A simulation was performed as prescribed in the tutorial using a mesh supplied with the Star-CCM+ software. The motion of the hull was determined by the dynamic fluid body interaction (DFBI) motion model in Star-CCM+, not by PhysX. The vertical fluid force component was compared with the vertical rigid body force component and Figure 4-4 shows that the results were not identical. They differed by a varying amount throughout the simulation: typically less than 10 N, but initially by as much as 56.8 N.

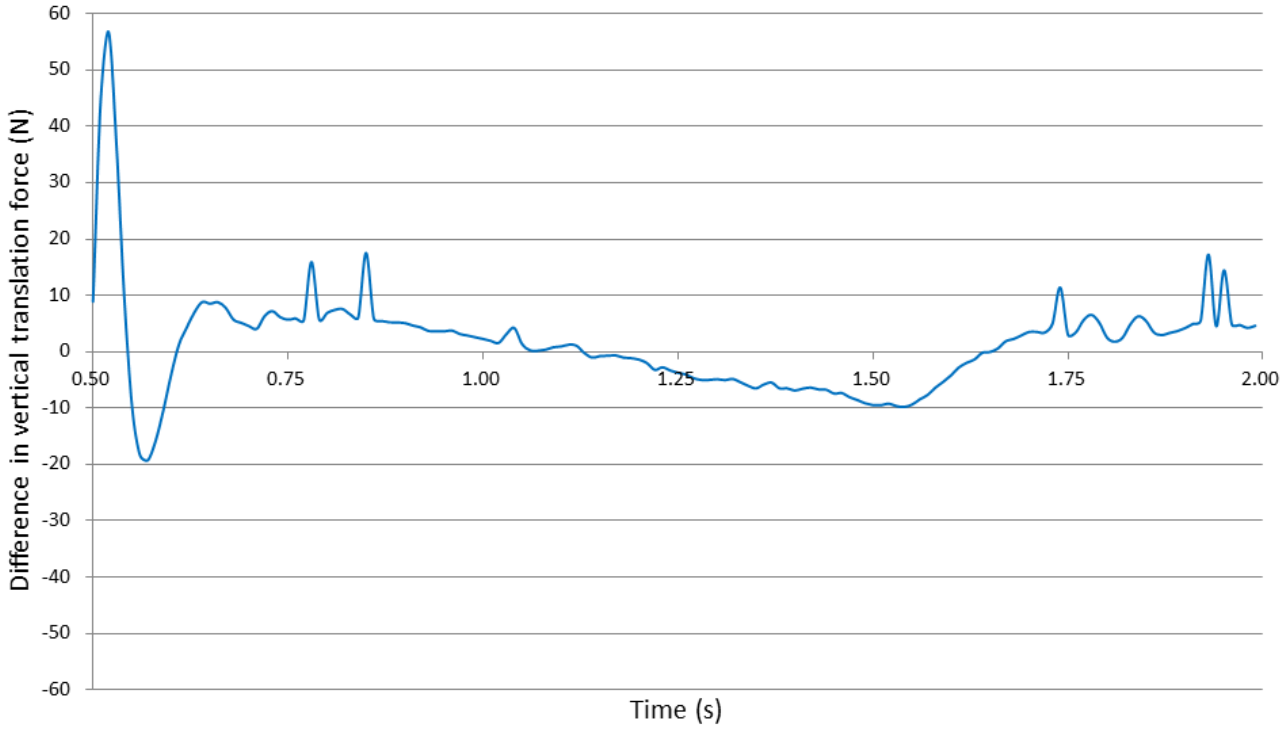


Figure 4-4 Difference between F_{ff} and F_{rbf} using five inner iterations

The differences appeared small compared to the magnitude of the forces, as shown in Figure 4-5 (5 inner iterations specified) and Figure 4-6 (100 inner iterations specified). They do however produce significant differences in the translation, as shown in Figure 4-9 (5 inner iterations specified) and Figure 4-10 (100 inner iterations specified). Since relatively small force differences could result in significant translation differences, it was decided to calculate the force necessary for the reported displacement from first principles.

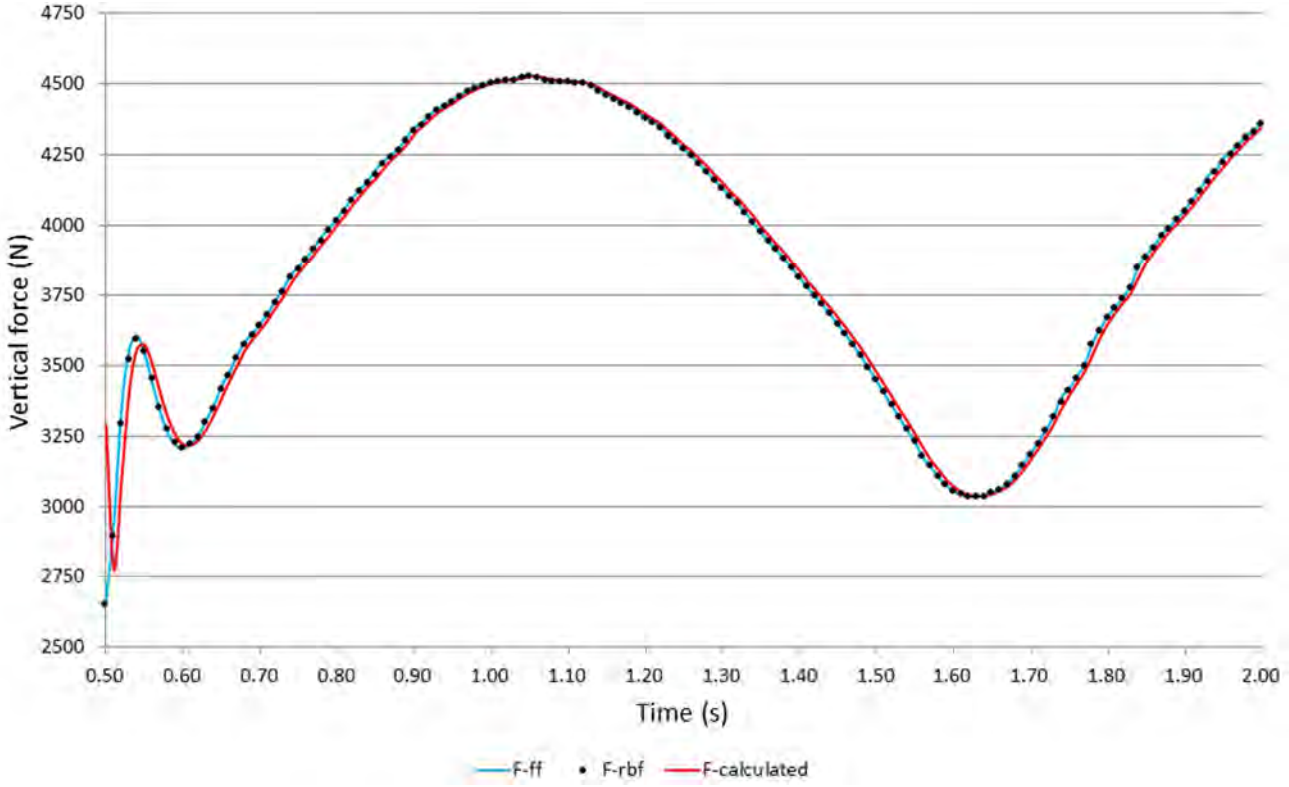


Figure 4-5 Vertical force (five inner iterations)

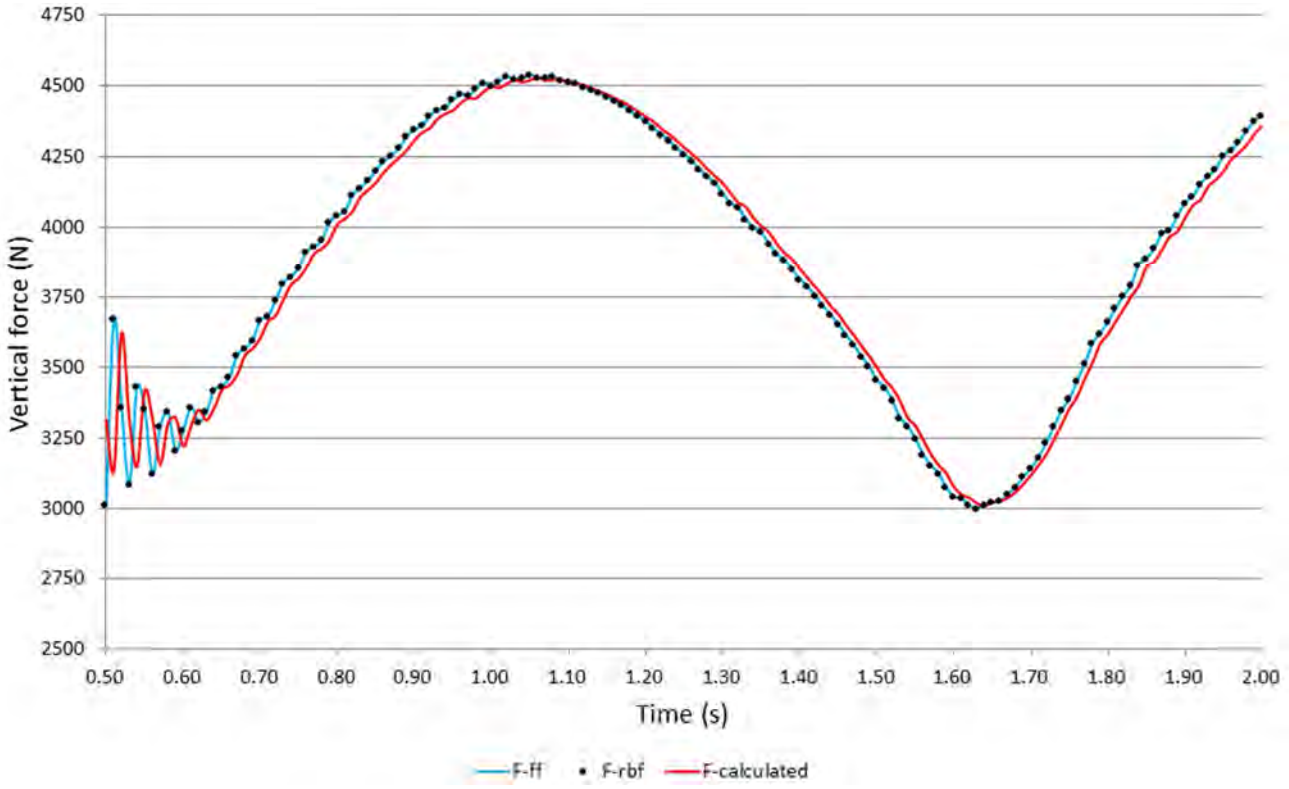


Figure 4-6 Vertical force (hundred inner iterations)

The vertical force will result in a vertical displacement of the hull after t_s has been reached. Since the vertical displacement of the hull was available as a function of time, the force required to produce the observed displacement could be determined using equations (10) to (12).

$$F = ma \tag{10}$$

$$v = a\Delta t + v_0 \tag{11}$$

$$x = \tfrac{1}{2}a\Delta t^2 + v_0\Delta t + x_0 \tag{12}$$

Constant acceleration (a) was assumed during any given time step (Δt). The displacement during the first time step after the removal of the constraints on movement was calculated with an initial velocity (v_0) and an initial displacement (x_0) equal to zero. Equation (11) was used to calculate the velocity (v) at the end of each time step and it was used as the initial velocity for the following time step. The force (F) required to produce the displacement (x) of the mass (m) was obtained with equation (10).

The forces calculated in this manner differed from both F_{ff} and F_{rbf}. Figure 4-7 shows the difference between F_{ff} and the calculated force ($F_{calculated}$) using the same scale as was used in Figure 4-4.

During a transient analysis the governing equations described in section 2.1 have to be solved at each time step. This requires a number of iterations, referred to as “inner iterations”. The inner iterations are similar to the iterations required during a steady-state analysis, but far fewer inner iterations are usually needed for a transient time-step because the transient equations are numerically more stable (Autodesk, 2013). An increase in the number of inner iterations from 5 to 100 significantly reduced the difference between F_{ff} and F_{rbf} . Figure 4-8 shows the results obtained with 100 inner iterations using the same scale as was used in Figure 4-4.

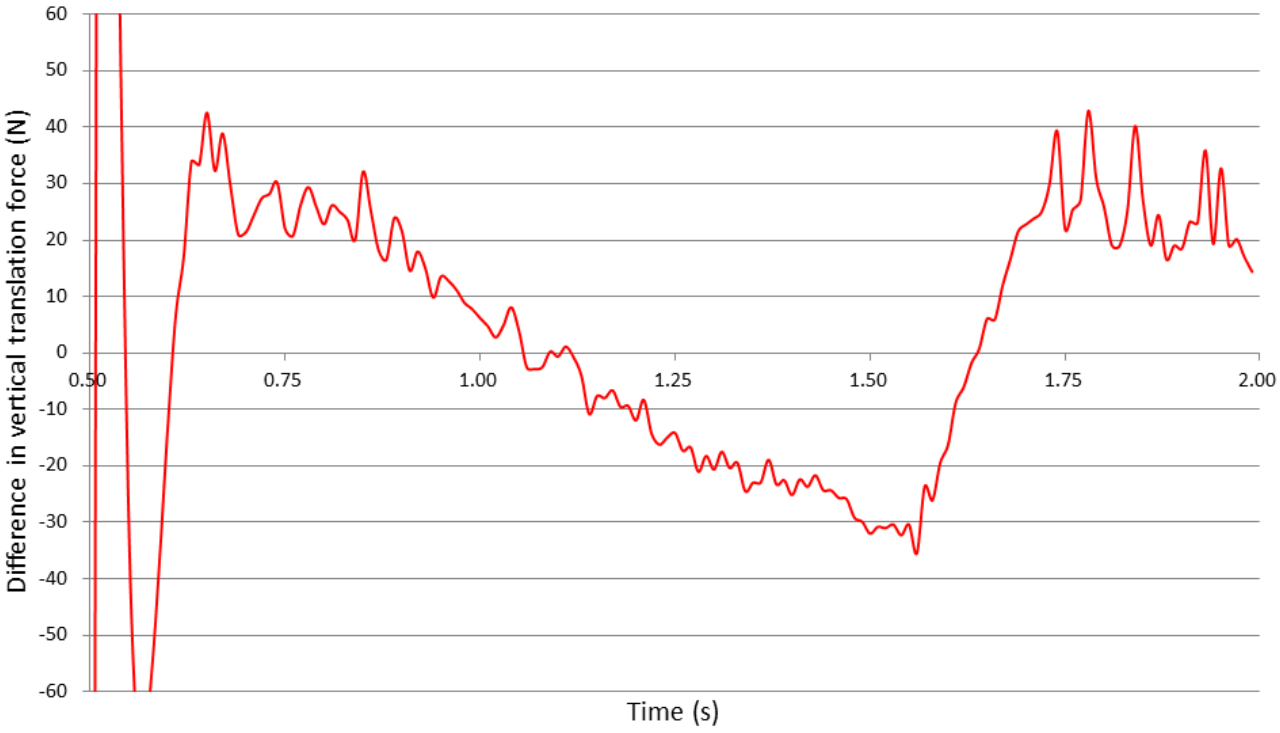


Figure 4-7 Difference between F_{ff} and $F_{calculated}$

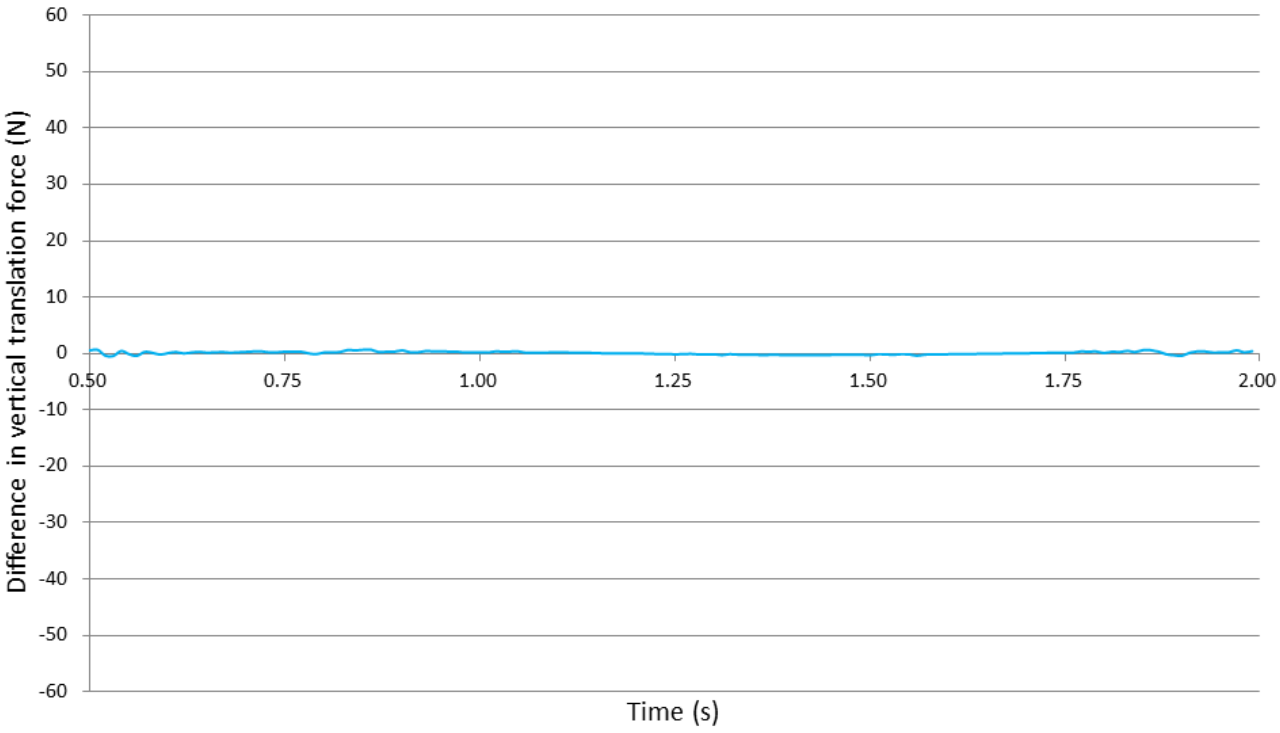


Figure 4-8 Difference between F_{ff} and F_{rbf} using a hundred inner iterations

The data obtained from the Star-CCM+ tutorial was used to test the response of PhysX to various inputs, prior to coupling it with the CFD solver. The behaviour of rigid bodies is governed by the force and moment equations, shown below.

$$\sum F = ma \tag{13}$$

F is the resultant force acting on the centre of mass of the rigid body, a is the acceleration and m is the mass. F and a are vectors (generally in 3D) while m is a scalar quantity.

$$\sum M = I\alpha \tag{14}$$

M is the moment about the centre of mass of the rigid body, α is the angular acceleration and I is the inertia of the rigid body. M and α are vectors (generally in 3D) and I is a tensor (generally a 3 by 3 tensor in 3D). Six equations are obtained which are solved simultaneously to yield the linear and angular acceleration. The accelerations are integrated numerically to obtain the velocities and are integrated again to give the position and pose of the rigid body.

When employing PhysX, the user is not required to perform these calculations, they are done automatically. The user only has to supply the forces and moments acting on the centre of mass of each rigid body. If these forces and moments are available at a different location, a calculation is required to determine values at the centre of mass.

PhysX was used in isolation to calculate the vertical displacement of the hull using F_{ff}, F_{rbf} and $F_{calculated}$ and the results are shown in Figure 4-10. The forces F_{ff} and F_{rbf} produced very similar results, but they differed significantly from the displacement reported by Star-CCM+. Three possible reasons for the discrepancies were considered:

1. The data reported by Star-CCM+ (F_{ff} and F_{rbf}) produce the reported displacement, but PhysX (in isolation) calculated the displacement incorrectly
2. F_{ff} and F_{rbf} do not produce the reported displacement, as correctly shown by PhysX
3. F_{ff}, F_{rbf} and the PhysX calculation were incorrect

The $F_{calculated}$ data were obtained from first principles using equations (10) to (12) along with the reported displacement, as explained previously. PhysX was able to reproduce the reported displacements with $F_{calculated}$ as input, indicating that the PhysX calculations were correct. The close correlation between the results showed the degree of accuracy that could be achieved with PhysX.

It is interesting to note the results obtained when using the data obtained from the simulations using 5 inner iterations, as shown in Figure 4-9. The displacement reported by Star-CCM+ correlates closely with the PhysX calculations using F_{rbf} and $F_{calculated}$ as input. This could have led to the premature conclusions that:

1. The small differences between F_{rbf} and $F_{calculated}$ were negligible and
2. The F_{ff} data produced an incorrect result

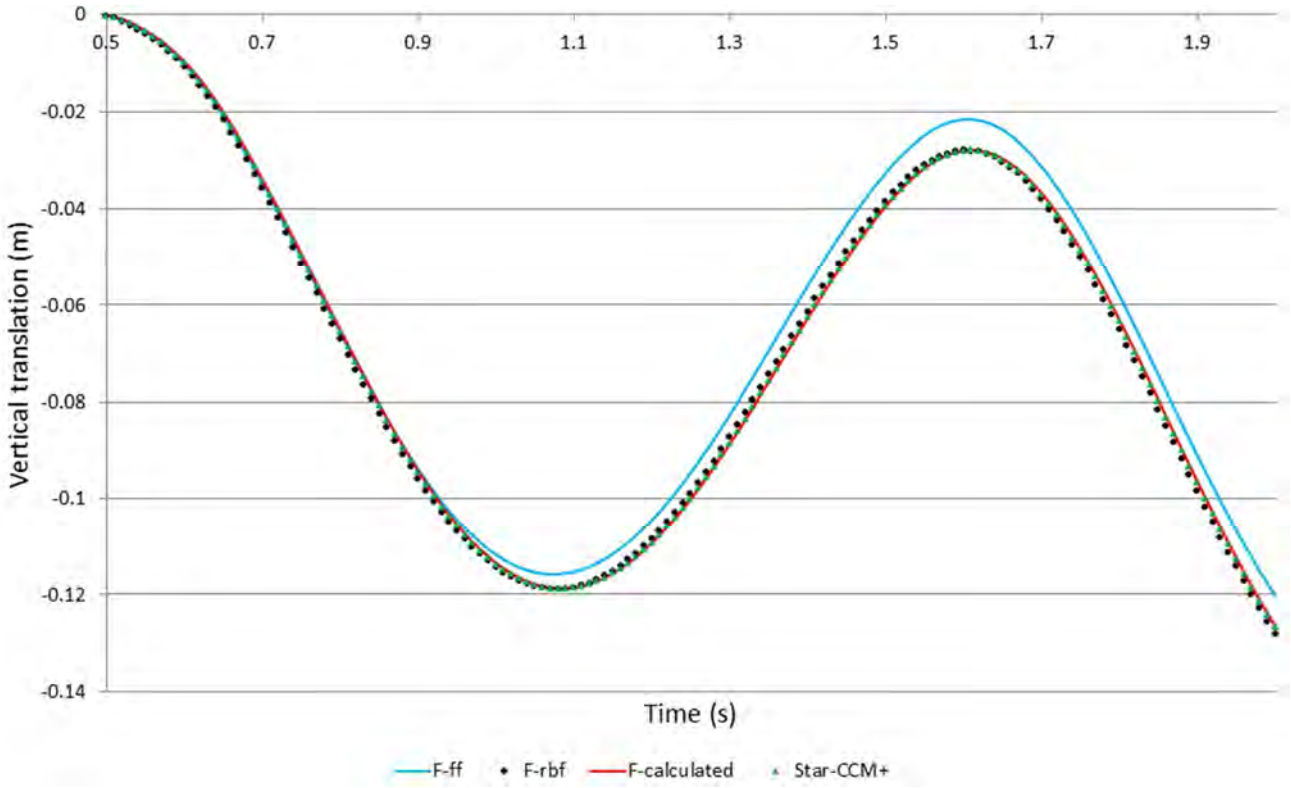


Figure 4-9 Results obtained using a five inner iterations

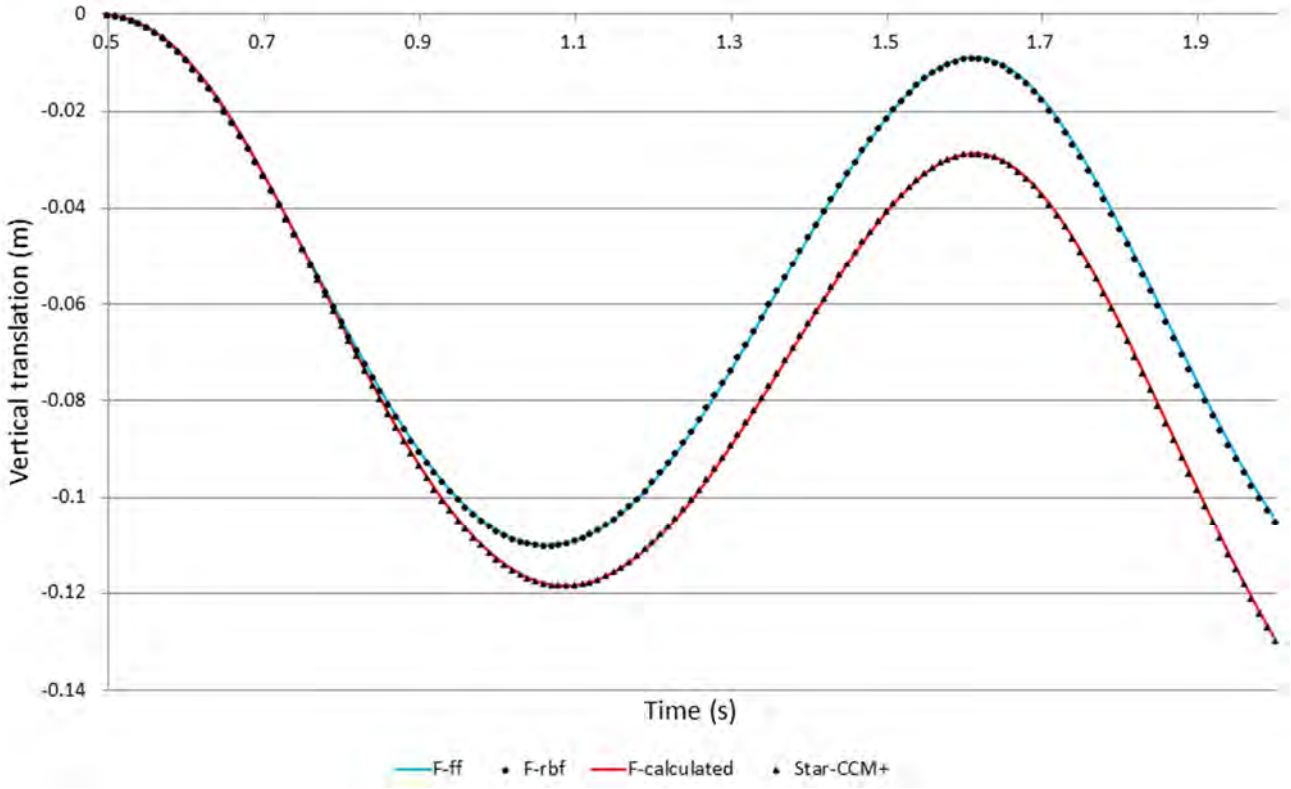


Figure 4-10 Results obtained using a hundred inner iterations

It was not possible to extract force data from the simulation that closely resembled $F_{calculated}$. The issue was raised with the developers of Star-CCM+ but unfortunately a timeous response was not received from the company's development-support department.

The moment around the y-axis of a coordinate system attached to the rigid body ($M_{y(rbm)}$) was available directly from the "Rigid Body Moment Report" or it could be calculated using equation (15).

$$M_{y(rbm)} = M_y + F_z c_x - F_x c_z \tag{15}$$

The terms on the right-hand side of equation (15) are all measured at the origin. M_y is the moment as measured at the origin; $F_z c_x$ is the product of the force component in the z-direction (F_z) and the

displacement of the rigid body in the x-direction (c_x); and $F_x c_z$ is the product of the force component in the x-direction (F_x) and the displacement of the rigid body in the z-direction (c_z).

In general, the values c_x and c_z have to be calculated by subtracting the position where the measurements are made from the position of the centre of mass. In this particular case, the boat is positioned such that the initial centre of mass lays at the origin, which means that the centre of mass of the boat is always given by the displacement of the boat from the origin. Since the measurements are always taken at the origin, the values of c_x and c_z reduce to the reported displacement of the boat in the x and z-direction, respectively. Furthermore, the motion of the boat is constrained in the x-direction, which means that the term $F_z c_x$ remains zero.

The rotation of the hull was calculated with PhysX (operated in isolation) using different moment values as input and the results are shown in Figure 4-11. The value of c_z could be taken either as the displacement calculated with F_{ff}, or as the value obtained from the Star-CCM+ tutorial. The legend "$M_{y(\text{calculated displacement})}$" in Figure 4-11 refers to calculations based on the former value and the legend "$M_{y(\text{reported displacement})}$" refers to calculations based on the latter. The results correlated well with the rotation reported by Star-CCM+, but none exactly matched it.

The rotation obtained using the moment from the $M_{y(rbm)}$ report and the rotation obtained using the moment calculated with equation (15), along with the displacement reported by Star-CCM+, correlated very well. The reason was that the two moments were practically identical (see Figure 4-12) which confirmed that equation (15) was correctly relating the moment measured at the origin to the moment at the centre of mass.

The mesh used in the tutorial was supplied with Star-CCM+ and could not be altered. Simulations involving PhysX required frequent regeneration of the mesh, and a mesh was constructed for this purpose. Figure 4-13 shows the mesh used in the tutorial and Figure 4-14 shows the mesh used with PhysX.

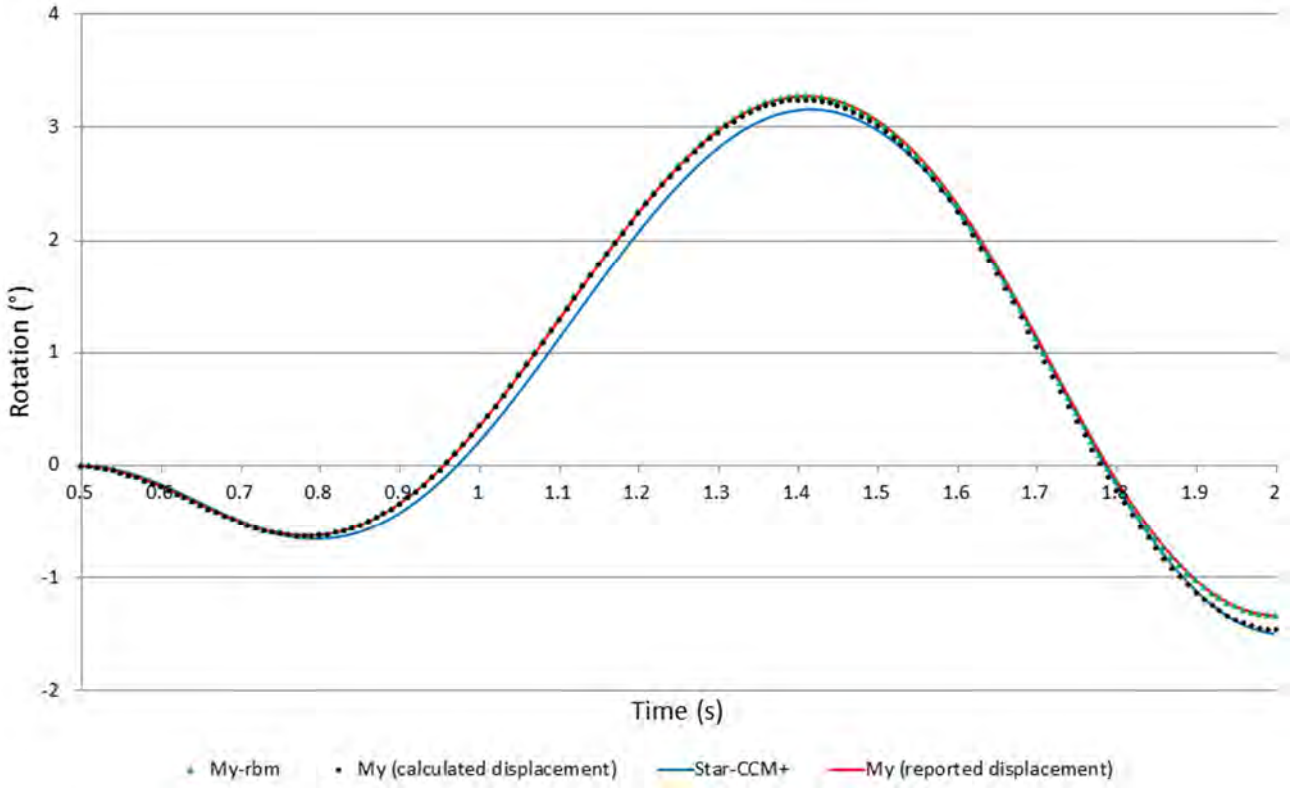


Figure 4-11 Rotation of the hull calculated with different moments as input

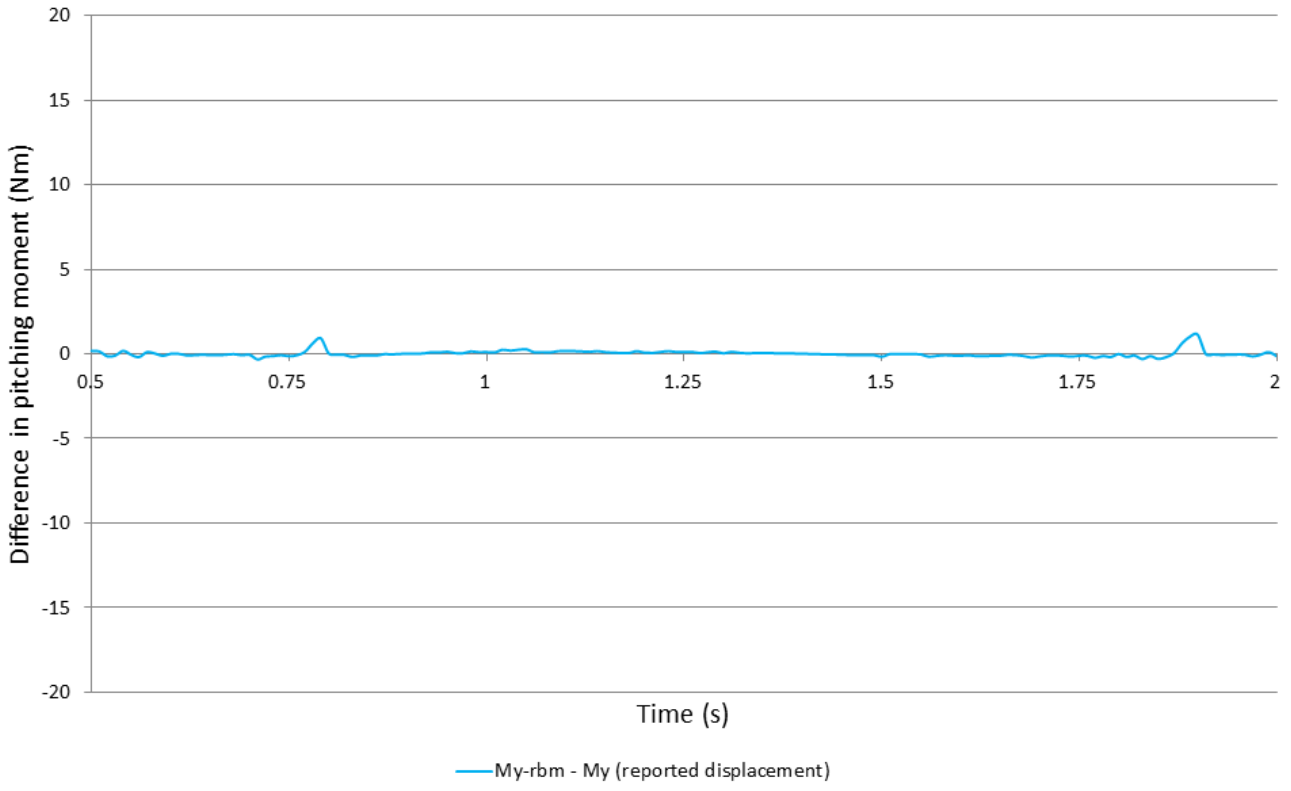


Figure 4-12 Difference between calculated and reported pitching moment

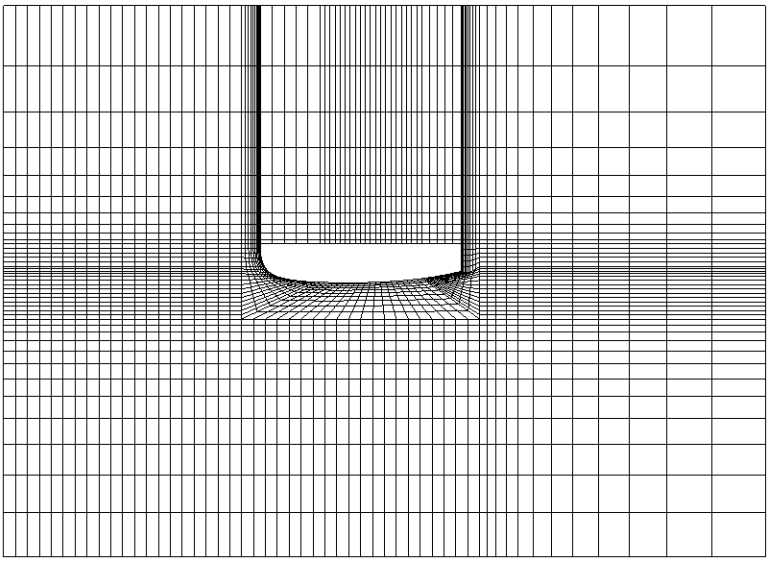


Figure 4-13 Mesh used in Star-CCM+ tutorial

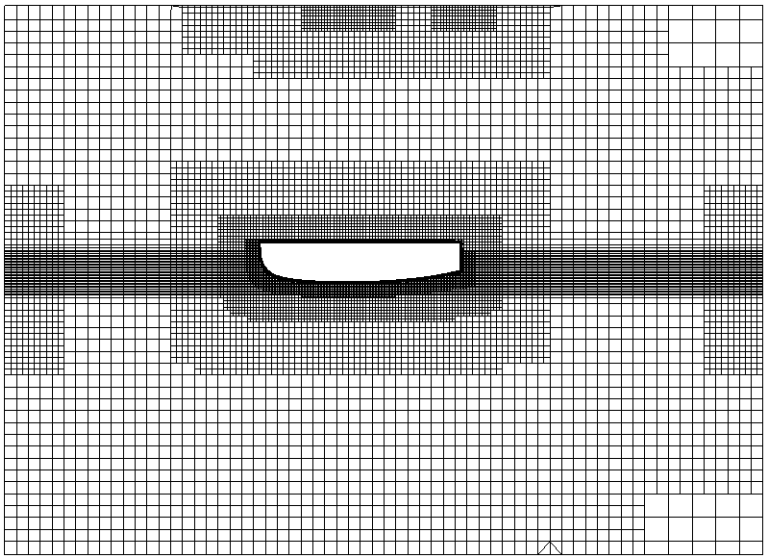


Figure 4-14 Mesh used with PhysX

The mesh used in the tutorial consisted of 50176 control volumes and the PhysX mesh consisted of 803902 control volumes. The simulation described in the tutorial was repeated using the finer mesh. The translation and rotation results obtained with the two meshes correlated well, as shown in Figure 4-15 (vertical translation) and Figure 4-16 (rotation). Please note that neither of these simulations

involved PhysX, although the finer mesh is referred to as the “PhysX mesh”. The name was chosen because it was later used in simulations that did make use of PhysX.

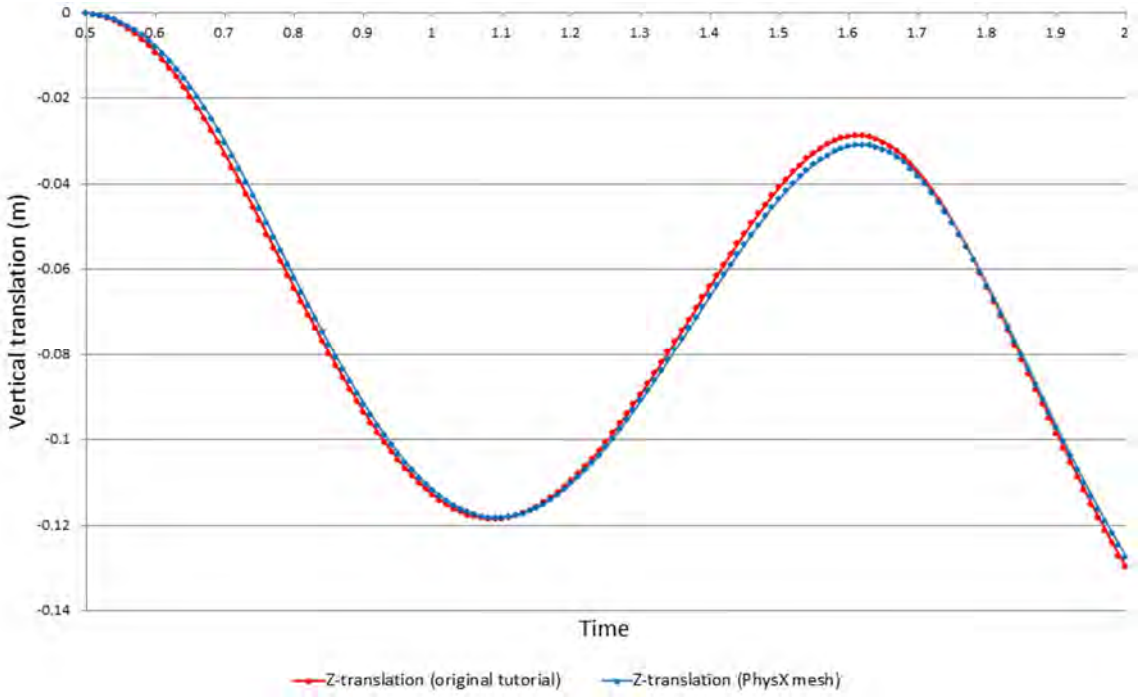


Figure 4-15 Vertical translation obtained from coarse mesh (red) and fine mesh (blue)

4.3 Boat floating in head waves (using Star-CCM+ and PhysX)

Patran is software developed by the MSC Software Corporation and is widely used in Finite Element Analysis (FEA) to provide solid modelling, meshing, analysis setup and post-processing capability for a number of FEA solvers (Zhang, et al., 2012). The hull geometry was exported from Star-CCM+ in Patran format. This was done to extract the coordinates of the vertices defining the surface of the hull. The data file also stated which four vertices belonged to each of the surface elements. This information was used to generate the PhysX model shown in Figure 4-17. The hull geometry in PhysX was identical to the hull geometry in Star-CCM+, except that it had double the surface elements. Each quadrilateral element of the Star-CCM+ geometry (Figure 4-3) was divided in half to form two triangular elements in the PhysX geometry.

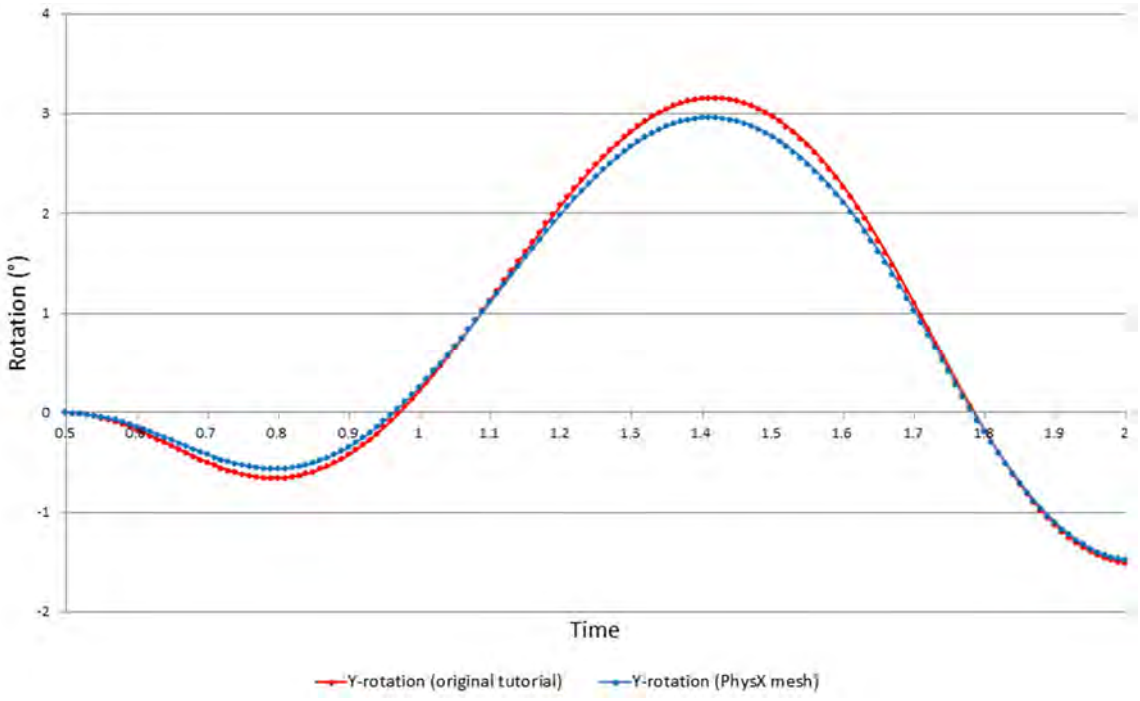


Figure 4-16 Rotation obtained from coarse mesh (red) and fine mesh (blue)

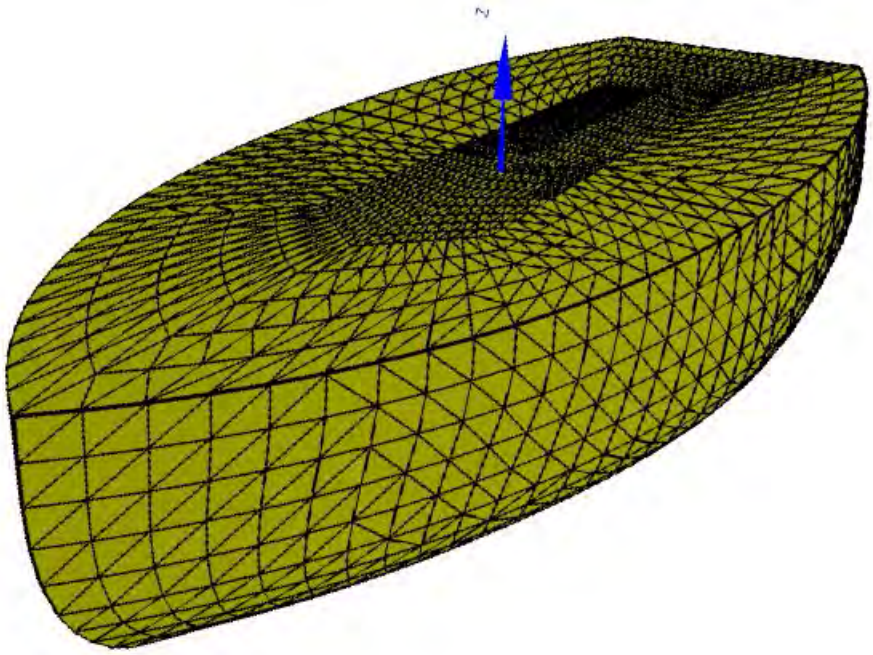


Figure 4-17 Hull geometry (PhysX)

The simulation described in the tutorial was repeated using PhysX to determine the position and orientation of the hull. A comparison of the force in the x-direction is shown in Figure 4-18; a comparison of the vertical force component is shown in Figure 4-19; and a comparison of the moment around the y-axis is shown in Figure 4-20.

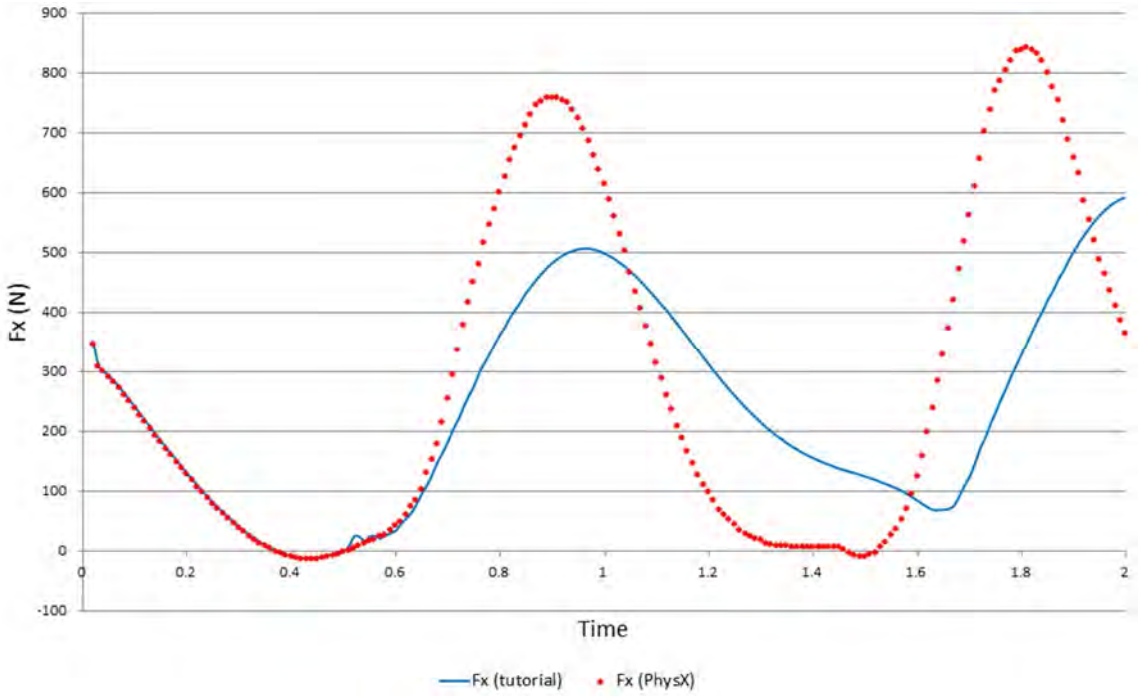


Figure 4-18 F_x obtained from tutorial (blue) and PhysX simulation (red)

The F_x, F_z and M_y graphs all showed an oscillating trend and the minima and maxima as well as the time at which they occurred differed significantly between the simulations. The tutorial data showed a sharp increase in F_z immediately after the boat was released at 0.5 s.

The reason for this sudden increase was not apparent and it did not appear to have a physical cause. It appeared as if the buoyancy force acting on the hull had suddenly increased. In order to keep any stationary object stationary, Newton's laws require that the resultant force acting on it should be equal to zero. If the effects of the surrounding air are neglected, there are three forces that balance prior to release:

- Weight (in direction of gravity – in this case in the negative z-direction)
- Buoyancy force (opposite direction as gravity – in this case in the positive z-direction)
- Balancing force (direction dependent on circumstances)

In practice, the balancing force may be from the hand of the person holding the object or from a structure keeping the object in place. In the simulation such a force is not directly applied, but implied by constraining the motion. At the time of release, the balancing force disappears instantly, and the resultant force jumps from zero to the difference between the weight and the buoyancy forces. The body then starts accelerating in the direction of the resultant force with the velocity and displacement increasing (smoothly) from zero.

The buoyancy force equals the weight of the fluid that is displaced and can change if a larger or smaller volume is immersed. This will typically happen as a partially submerged objects starts to move, but since the displacement changes smoothly, so does the buoyancy force. Alternatively, the buoyancy force will change if the shape of the free surface changes. In this case it happens due to the passing waves which have a relatively low frequency compared to the size of the time step and will therefore result in a smooth change.

A graph showing the resultant force on the object (not the buoyancy force) is expected to show a sharp change immediately after release and in this case the value should drop to below zero, since the positive z-direction was chosen upwards. The resultant force immediately after release is expected to be approximately 2700 N - 3924 N = -1224 N. This negative resultant force results in the initial downward movement seen in Figure 4-1.

The volume of water displaced by the hull was calculated for different levels of submersion. The buoyancy force was determined from the weight of the displaced water and the results are shown in Table 4-1. Although the still-water assumption was not valid at 0.5 s, the information was still useful for a rough (order-of-magnitude) estimate. It showed that the hull had to sink an estimated 40 mm in

0.02 s to account for the increase in vertical force component. The vertical translation data (Figure 4-1) showed no such movement.

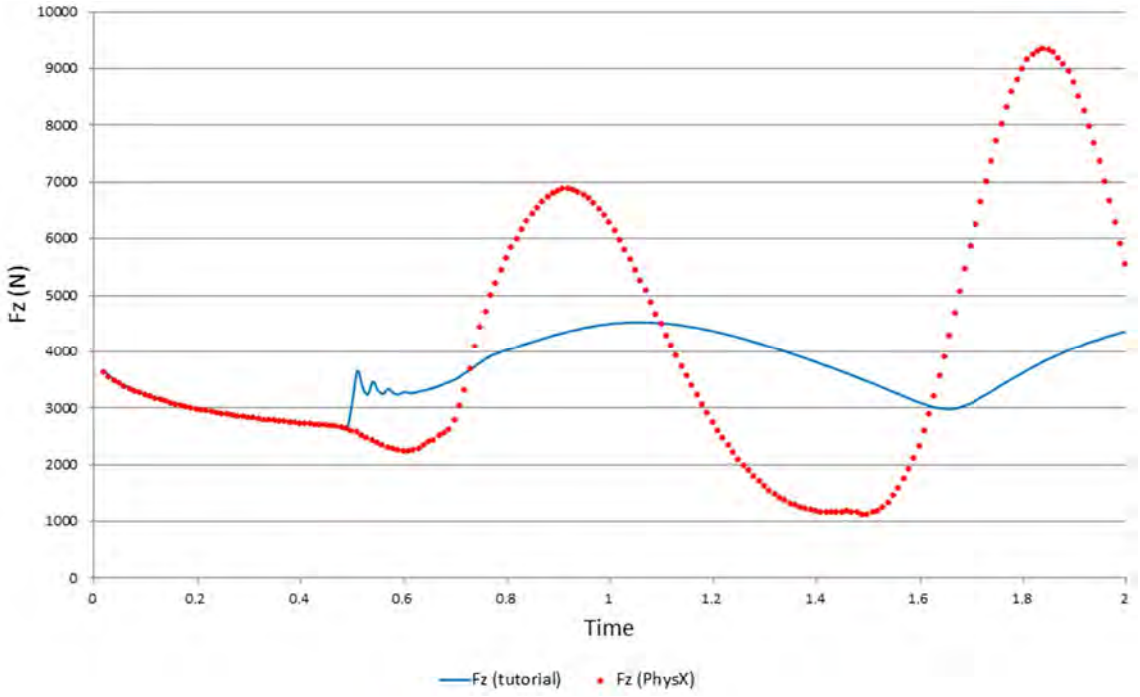


Figure 4-19 F_z obtained from tutorial (blue) and PhysX simulation (red)

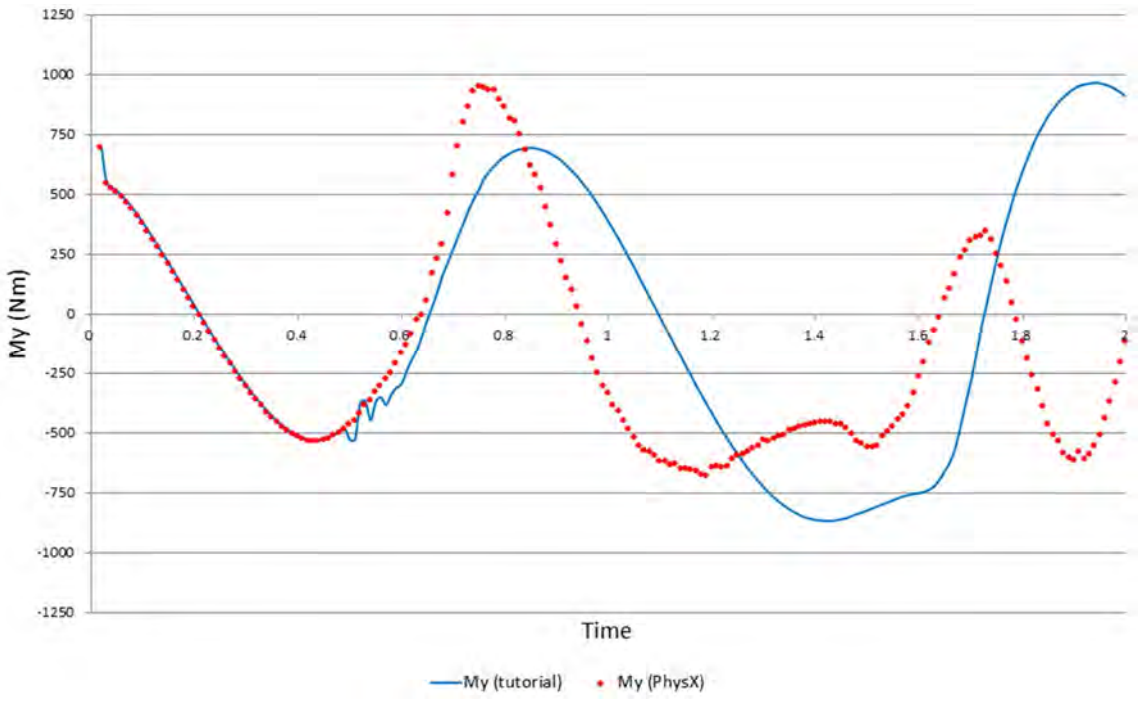


Figure 4-20 M_y obtained from tutorial (blue) and PhysX simulation (red)

Table 4-1 Buoyancy of hull submerged in still water

Z-coordinate of still water line (m)	Volume of displaced water (m^3)	Weight of displaced water (kN)
0.000	0.6088	5.96
0.100	1.078	10.5
0.200	1.587	15.5
0.300	2.121	20.8

The translation (Figure 4-21) and rotation (Figure 4-22) results reflected the significant differences in the vertical forces and pitching moments of the simulation using PhysX and the tutorial.

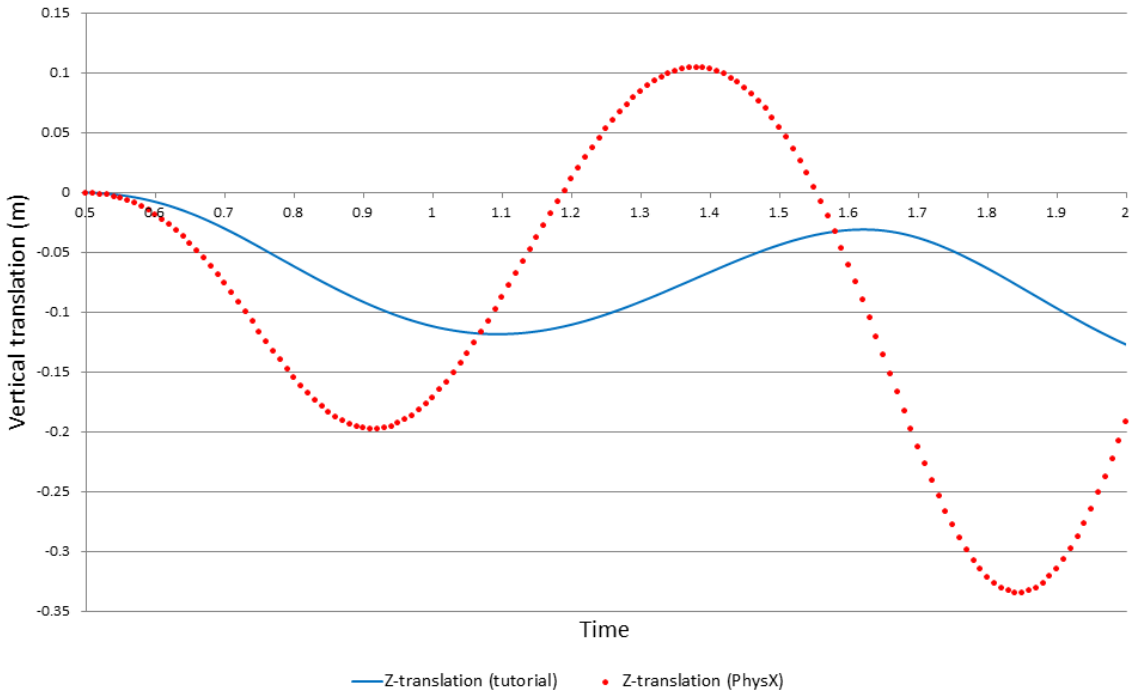


Figure 4-21 Vertical translation obtained from tutorial (blue) and PhysX simulation (red)

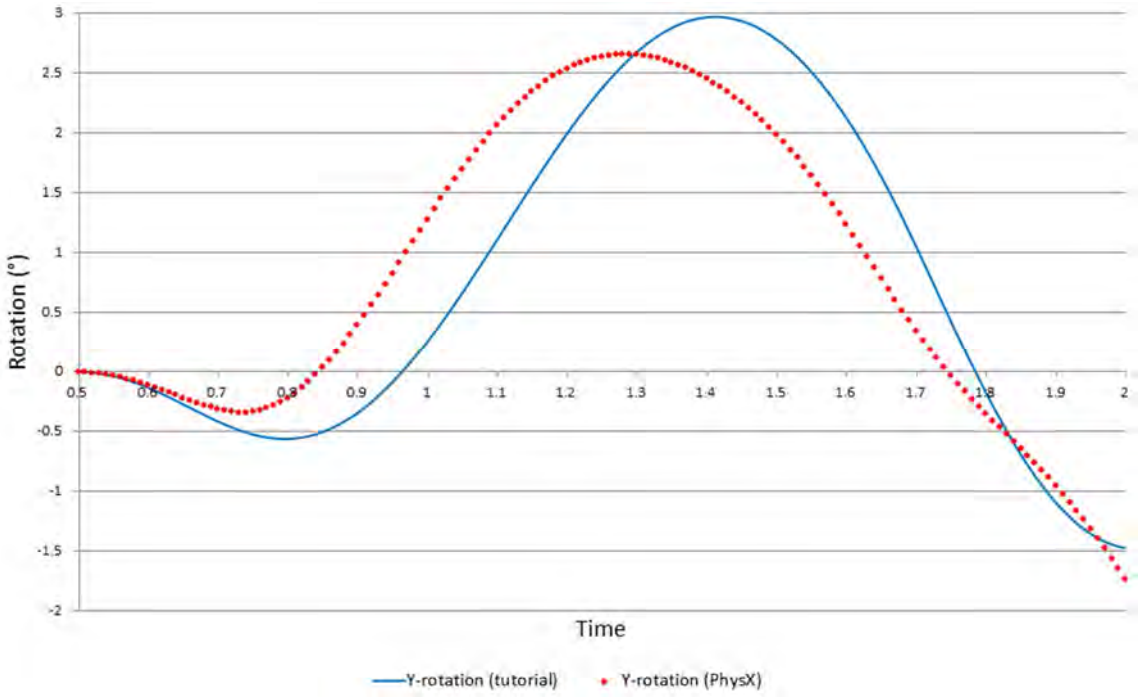


Figure 4-22 Rotation obtained from tutorial (blue) and PhysX simulation (red)

Translation and rotation data were taken from the Star-CCM+ tutorial and used in a simulation that did not use PhysX or a motion model. Instead, the motion during the first 0.3 s after release (0.5 s to 0.8 s) was prescribed such that the hull translated and rotated as in the tutorial. Figure 4-23 shows the vertical force component (F_z) obtained from the "prescribed motion" simulation superimposed on part of Figure 4-21. A significant offset was noticeable between the values of the "prescribed motion" simulation and the tutorial, which could not be explained since a close correlation was expected.

The "prescribed motion" simulation did not produce any sudden changes in F_z. The values initially correlated closely with the values obtained from the simulation using PhysX but later diverged because the motion of the hull differed in the two simulations.

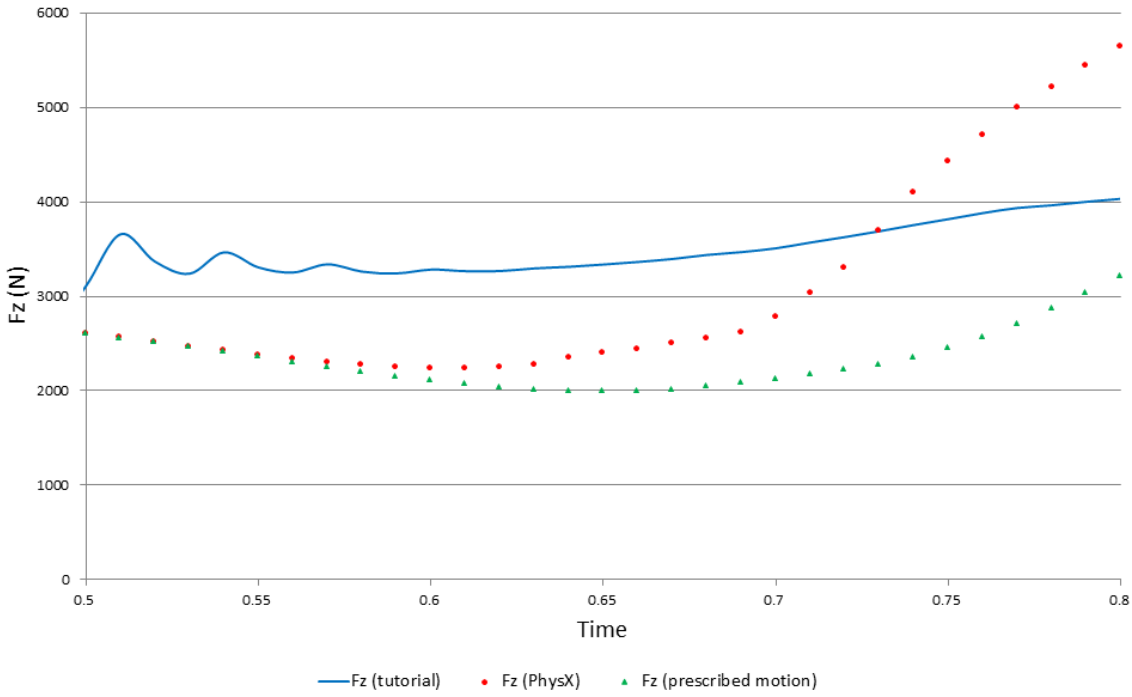


Figure 4-23 F_z obtained from tutorial (blue); PhysX simulation (red) and simulation with prescribed motion (green)

4.4 Discussion of results

The value of the numerical verification exercise was limited by a number of factors. The first was encountered with the original Star-CCM+ tutorial. A mismatch was observed between the force and moment data on the one hand and the translation and rotation data on the other. A number of possible causes were investigated but the reason for the mismatch remained unclear. The correct force profile could be extracted from the translation data using first principles, but such a profile could not be obtained from the simulation data. Since the proposed solution algorithm was dependent on the correct profile being available, the method could not be verified comprehensively.

In addition to the force–translation and moment–rotation mismatch, it was also not clear why the values of the "Fluid Force Report" (F_z) and "Rigid Body Force Report" (F_{rbf}) matched when using a hundred inner iterations, but not when using five. Based on the definition of the force reports in the

user manual, such a discrepancy should not exist regardless of the number of inner iterations specified.

A third important issue was the sudden changes in the values of the forces and moments reported immediately after the release time had been reached. The change was particularly significant in the case of the buoyancy force and could not be accounted for by any physical phenomenon. A dramatic change is expected in the resultant force, not the buoyancy force. In contrast, the simulation using PhysX did not show such rapid changes and appeared more realistic.

The user has the option to specify a ramp time, which allows for a gradual transfer of the fluid force to the rigid body. It was possible to use the ramp time to manipulate the buoyancy force by applying the resultant force (which expected to change instantly), gradually. Using a ramp time other than zero would have been of no benefit to the study since PhysX relies on F_z. The motion of the hull is determined by the value of F_{rbf} and unless the ramp time was zero, F_z would (per definition) not equal F_{rbf}.

The results of the "prescribed motion" simulation further confirmed the significant difference between the force data provided by Star-CCM+ in its reports and the data required as input for PhysX to ensure a close correlation of results. In spite of all the limiting factors, it was shown that the simulation algorithm produced the correct response if provided with the correct input, as shown in Figure 4-10.

4.5 Conclusion

A number of numerical simulations were performed to verify the accuracy of the proposed simulation algorithm. Some unexpected difficulties were encountered which could not be resolved. Indications were that the difficulties related to the numerical methods implemented in the CFD software, and the issues were raised with the developers. The ability of PhysX to provide an accurate response if provided with the correct force input was demonstrated which made the numerical verification exercise at least partially successful.

5 Experimental verification

5.1 Introduction

In addition to the numerical verification discussed in the previous chapter, a number of physical experiments were also conducted in the hydraulics laboratory of the CSIR. The experiments were conducted over a three-day period in one of the flow channels at the facility. A digital movie camera was used to film moving objects fitted with fiducial markers. Their positions were determined using image processing. The experimental work is discussed, as well as a numerical simulation of a representative experiment. The chapter concludes with a discussion of the main results.

5.2 Design of experiment

In practice, breakwater structures are expected to retain their structural integrity, but individual armour units may wobble or occasionally dislodge under severe environmental conditions (Latham, et al., 2008). The experiments were designed to simulate this process by allowing a solitary wave to collide with a number of rigid bodies. Momentum transfer from the wave to the rigid bodies would

result in significant changes in their positions and orientations if they were made of a relatively light material such as wood. During the experiments a number of the fiducial markers had to remain visible to be captured on video. It was assumed that this could be achieved by an appropriate selection of the diameters of fiducial markers, the dimensions of the rigid bodies and the amplitude of the solitary wave.

The hydraulics laboratory of the CSIR is situated in Stellenbosch. The facility offered the following experimental infrastructure: two wave basins for vessel response and wave penetration tests; three 2D flumes; two concrete flow channels; a wind flume; a dam model basin; and a deep-water tank (CSIR, 2013). One of the concrete flow channels was available and used for all the experimental work. It had a length of 25 m, a width of 0.75 m and a height of 1 m. The upstream end of the flow channel was equipped with wave paddles, as shown in Figure 5-1. Dynamic wave absorption eliminates the influence of reflected waves but the feature was redundant since the experiments focussed only on the initial interaction of the wave with the rigid bodies.

Figure 5-1 Wave paddles featuring dynamic wave absorption

The decision to use the hydraulics laboratory was based on the fact that it enjoys excellent technical support, and that the group operating the facility was part of the larger project team and received funding to cover their labour and operating costs. Furthermore, this facility is the only one of its size and complexity in South Africa (CSIR, 2009).

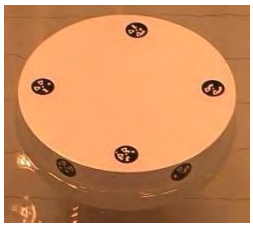
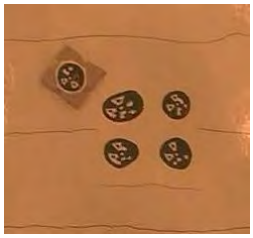
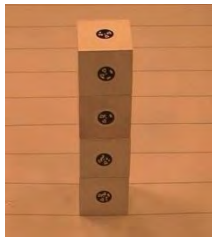
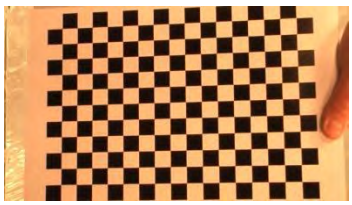
Fiducial marker technology was regarded as a practical and accurate option for tracking movement during the experiments. The use of accelerometers was not considered feasible due to concerns regarding their influence on the movement of the rigid bodies, some of which were relatively light and small. The group that developed the fiducial marker technology were members of the Remote Sensing Research Unit (RSRU) of the CSIR, a group focused on computationally-intensive remote sensing problems (CSIR, 2007) and part of the larger project team. They were responsible for processing the images captured during the experiments and for providing the positions of the fiducial markers in three dimensions.

The author was responsible for capturing the images and conducting the experiments. Travelling to the facility by air necessitated a measure of planning to ensure that productivity was maximised during the three-day time slot allocated for the experimental work. There was no opportunity to conduct test runs before the measurement campaign nor to repeat any of the experiments at a later stage, so a strategy was required to deal with unexpected difficulties. It was decided to prepare various experiments, all of which would be attempted to provide some data redundancy. Table 5-1 contains a summary of the experiments conducted. The last item in the table refers to a calibration exercise that involved moving a chequerboard pattern in front of the camera. The calibration data were used to compensate for lens distortion during image processing.

After filling the flow channel with water and noting the depth, the following procedure was followed for each experiment:

- Allow at least 5 minutes for the water in the flow channel to settle after any previous experiments

Table 5-1 Experiments conducted in concrete flow channel

Description	Example	Description	Example
Floating sphere		Two cubes on platform	
Floating disc		Three cubes on platform	
Submerged armour unit		Four cubes on platform	
One cube on platform		Lens distortion calibration	

- Place the rigid body/bodies in position
- Generate a solitary wave with a specified amplitude
- Start capturing images when the wave is approximately 2 m upstream of the rigid body/bodies
- Continue recording images until the desired interaction is captured
- Download the images onto a laptop computer
- Verify that the images are of reasonable quality through visual inspection

- Repeat the same experiment until at least three sets of images are obtained
- Continue to the next experiment

The solitary wave travelled down the flow channel for approximately 18 m before colliding with one or more of the rigid bodies. The event was filmed using a Panasonic HDC-TM300 camera. The camera was chosen because it was available free of charge and had a feature that allowed a video to be recorded as a set of images in the Joint Photographic Experts Group (JPEG) format. The exposure time of each image was determined automatically by the camera based on the amount of light available. In order to prevent blurring, relatively short exposure times were desired, typically shorter than $1/500^{th}$ of a second. Eight 500 W incandescent lamps were used to ensure a well-lit scene. Figure 5-2 shows a close-up of the area above the platform while Figure 5-3 shows the view from the wave maker end of the flow channel.

Figure 5-2 Experimental setup (lights off)

Figure 5-3 Experimental setup (lights on)

5.3 Data capture and processing

All the planned experiments outlined in Table 5-1 were conducted and data sets consisting of 180 images were obtained from each experiment. Table 5-2 gives a summary of the measurement campaign.

Table 5-2 Summary of the measurement campaign

Experiment / activity	Day	Data sets	Exposure time (s)	Notes
Paint objects white	-	-	-	-
Floating sphere experiment	1	7	1/750	Not rotationally stable
Lens distortion calibration	1	4	1/350 – 1/250	-

Floating disc	1	10	1/1000 – 1/750	-
Submerged armour units	2	7	1/1000	Fiducial images distorted
Construct platform	2	-	-	-
Length, mass measurements	2	-	-	Detailed in Appendix D
One cube on platform	3	3	1/1000 – 1/750	-
Two cubes on platform	3	3	1/1000 – 1/750	-
Three cubes on platform	3	3	1/1000 – 1/750	-
Four cubes on platform	3	3	1/1000 – 1/750	-
Lens distortion calibration	3	2	1/500	-

In addition to the images captured during the experiments, measurements of mass and length were taken and calculations were performed based on these measurements. The data served as input for the numerical simulation and are summarised in Appendix D. A standard steel ruler with graduations in centimetres and millimetres was used for measuring lengths and an electronic scale capable of measuring one-gram increments was used for the mass measurements. Calculations were performed in accordance with the standard practice regarding the use of significant digits (University of Guelph, 2013).

Before any images were captured, all the objects were painted white and Figure 5-4 demonstrates why this was important. The two images on the left were captured with a 1/125 second exposure time while the two images on the right were obtained with a 1/1000 second exposure time under the same lighting conditions. The significant reduction in exposure time was achieved by making the scene almost entirely white, allowing more light to enter the lens and prompting the camera to automatically reduce the exposure time. The first and third images show the disk at rest and the second and fourth show the disk while it is being moved upward as if floating on water when a wave passes underneath. The inserts at the bottom of the second and fourth images show enlargements of the fiducial marker on the top surface of the disk. The longer exposure time resulted in significant blurring of the image when the object was in motion but this was effectively eliminated by reducing the exposure time. Videos demonstrating this test are stored on the DVD attached to the inside of the back cover page. They are called “Video clip 03.avi” and “Video clip 04.avi”.

Figure 5-4 Influence of exposure time on image quality (from left to right): 1/125 s at rest; 1/125 s in motion; 1/1000 s at rest; 1/1000 s in motion

The index of refraction of water is approximately 33% higher than that of air, which results in light bending away from the surface normal when it travels from water to air. This effect caused significant distortion of the fiducial images captured during the “submerged armour units” experiments. Figure 5-5 shows images of the fiducial markers attached to the bottom of the flow channel before (left) and after (right) a wave had passed.

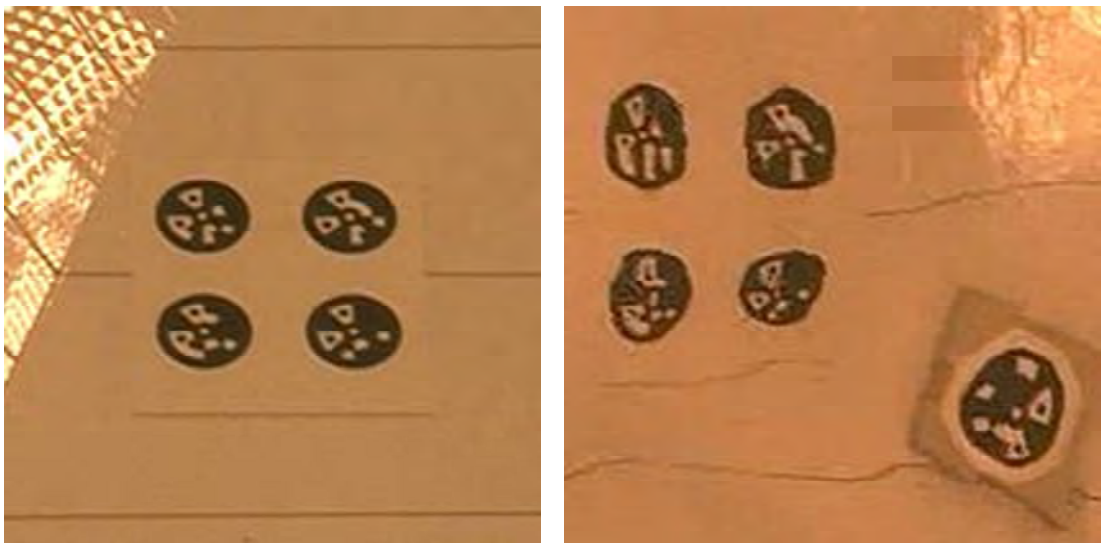


Figure 5-5 Distortion of fiducial images

Corresponding images from different data sets of the same experiment were compared to allow a qualitative assessment of the statistical dispersion. Figure 5-6 shows such a comparison for the “three cubes on a platform” experiment. Vertical blue lines pass through the fiducial markers located on the top of each cube. Horizontal lines pass through the fiducial markers located on the top of each cube in the image on the left and are then extended across the other images to allow a comparison with the positions of the corresponding fiducial markers in the other images.

The images in Figure 5-6 were recorded when the cubes had travelled approximately 0.5 m downstream and the dispersion in their positions appears to be of the order of 0.03 m, or 6% of the distance travelled. The distribution pattern of the cubes was consistent: the cubes that were originally placed on the left and right of the stack remained on the left and right throughout, with the cube originally placed on top of the stack lagging some distance behind. It is however clear that the precision error associated with the experiments far outweighs the sub-millimetre accuracy claimed by the developers of the fiducial marker technology (Vieira, et al., 2008).

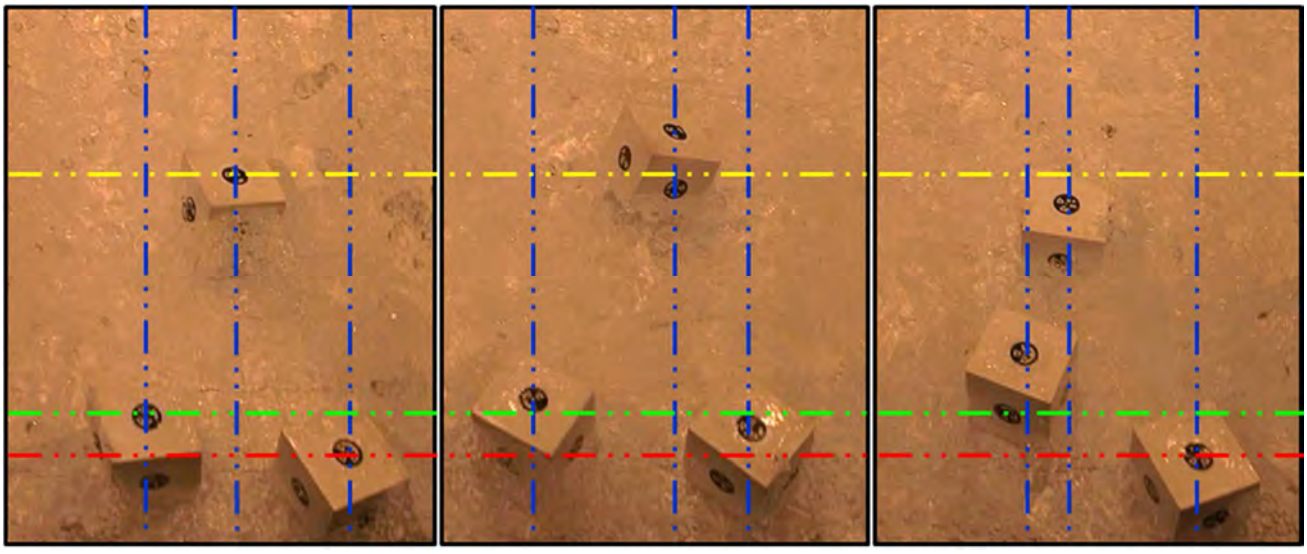


Figure 5-6 Qualitative assessment of statistical dispersion

The RSRU had the capacity to analyse only one of the experiments and a decision had to be made regarding the best candidate. The experiments that involved floating objects were eliminated for two reasons. First, they were very similar to the simulation discussed in the previous chapter and second, buoyancy forces played a significant role in their movement as opposed to momentum transfer as described by Latham. The experiments with the submerged armour units were a risky choice as they suffered from significant image distortion. The remaining options all involved cubes stacked on a platform where the wave's momentum was transferred to the rigid bodies, causing them to slide across the platform. The "four cubes on a platform" experiment was considered to be the best candidate since it had more fiducial markers available to be captured on film and also included free fall of the elevated cubes.

The images captured during the "four cubes on a platform" experiment were submitted to the RSRU, analysed and returned as files listing the temporal position of the fiducial markers. One file was received for each cube in each of the three experiments to give a total of twelve files. The cube placed on the platform at the base of the stack was labelled "Block 1", the cube directly above it was labelled "Block 2", followed by "Block 3" and the top cube was labelled "Block 4".

The information in the files was mostly based on tracking a single fiducial marker, but some of the files were augmented by merging information from other fiducial markers. This ensured longer sequences in cases where a cube turned, presenting a different fiducial marker to the camera. "Block 1" was most affected since its downstream-facing fiducial marker was the only one initially visible but in many cases it was soon obscured by water and was also the first to leave the field of view.

A single fiducial marker was attached to the top surface of the platform at the start of the measurement campaign and it was used to define the origin of a coordinate system. The fiducial marker is visible on the left of the cubes in Figure 5-7. All positions were measured in this coordinate system with the positive x-axis pointing downstream, the positive y-axis pointing to the right of the viewer and the positive z-axis pointing upwards, normal to the platform.

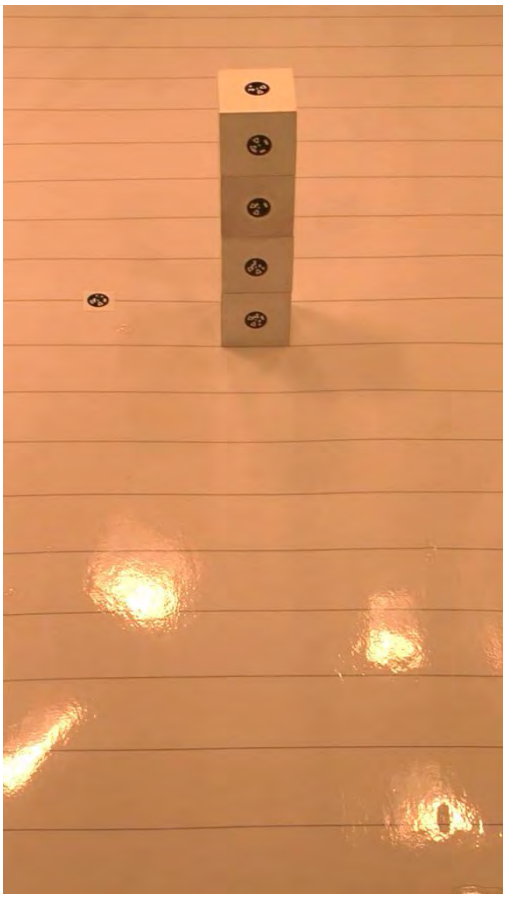


Figure 5-7 Cubes at rest during “Experiment 1”

Filming started well before the solitary wave reached the platform, providing a significant number of images of the cubes at rest. The reflection of the lights on the surface of the platform shown in Figure 5-7 revealed no observable variation when the images were viewed in rapid succession. It was assumed that any vibrations resulting from the approaching wave or any other source were negligible. Changes in the position of an individual cube during this period could therefore be mainly attributed to precision errors associated with image processing.

The mean x-coordinate values of the cubes at rest are listed in Table 5-3 along with the standard deviation for each cube. Although the standard deviations of individual cubes remained relatively small, the data revealed offsets of several millimetres between different cubes. Care was taken to position the cubes at the same x-coordinate in each experiment and to stack them directly on top of each other. The offset of almost 15 mm between “Block 2” and “Block 4” in “Experiment 1” was

therefore regarded as excessive. It represented approximately 30% of the length of the cube and did not correlate well with Figure 5-7.

Table 5-3 Cubes at rest

Experiment	Cube	x-coordinate mean (mm)	Standard deviation (mm)
"Experiment 1"	"Block 1"	50.630	0.545
	"Block 2"	46.809	0.411
	"Block 3"	51.880	1.116
	"Block 4"	61.797	0.357
"Experiment 2"	"Block 1"	53.378	1.253
	"Block 2"	55.487	0.449
	"Block 3"	56.457	0.892
	"Block 4"	52.840	0.814
"Experiment 3"	"Block 1"	54.484	1.460
	"Block 2"	52.497	1.331
	"Block 3"	50.022	0.980
	"Block 4"	60.061	0.562

The variation in the positions of cubes in the same stack and between stacks in different experiments was assessed. This was done by using a single sample consisting of the x-coordinate values of all the cubes in all three experiments. This sample included precision errors resulting from fiducial markers not placed exactly at the centre of the cube's face; precision errors due to inconsistent cube dimensions; and differences in the positions of individual cubes. A standard deviation of 4.124 mm was obtained from a mean of 53.867 mm. The y-coordinate values were reduced by multiples of 50 mm where necessary to account for increased elevation, and a similar calculation was performed.

It yielded a standard deviation of 3.639 mm from a mean of 25.871 mm. The corresponding histograms based on these data samples as well as those in Table 5-3 are given in Appendix E.

Significant changes in the positions of the fiducial markers were recorded during the interaction of the cubes with the solitary wave, as shown in Figure 5-8, Figure 5-9 and Figure 5-10. The figures show the x, y and z-coordinate values as a function of time, respectively. A video of "Experiment 1" is stored as "Video clip 05.avi" on the DVD attached to the inside of the back cover page.

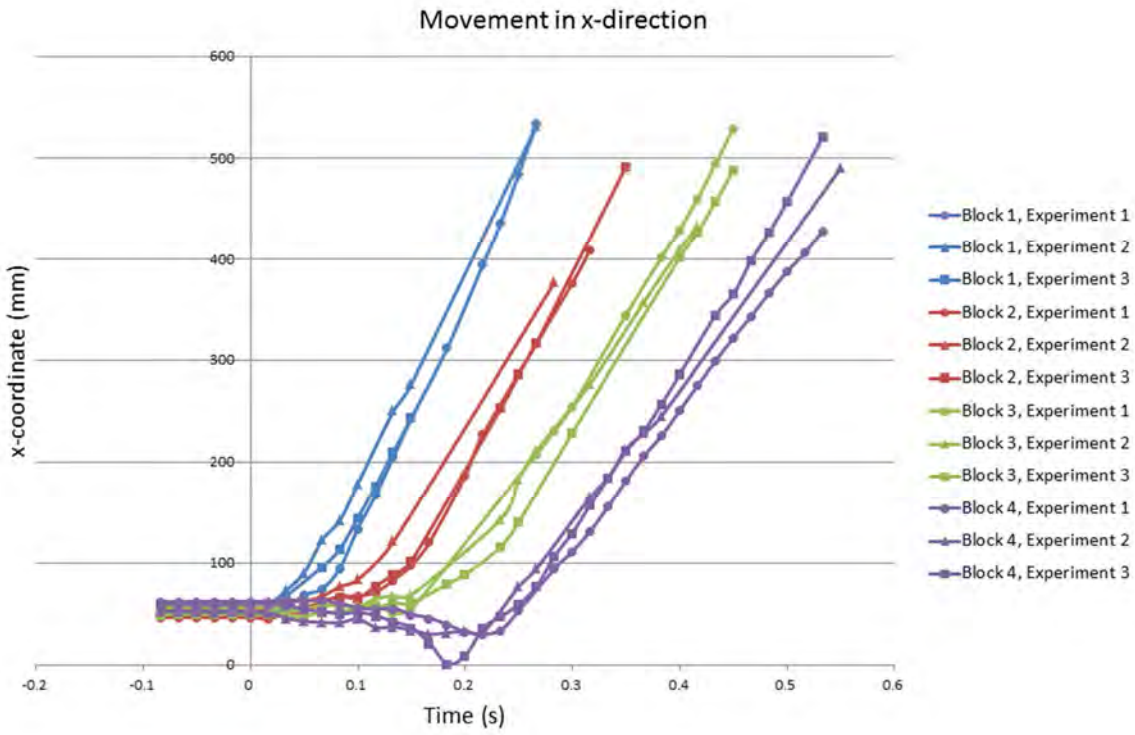


Figure 5-8 X-coordinates of the fiducial markers

The data sets were analysed to determine the statistical dispersion in the samples. As mentioned in the discussion of Figure 5-6, it was expected that the repeatability of the experiments was the greatest contributor to the precision error. The difference between the first and second, first and third and second and third experiments were determined by subtraction of the relevant x, y and z-coordinate values. A total of twelve sets of results were obtained in this way since each of the four cubes was assessed for each of the coordinate directions. A graphical representation of the results is given in Appendix F with an example shown in Figure 5-11.

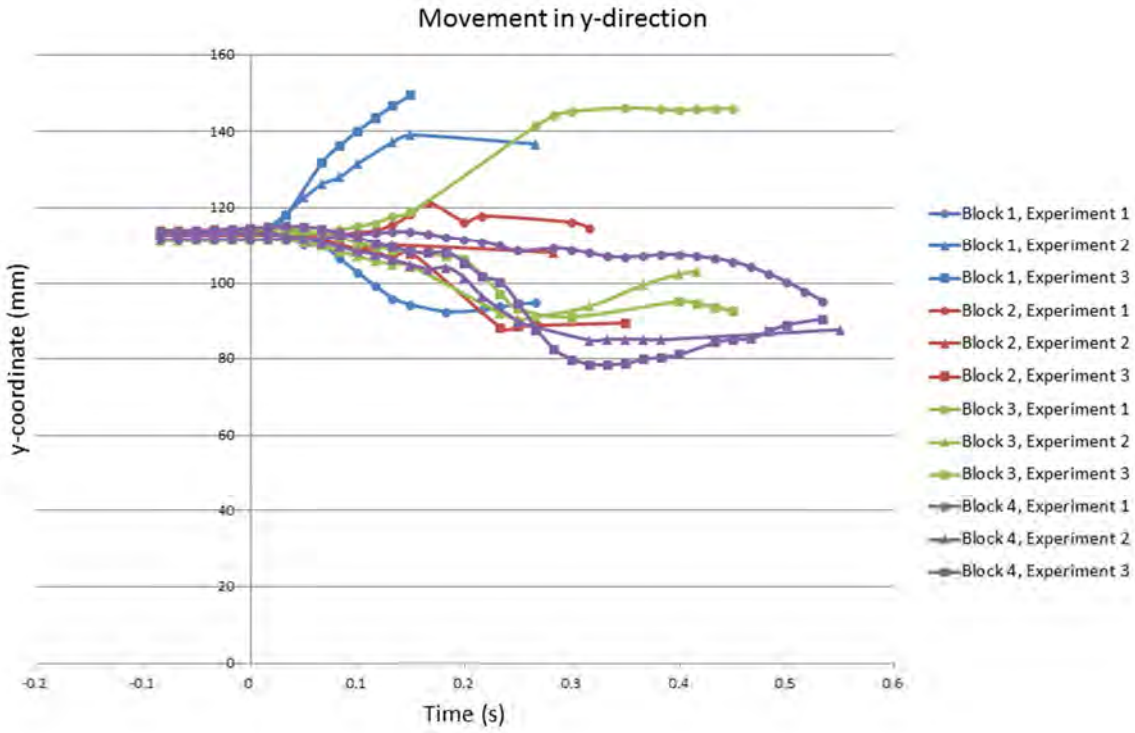


Figure 5-9 Y-coordinates of the fiducial markers

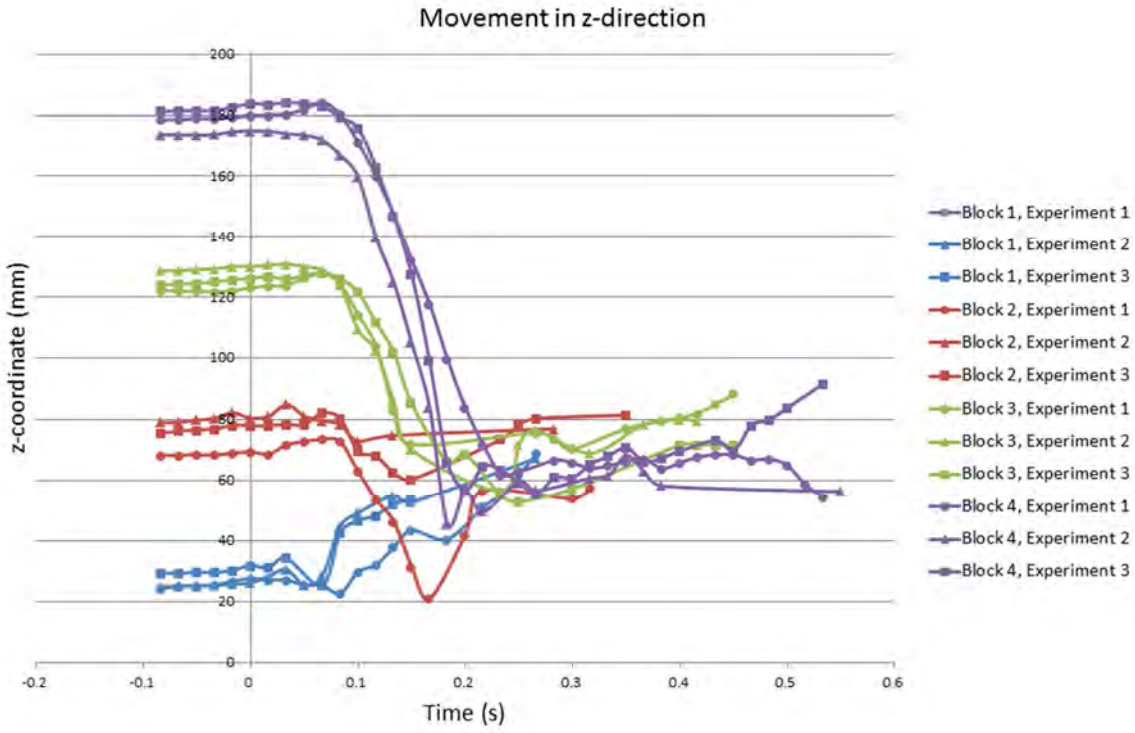


Figure 5-10 Z-coordinates of the fiducial markers

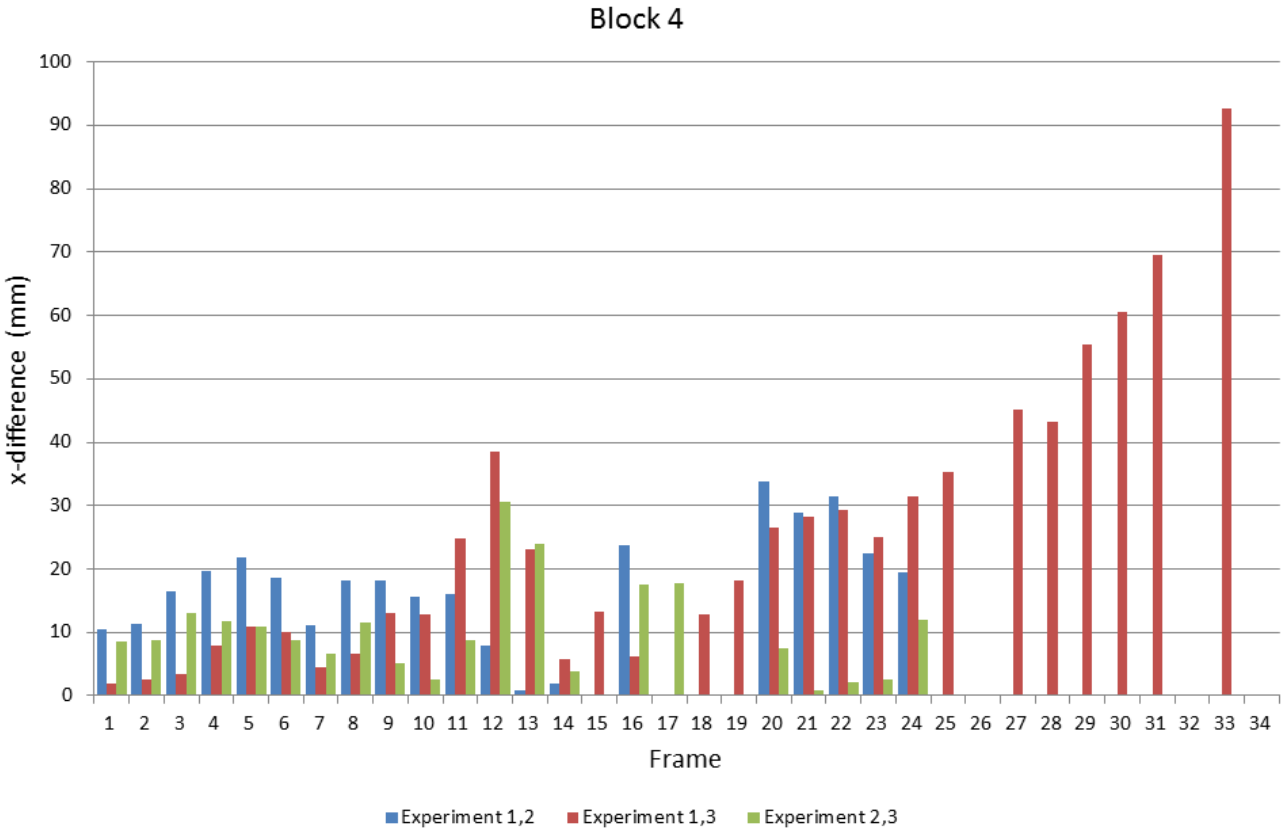


Figure 5-11 Quantitative assessment of statistical dispersion

Differences in the x-coordinate values were generally below 50 mm except for “Block 4”, which showed a maximum difference of 93 mm. Differences of up to 69 mm were recorded in the y-coordinate values although the differences between the y-coordinate values of “Block 2” and “Block 4” were all below 25 mm.. All the differences in the z-coordinate values were below 30 mm except for “Block 4”, which exceeded this limit on four occasions and reached a maximum z-coordinate difference of 54 mm.

5.4 Numerical simulation of flow channel experiments

A CFD model was generated based on the geometry of the flow channel used during the experiments. The solitary wave was generated by collapsing a column of water upstream of the platform and the dimensions of the column were adjusted to obtain a wave with the correct

amplitude. The technical details of the final model are shown in Appendix H, with the mesh and initial flow field shown in Figure 5-12. Additional mesh refinement was applied in the vicinity of the cubes, including the region through which they were expected to travel, as shown in Figure 5-13.

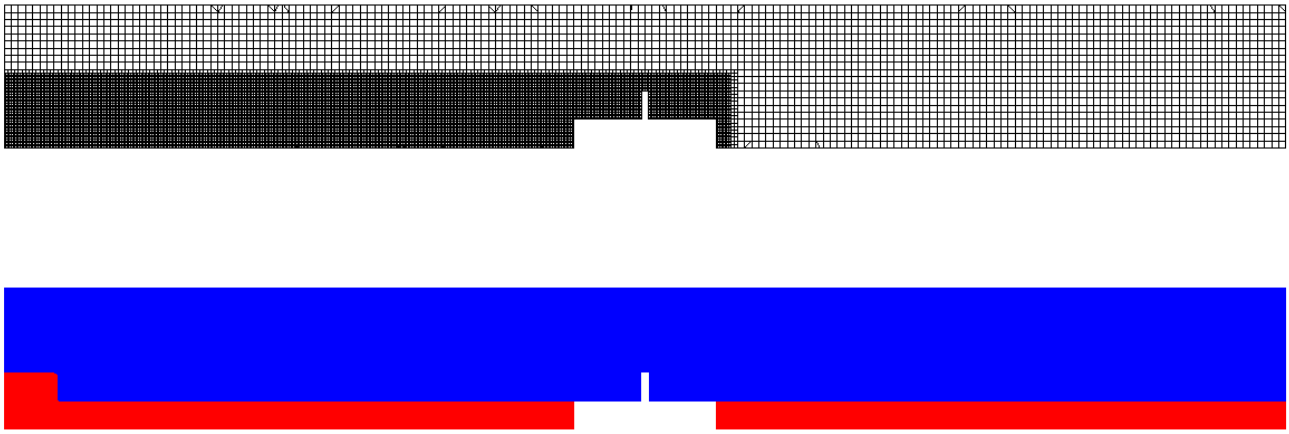


Figure 5-12 Initial mesh (top) and flow field (bottom) of flow channel model

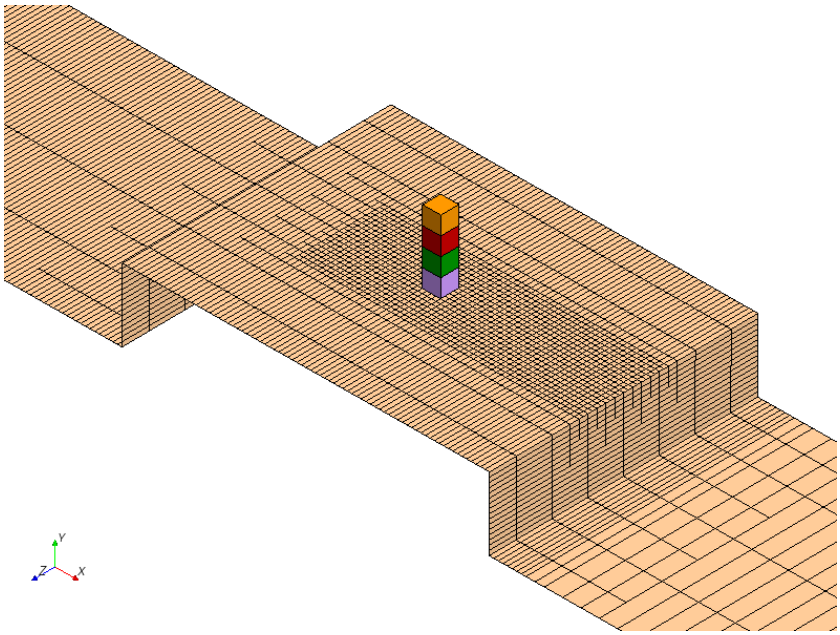


Figure 5-13 Mesh refinement in vicinity of cubes

The CFD simulation used a time step of 0.0025 s, which required Star-CCM+ to perform sets of four time steps per cycle to match the one 0.01 s time step of PhysX. The simulation produced data at a frequency similar to the experiments, which provided a datum every 0.0167 s while a fiducial marker was visible.

The choice of friction coefficients was difficult since they were not measured during the experiments. Published dynamic friction coefficient data were questioned since the cubes were buoyant and could effectively be floating downstream. It was finally decided to use a static friction coefficient of 0.3 and a dynamic friction coefficient of 0.015.

Quantitative comparisons of the simulation and experiment are shown in Figure 5-14 (x-coordinate), Figure 5-15 (y-coordinate) and Figure 5-16 (z-coordinate). The y-coordinate data of the simulation were adjusted by applying a constant offset to allow them to coincide with the initial position of the cubes in the experiment. A video clip of the simulation is stored as “Video clip 06.avi” on the DVD attached to the inside of the back cover page.

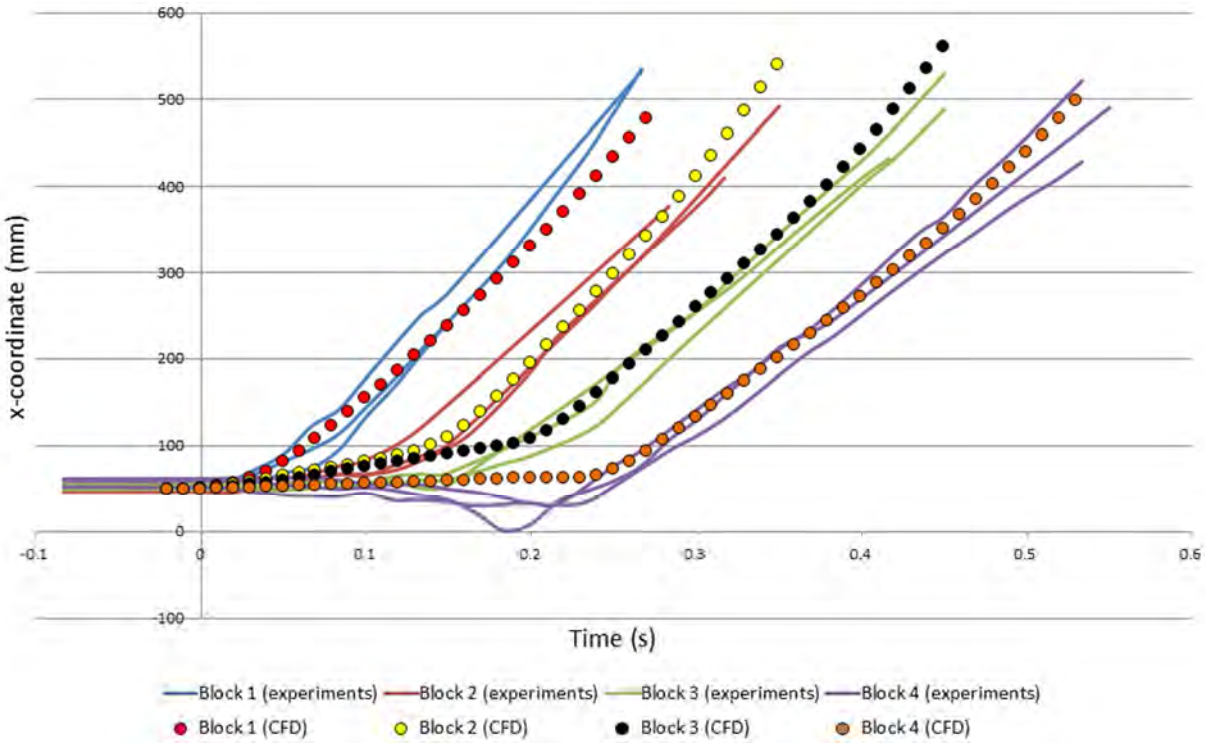


Figure 5-14 Comparison of simulation and experiment (x-coordinate)

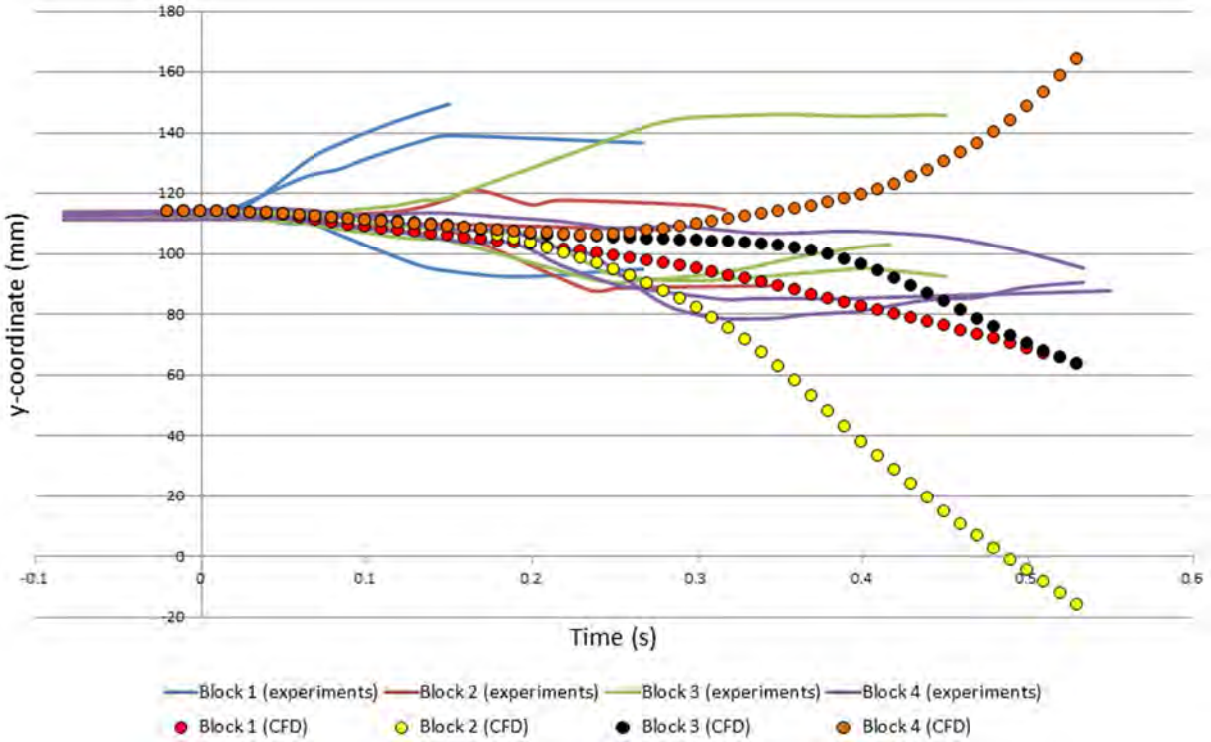


Figure 5-15 Comparison of simulation and experiment (y-coordinate)

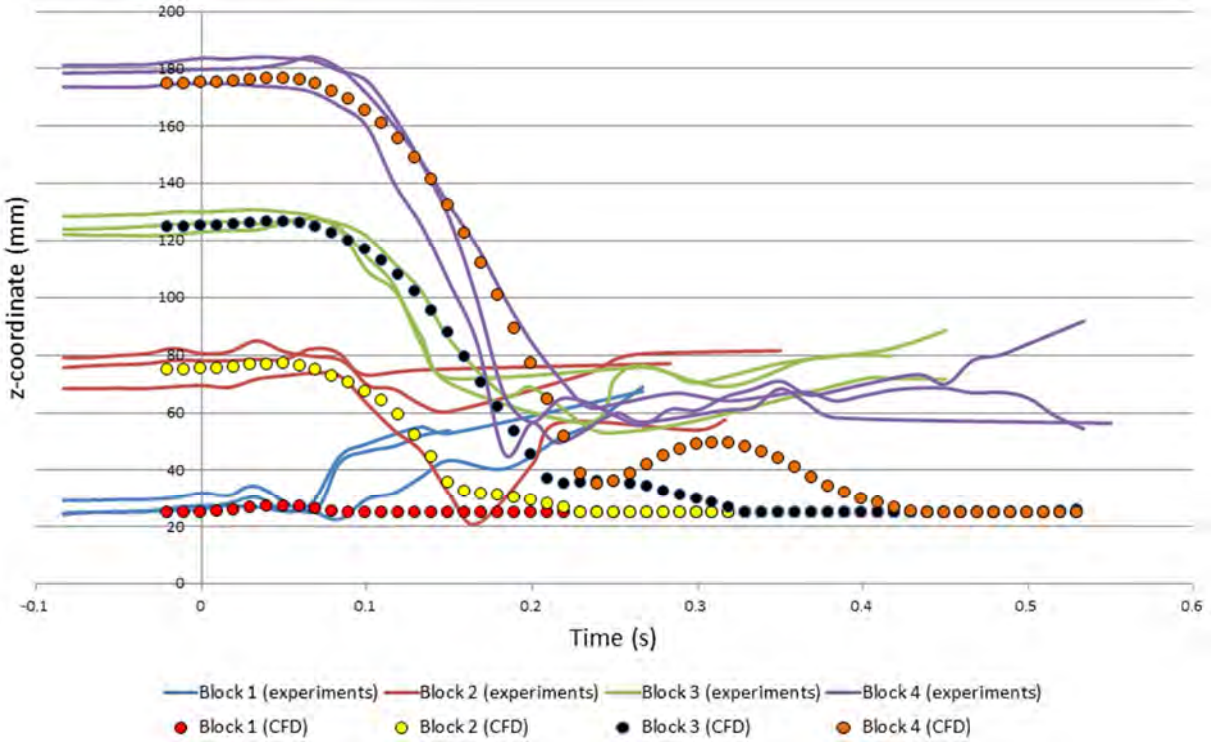


Figure 5-16 Comparison of simulation and experiment (z-coordinate)

5.5 Discussion of results

Not all the experiments produced usable data and the decision taken to conduct a range of experiments from which the best could be selected proved to be a valuable strategy. The disadvantage of such an approach is that it limits the time available for each experiment. This can be problematic when large precision errors are expected since the statistical techniques available to deal with such errors rely on large data samples (Mills & Chang, 2004).

A solitary wave interacting with wooden objects is expected to be a relatively chaotic process, as confirmed by noticeable variations observed in the behaviour of the stacked cubes. During the “four cubes on a platform” experiment, for example, the top three cubes rotated through 90° as they fell during two of the experiments, but in the third experiment the cube second from the bottom did not complete the full rotation. It is suggested that the three sets of data were not enough for a proper statistical analysis, and an opportunity to gather more data would have been of great benefit to the study.

The use of fiducial markers to determine the positions of the cubes proved successful. The precision error associated with image processing was determined using stationary cubes and produced standard deviation values that were small compared to variations in behaviour observed between experiments. No assessment was made of bias errors but there were indications that they could be significant. Differences in position of up to 15 mm were recorded for stacked cubes when photographic evidence suggested that their vertical alignment was correct to within a few millimetres. Other evidence also suggested a significant bias error: the elevation of the cubes was measured to be as high as 92 mm as they reached the end of the platform (Figure 5-10), while photographic evidence suggested that they all slid across the platform close to the surface.

In spite of the small sample size and possible bias errors, the data of each cube in the “four cubes on a platform” experiment were clearly grouped together as shown in Figure 5-8 and Figure 5-10. These distinguishable trends allowed a comparison of the experimental data with the results of the CFD simulation.

The static and dynamic friction coefficients were not measured during the experiments and had to be chosen based on published data. It would have been of great benefit if more reliable friction coefficient data were available since tests done with PhysX indicated that the choice of friction coefficient had a noticeable effect on the behaviour of the cubes. A very low dynamic friction coefficient allowed the bottom cube to slide out from under the stack, with the stack then descending vertically. A moderate dynamic friction coefficient, as was used in the simulation, resulted in the top three cubes falling backwards as the momentum of the wave swept the bottom cube downstream.

The initial impulse of the wave caused the bottom cube to move downstream with the rushing water. As the rest of the cubes descended, they came in contact with the water and the resulting momentum transfer also swept them downstream. This transfer of momentum from the wave to the cubes was captured successfully in the simulation as indicated by a close correlation with the experimental data.

5.6 Conclusion

Experiments conducted in the hydraulics laboratory of the CSIR provided the experimental data required for the verification of the proposed simulation algorithm. Images of fiducial markers taken with a digital camera were successfully processed to yield a large amount of data regarding changes in the position of objects as they interacted with a solitary wave. Although the interaction of the wave and the objects produced noticeable variations with each repetition of the experiment, clear trends emerged. A CFD simulation of the “four cubes on a platform” case produced results that compared well with the experimental data.

6 Conclusions

6.1 Preamble

The goal of the study was to determine if commercial CFD software and physics middleware could be used to simulate the interactions between a relatively large number of rigid bodies and a two-phase fluid. The intended application was the analysis of breakwater structures. A simulation algorithm was developed after a review of available literature, and then verified numerically and experimentally. The main findings are summarized and recommendations are made for further study.

6.2 Summary of findings

A study of available literature revealed significant research activity in the field of multi-physics and, specifically, the simulation of fluid–structure interaction. The interaction between waves and coastal

structures is one such example and receives keen interest from the engineering community due to the vital role that harbours play in most economies.

The analysis of breakwaters is challenging since they consist of large numbers of complex structures that interlock, with individual armour units that may move or even dislodge. The strategies followed by various researchers were reviewed before a simulation method was developed where “shell scripts” are used to make commercial CFD software (Star-CCM+) and freely available middleware (PhysX) work in unison. The CFD software models the two-phase fluid and provides force and moment data to the middleware which, in turn, is responsible to determine how the armour units move. The “back-and-forth” transfer of information is managed by the operating system (Linux).

The method was first validated using two simple test cases and then verified numerically and experimentally. The numerical verification process involved a simulation of a boat hull encountering head waves. The translation and rotation of the hull was first simulated with well-known commercial CFD software and then using the proposed simulation algorithm. A comparison of the results raised some issues with the way in which the commercial CFD software related forces and moments with the translation and rotation of the rigid body. The issues were raised with the software developers but they remain unresolved. The force responsible for the reported vertical translation could be calculated from first principles and used as input for a PhysX simulation. The results correlated very well with the original translation data, as shown in Figure 6-1.

The method was verified experimentally with tests performed in the hydraulics laboratory of the CSIR. The experiments all studied the interaction of a solitary wave with rigid objects. A range of experiments were conducted and one was selected for further data processing. Fiducial targets were placed on the objects to track their movement. Digital images of the fiducial targets were processed to obtain their positions as a function of time. In spite of suspected bias errors, the method was found to be suitably accurate and clear trends were observed in the experimental data.

The experimental data were compared to the results of a simulation that employed the proposed simulation algorithm. The transfer of momentum from the wave to the rigid objects was of particular

interest. The simulations showed that as the cubes came in contact with the water, they were swept downstream in a manner consistent with the observations from the experiment. Figure 6-2 gives an indication of the correlation that was achieved.

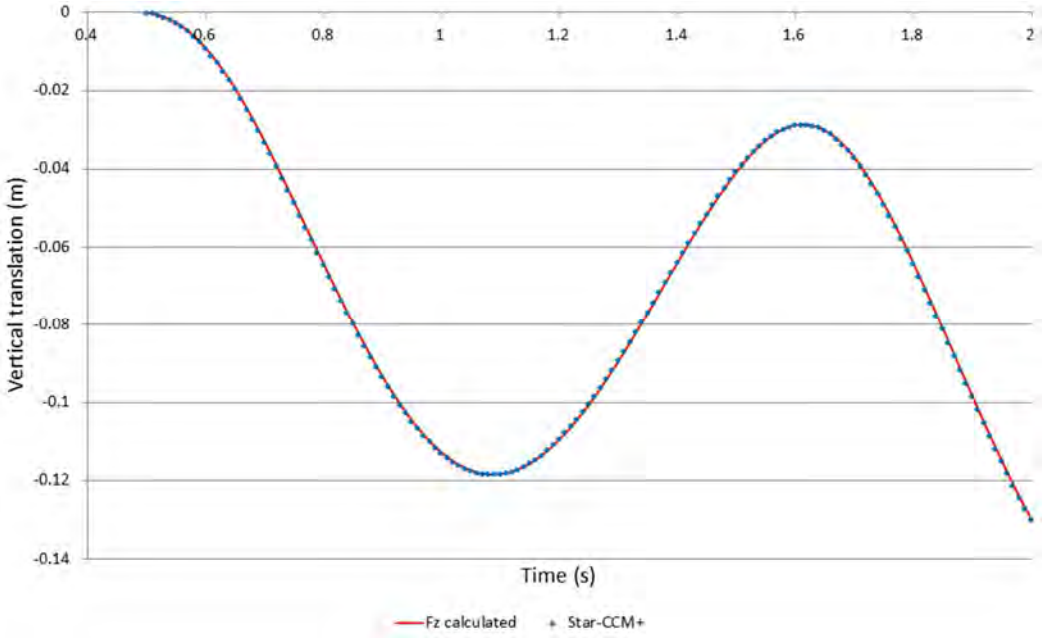


Figure 6-1 Vertical translation reported by Star-CCM+ and calculated with PhysX

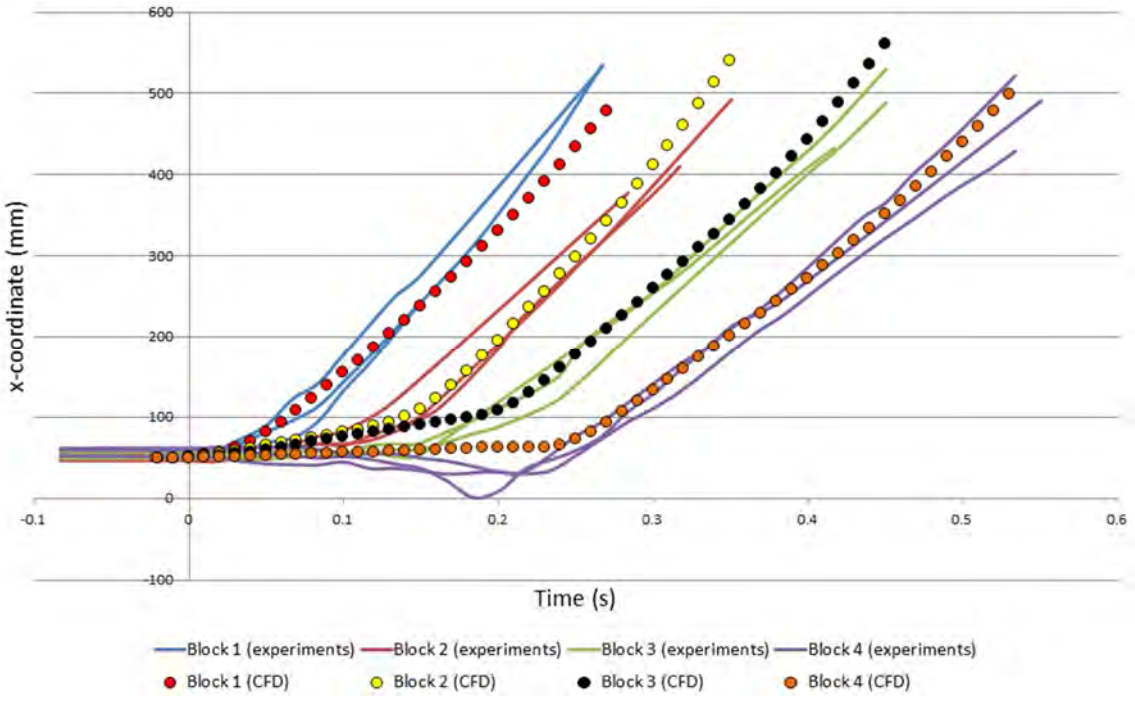


Figure 6-2 Correlation between experiment and simulation (x-coordinates)

6.3 Recommendations for further study

The development, validation and verification of the proposed simulation algorithm was completed, but some suggestions for further study are outlined below.

The simulation algorithm performed well, but can be streamlined. Currently the simulation file has to be opened, saved and closed for every time step simulated. In addition, another (empty) simulation file must be opened to generate a new geometry using the latest position and orientation data, and then closed. Opening, saving and closing files takes time, especially if the files are large. The algorithm could be written such that it opens the simulation file once at the start of the simulation without the need to save and close it until the end of the simulation.

Studies investigating the interaction of waves and rigid bodies often analyse cases involving floating objects. The buoyancy forces responsible for movement in such cases are not considered as important as the forces resulting from momentum transfer when investigating the mechanisms responsible for breakwater damage. The movement observed during the “four cubes on a platform” experiment was mainly due to momentum transfer from the solitary wave to the cubes, which made it a good choice for analysis. The precision error, which appears to be significant, can be quantified using standard statistical analysis tools which all rely on a large data sample. Priority should be given to increasing the data sample should future opportunities for additional experimental work arise.

Results indicated a bias error associated with the image processing of the fiducial markers. Although the responsibility for improving the technology lies with the developers, future experiments can be designed to assist. For example, it would be relatively easy to capture images of a platform on which a large number of stationary cubes are stacked. The number of fiducial markers can be increased from one to five per face by placing one fiducial in the middle and one in each of the four corners. Placing the stacks along the length and breadth of the platform at known distances will produce a large amount of data that can be used to determine how the bias error varies as a cube moves across the platform.

Star-CCM+ offers powerful post-processing features and these could be harnessed to add value to the method. For example, a scene could be created showing the armour units in grey, with individual armour units that conform to certain criteria in another colour. A prominent colour can be chosen to identify the armour units that experience moments, forces, translations or rotations that are above a user-specified threshold. It would then be easy to identify armour units that carry a high risk of breakage or dislodgement.

7 References

4Front Technologies, 2010. *PhysXforQuest.* [Online]
Available at:
http://www.physxforquest.com/onlinehelp/DynamXChannels/Channels/DynamX_Actor/PhysX_Actors.htm
[Accessed 29 March 2013].

Ai, C. & Jin, S., 2010. Non-hydrostatic finite volume model for non-linear waves interacting with structures. *Computers & Fluids 39*, pp. 2090-2100.

Ai-min, L. I., Rui-ling, L. V. & Chu-sheng, L. I. U., 2008. A virtual test of screening technology based on the AGEIA PhysX. *Journal of China University of Mining & Technology 18*, pp. 300-304.

Autodesk, 2013. *Transient Parameters.* [Online]
Available at: http://wikihelp.autodesk.com/Simulation_CFD/enu/2013/Help/0138-The_Proc138/0347-Solving_347/0357-Solve_Qu357/0359-Transien359
[Accessed 17 October 2013].

Bakker, P. et al., 2003. *Development of Concrete Breakwater Armour Units.* Moncton, Canadian Society for Civil Engineering.

Berger, J. W. et al., 2000. Computerized stereochronoscopy and alternation flicker to detect optic nerve head contour change. *Ophthalmology,* Volume 107 (number 7), p. 1316–1320.

Bose, C. B. & Amir, I., 1990. Design of Fiducials for Accurate Registration Using Machine Vision. *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence,* December, Volume 12 (number 12), pp. 1196-1200.

Bungartz, H.-J. & Schäfer, M., 2006. *Fluid-Structure Interaction: Modelling, Simulation, Optimization.* New York(New York): Springer.

CD-adapco, 2012. *USER GUIDE: STAR-CCM+ Version 7.02.008*, London: CD-adapco.

CFD Research Corporation, 1997. *CFD-FASTRAN™ User Manual.* Huntsville(Alabama): CFD Research Corporation.

Chevron Australia, 2013. *Economic Benefits.* [Online]
Available at: http://www.chevronaustralia.com/ourbusinesses/wheatstone/economicbenefits.aspx
[Accessed 25 May 2013].

Cooper, A. K. et al., 2008. *A Preliminary Physics-engine Model of Dolosse Interacting with One Another.* Cape Town, South African Conference on Computational and Applied Mechanics.

CSIR, 2007. *Remote Sensing Research Unit.* [Online]
Available at: http://www.csir.co.za/meraka/rsru/
[Accessed 13 August 2013].

CSIR, 2009. *Model Hall Rev 290609 01 - Model_Hall_Rev29060901.pdf.* [Online]
Available at:
http://www.csir.co.za/Built_environment/Infrastructure_engineering/pdfs/Model_Hall_Rev29060901.pdf
[Accessed 3 August 2013].

CSIR, 2013. *Coastal Engineering and Port Infrastructure.* [Online]
Available at:
http://www.csir.co.za/Built_environment/Infrastructure_engineering/cepi.html#hydraulics
[Accessed 2 August 2013].

Cummings, R. M., Forsythe, J. R., Morton, S. A. & Squires, K. D., 2003. Computational challenges in high angle of attack flow prediction. *Progress in Aerospace Sciences 39,* p. 369–384.

Cundall, P. A. & Strack, O. D. L., 1979. Discrete numerical-model for granular assemblies. *Geotechnique 29,* p. 47–65.

Dentale, F., Donnarumma, G. & Carratelli, E. P., 2012. Wave Run Up and Reflection on Tridimensional Virtual Breakwater. *Journal of Hydrogeology & Hydrologic Engineering,* 1(1).

Drumwright, E., 2008. A Fast and Stable Penalty Method for Rigid Body Simulation. *Institute of Electrical and Electronics Engineers Transactions on Visualization and Computer Graphics,* 14(1), pp. 231-240.

Efrat, A. & Gotsman, C., 1994. Subpixel Image Registration Using Circular Fiducials. *International Journal of Computational Geometry & Applications 04,* pp. 403-422.

Finnegan, W. & Goggins, J., 2012. Numerical simulation of linear water waves and wave–structure interaction. *Ocean Engineering 43*, pp. 23-31.

Gledhill, I. et al., 2012. *The use of orientational order parameters for characterisation of coastal breakwaters.* Johannesburg, South African Conference on Computational and Applied Mechanics.

Hadžić, I., Hennig, J., Perić, M. & Xing-Kaeding, Y., 2005. Computation of flow-induced motion of floating bodies. *Applied Mathematical Modelling 29,* pp. 1196-1210.

Hahn, J. K., 1988. *Realistic Animation of Rigid Bodies.* New York, Association for Computing Machinery, pp. 299-308.

Harlen, O. G., 2006. *Chapter 3: Index Notation.* [Online]
Available at: http://www1.maths.leeds.ac.uk/~oliver/MATH2360/chap3.pdf
[Accessed 11 May 2013].

Higuera, P., Lara, J. L. & Losada, I. J., 2013. Simulating coastal engineering processes with OpenFOAM®. *Coastal Engineering 71*, pp. 119-134.

Hongchang, L. et al., 2012. CFD–DEM simulation of material motion in air-and-screen cleaning device. *Computers and Electronics in Agriculture 88,* p. 111–119.

Hosseininia, E. S., 2012. Discrete element modeling of inherently anisotropic granular assemblies with polygonal particles. *Particuology 10,* p. 542–552.

Kaidi, S., Rouainia, M. & Ouahsine, A., 2012. Stability of breakwaters under hydrodynamic loading using a coupled DDA/FEM approach. *Ocean Engineering 55,* pp. 62-70.

Lara, J. L., Losada, I. J. & Guanche, R., 2008. Wave interaction with low-mound breakwaters using a RANS model. *Ocean Engineering 35,* pp. 1388-1400.

Latham, J.-P.et al., 2008. Modelling of massive particulates for breakwater engineering using coupled FEMDEM and CFD. *Particuology 6,* pp. 572-583.

Liu, Y., Ma, D. J. & Sun, D. J., 2007. Numerical investigation on influence of forebody geometrical disturbance on asymmetry of flow over a slender body. *Acta Aerodyn Sin,* pp. 43-47.

Li, Y. & Lin, M., 2012. Regular and irregular wave impacts on floating body. *Ocean Engineering 42,* pp. 93-101.

López de Ipiña, D., Mendonça, P. R. S. & Hopper, A., 2002. TRIP: A Low-Cost Vision-Based Location System for Ubiquitous Computing. *Personal and Ubiquitous Computing 6,* pp. 206-219.

Löhner, R., 2008. *Applied Computational Fluid Dynamics Techniques.* 2nd ed. Chichester: John Wiley & Sons Ltd.

Madsen, J., Pechdimaljian, N. & Negrut, D., 2007. *SBEL - University of Wisconsin.* [Online] Available at: http://sbel.wisc.edu/Publications/
[Accessed 24 March 2013].

Mills, A. F. & Chang, B. H., 2004. *Error Analysis of Experiments,* Los Angeles: University of California.

Mollanouri Shamsi, M. M. & Mirghasemi, A. A., 2012. Numerical simulation of 3D semi-real-shaped granular particle assembly. *Powder Technology 221,* p. 431–446.

Muzaferija, S., Peric, M., Sames, P. & Schelin, T., 1999. *A two-fluid Navier-Stokes solver to simulate water entry.* Washington, D.C., National Academy Press.

Naimark, L. & Foxlin, E., 2002. *Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker.* Darmstadt, Institute of Electrical and Electronics Engineers.

Negrut, D. et al., 2011. *Simulation of Multibody Dynamics Leveraging New Numerical Methods and Multiprocessor Capabilities.* Atlanta, National Science Foundation.

Owen, C. B., Xiao, F. & Middlin, P., 2002. *What is the best fiducial?.* Darmstadt, Institute of Electrical and Electronics Engineers.

Paik, K.-J., Carrica, P. M., Lee, D. & Maki, K., 2009. Strongly coupled fluid–structure interaction method for structural loads on surface ships. *Ocean Engineering 36,* pp. 1346-1357.

Pennec, F. et al., 2013. A combined finite-discrete element method for calculating the effective thermal conductivity of bio-aggregates based materials. *International Journal of Heat and Mass Transfer 60,* p. 274–283.

Progressive Media Group Limited, 2013. *Nuclear Engineering International.* [Online] Available at: http://www.neimagazine.com/news/newsareva-signs-up-for-star-ccm-simulation-software
[Accessed 10 November 2013].

Ryzhakov, P. B., Rossi, R., Idelsohn, S. R. & Oñate, E., 2010. A monolithic Lagrangian approach for fluid–structure interaction. *Computational Mechanics 46,* pp. 883-899.

Shi, G. H., 1988. *Discontinuous Deformation Analysis: a New Numerical Model for Statics and Dynamics of Block Systems,* Berkeley: University of California.

Shiu, Y. C. & Ahmad, S., 1989. *3D location of circular and spherical features by monocular model-based vision.* Cambridge, Institute of Electrical and Electronics Engineers.

Sourceforge, 2009. *Reactivision 1.4.* [Online]
Available at: http://reactivision.sourceforge.net/
[Accessed 20 May 2013].

Tsuji, Y., Tanaka, T. & Ishida, T., 1992. Lagrangian numerical-simulation of plug flow of cohesionless particles in a horizontal pipe. *Powder Technology 71,* p. 239–250.

Ubbink, O. & Issa, R. I., 1999. A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes. *Journal of Computational Physics 153,* p. 26–50.

University of Guelph, 2013. *Significant digits.* [Online]
Available at: http://www.physics.uoguelph.ca/tutorials/sig_fig/SIG_dig.htm
[Accessed 15 August 2013].

Vieira, R., van den Bergh, F. & van Wyk, B. J., 2008. *Fiducial-based monocular 3D displacement measurement of breakwater armour unit models.* Cape Town, PRASA.

Wacławczyk, T. & Koronowicz, T., 2008. Comparison of CICSAM and HRIC High-resolution Schemes for Interface Capturing. *Journal of Theoretical and Applied Mechanics 46,* pp. 325-345.

Zhang, D. & Whiten, W. J., 1996. The calculation of contact forces between particles using spring and damping models. *Powder Technology 88,* pp. 59-64.

Zhang, X., Huang, Y., Han, W. & Chen, X., 2012. Research of flexible beam impact dynamics based on space probe-cone docking mechanism. *Advances in Space Research 49,* p. 1053–1061.

8 APPENDIX A: Mathematical Notation

Index notation

According to this convention, when an index variable appears twice in a single term it implies summation of that term over all the values of the index (Harlen, 2006). So, $\forall\ i \in \{1, 2, \cdots, n\}$:

$$a_i b_i = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

For multiple indices, $\forall\ i \in \{1, 2, \cdots, n\}$ and $j \in \{1, 2, \cdots, m\}$, $a_i = b_{ij} c_j$ implies:

$$a_1 = b_{11} c_1 + b_{12} c_2 + \cdots + b_{1m} c_m$$

$$a_2 = b_{21} c_1 + b_{22} c_2 + \cdots + b_{2m} c_m$$

$$\vdots$$

$$a_n = b_{n1} c_1 + b_{n2} c_2 + \cdots + b_{nm} c_m$$

Vector

Consider a coordinate system with axes $x_1, \cdots, x_n$ and unit basis vectors $\bar{e}_1, \cdots, \bar{e}_n$.

Then vector $\bar{u} = (u_1, u_2, \cdots, u_n) = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$

In index notation:

$$\bar{u} = u_i \bar{e}_i \ \forall\ i \in \{1, 2, \cdots, n\}$$

Vector differential operator

$$\nabla = \frac{\partial}{\partial x_i} \bar{e}_i \ \forall\ i \in \{1, 2, \cdots, n\}$$

Vector dot product

$$\bar{u} \cdot \bar{v} = u_i v_i \ \forall\ i \in \{1, 2, \cdots, n\}$$

$$\nabla \cdot \bar{u} = \frac{\partial u_i}{\partial x_i} \quad \forall \; i \in \{1, 2, \cdots, n\}$$

Divergence theorem

$$\iiint_V (\nabla \cdot \bar{u}) dV = \oiint_S (\bar{u} \cdot \bar{n}) dS$$

Where $\bar{n}$ is the outward pointing unit normal vector on the boundary of dV.

In index notation:

$$\iiint_V \frac{\partial u_i}{\partial x_i} dV = \oiint_S u_i n_i dS \quad \forall \; i \in \{1, 2, \cdots, m\}$$

Kronecker delta

$$\delta_{ij} = \begin{cases} 0 \; if \; i \neq j \\ 1 \; if \; i = j \end{cases} \quad \forall \; i \in \{1, 2, \cdots, n\}$$

Standard deviation

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y})^2}$$

Where N is the sample size, y is the observed values and $\bar{y}$ is the mean of the observed values.

9 APPENDIX B: Text Files used by C++ Program

general.txt

The file contains six lines of numbers, with each number separated by a space. All numbers in the file are rational numbers, unless indicated otherwise.

Line 1: Density (in kg/m^3); coefficient of restitution; coefficient of static friction; coefficient of dynamic friction; and friction combination method for the seabed material (an integer value)

Line 2: Density (in kg/m^3); coefficient of restitution; coefficient of static friction; coefficient of dynamic friction; and friction combination method for the armour unit material (an integer value)

The coefficient of restitution is determined by averaging the values of the two materials in contact, but the static and dynamic friction coefficients between materials are determined by the friction combination method. Since there are only two materials, both must use the same friction combination method.

The options are:

0 = the average of the values

1 = the minimum value

2 = the product of the values

3 = the maximum value

Line 3: The number of armour units modelled (an integer value); the type (an integer value); and a scale factor. The current options for the type of armour unit are:

0 = dolos

3 = cube

Line 4: The time step size (in seconds)

Line 5: Height of ground plane (in meter)

Line 6: The distances in the x, y and z-direction (respectively) from the origin to the domain boundaries (the domain is assumed to be symmetric around the origin, so only half distances are specified, in metres)

pose.txt

Contains a list of rational numbers describing the position and orientation of armour units, in the format:

P_{x1}

P_{y1}

P_{z1}

A_{x1}

A_{y1}

A_{z1}

…

P_{xn}

P_{yn}

P_{zn}

A_{xn}

A_{yn}

A_{zn}

P_{x1} gives the distance, in metres, by which the centroid of armour unit 1 is moved from the origin along the positive x-axis of a Cartesian coordinate system, P_{y1} the distance from the origin along the positive y-axis, and so forth for each of the n armour units modelled.

A_{x1} gives the angle, in radian, by which armour unit 1 is rotated around the positive x-axis of a Cartesian coordinate system, A_{y1} the angle around the positive y-axis, and so forth for each of the n armour units modelled.

force.txt

Contains a list of rational numbers describing the forces and moments acting on armour units, in the format:

F_{x1}

F_{y1}

F_{z1}

M_{x1}

M_{y1}

M_{z1}

…

F_{xn}

F_{yn}

F_{zn}

M_{xn}

M_{yn}

M_{zn}

F_{x1} gives the force, in Newton, acting on armour unit 1 along the positive x-axis of a Cartesian coordinate system, F_{y1} the force acting along the positive y-axis, and so forth for each of the n armour units modelled.

M_{x1} gives the moment, in Newton-metres, acting on armour unit 1 around the positive x-axis of a Cartesian coordinate system, M_{y1} the moment around the positive y-axis, and so forth for each of the n armour units modelled.

physx_execute.txt

Contains and integer number, either 0 or 1. PhysX will perform a calculation as long as the file contains a 1 and will not execute if the value is 0.

simulating.txt

Contains an integer number, either 0 or 1. The executable file “Breakwater_Basic.exe” will execute as long as the file contains a 1 and terminate if the value is 0.

ccm_execute.txt

Contains and integer number, either 0 or 1. Star-CCM+ will perform a calculation as long as the file contains a 1 and will not execute if the value is 0.

10 APPENDIX C: Technical Data of Two-Block Simulation

Variable	**Value**
Length of flow domain	15 m
Height of flow domain	5 m
Width of flow domain	5 m
Approximate mesh size	190976 cells
Number of blocks	2
Initial space between blocks	1 m
Block dimensions	1 m x 1 m x 1 m
Block density	400 kg/m^3
Coefficient of restitution (flow domain boundaries)	0.05
Static friction coefficient (flow domain boundaries)	0.5
Dynamic friction coefficient (flow domain boundaries)	0.5
Coefficient of restitution (blocks)	0.8
Static friction coefficient (blocks)	0.4
Dynamic friction coefficient (blocks)	0.4
Method of coefficient combination	Averaging
Time step	0.01 s
Initial water column length	3.75 m
Initial water column height	2 m
Initial water column width	5 m
Turbulence model	k-epsilon
Free surface method	VOF
Air density	Ideal gas law behaviour assumed
Time stepping method	Implicit unsteady

11 APPENDIX D: General Measurements

Description	**Parameter**	**Value**[1]	**Units**
Floating sphere	Circumference	0.7296	m
	Mass of plastic	0.131	kg
	Mass of air[2]	0.008034	kg
	Density[2]	21.2	kg/m^3
Floating disc	Diameter	0.15016	m
	Height	0.04917	m
	Mass	0.479	kg
	Density[2]	550	kg/m^3
Armour unit	Mass	0.0196	kg
	Density[3]	1048.3	kg/m^3
	Volume[2]	1.87×10^{-5}	m^3
Cubes	Length/Width/Height	0.4944	m
	Mass	0.0763	kg
	Volume[2]	0.0001209	m^3
	Density[2]	631	kg/m^3
Water depth (platform present)[4]	Depth	0.2023	m
Water depth (platform absent)[4]	Depth	0.2833	m
Wave amplitude (platform present)[4]	Amplitude	0.1054	m
Wave amplitude (platform absent)[4]	Amplitude	0.1339	m
Platform	Length	1.020	m
	Width	0.7267	m
	Height	0.2051	m

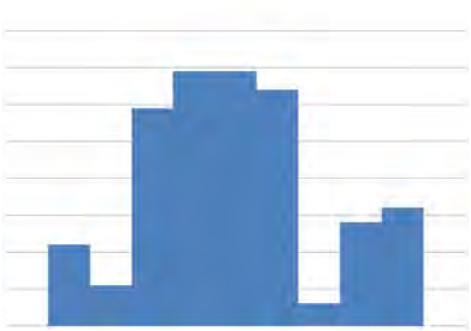
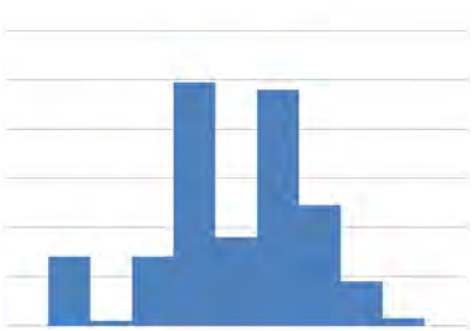
1 Measured data represents the average of three independent measurements.

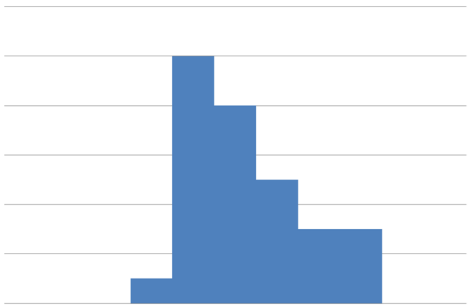
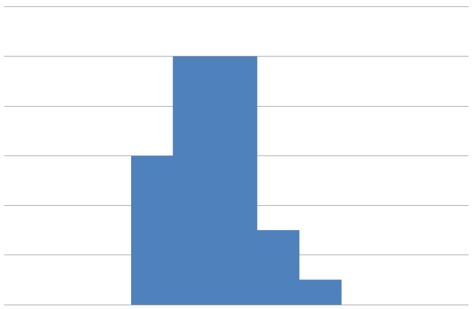
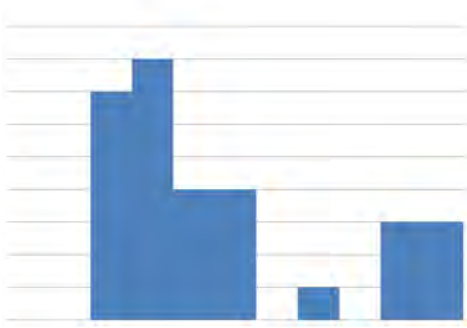
2 Calculated from measured data.

3 Data obtained from supplier.

4 Measured 15 m downstream of wave generator.

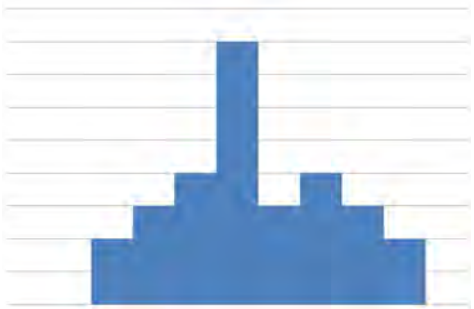
12 APPENDIX E: Histograms

Experiment 1, 2 and 3	x-coordinates 	y-coordinates 

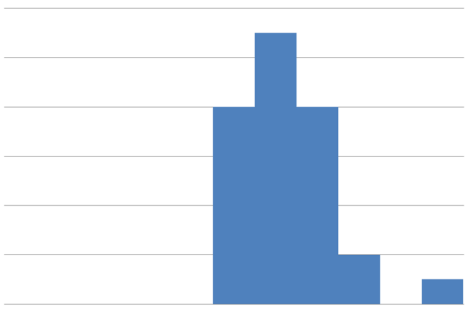
Experiment 1 (x-coordinates)	Block 1 	Block 2 
	Block 3 	Block 4 

Experiment 2 (x-coordinates)

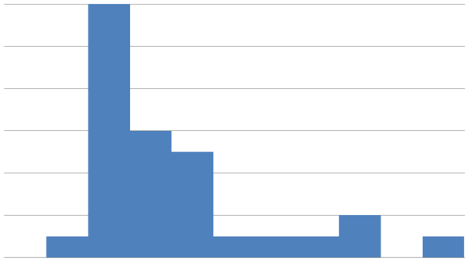
Block 1



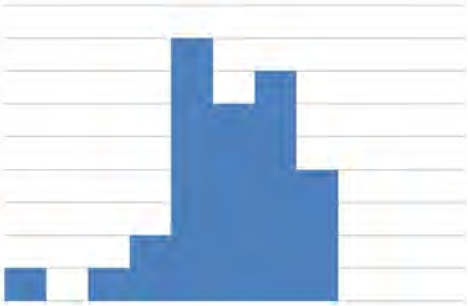
Block 2



Block 3



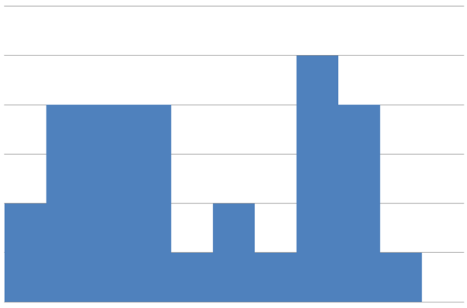
Block 4



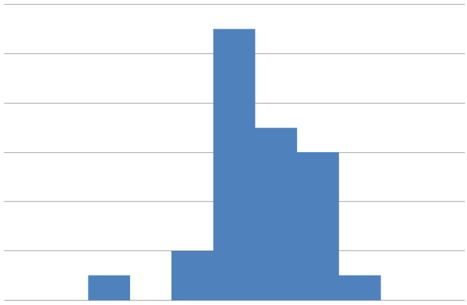
Experiment 3 (x-coordinates)

Block 1

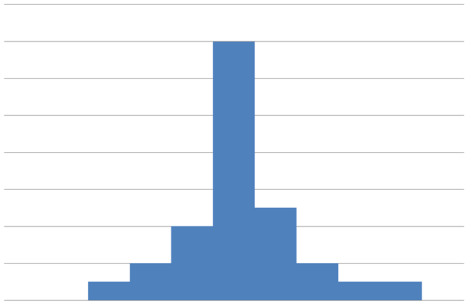
Block 2



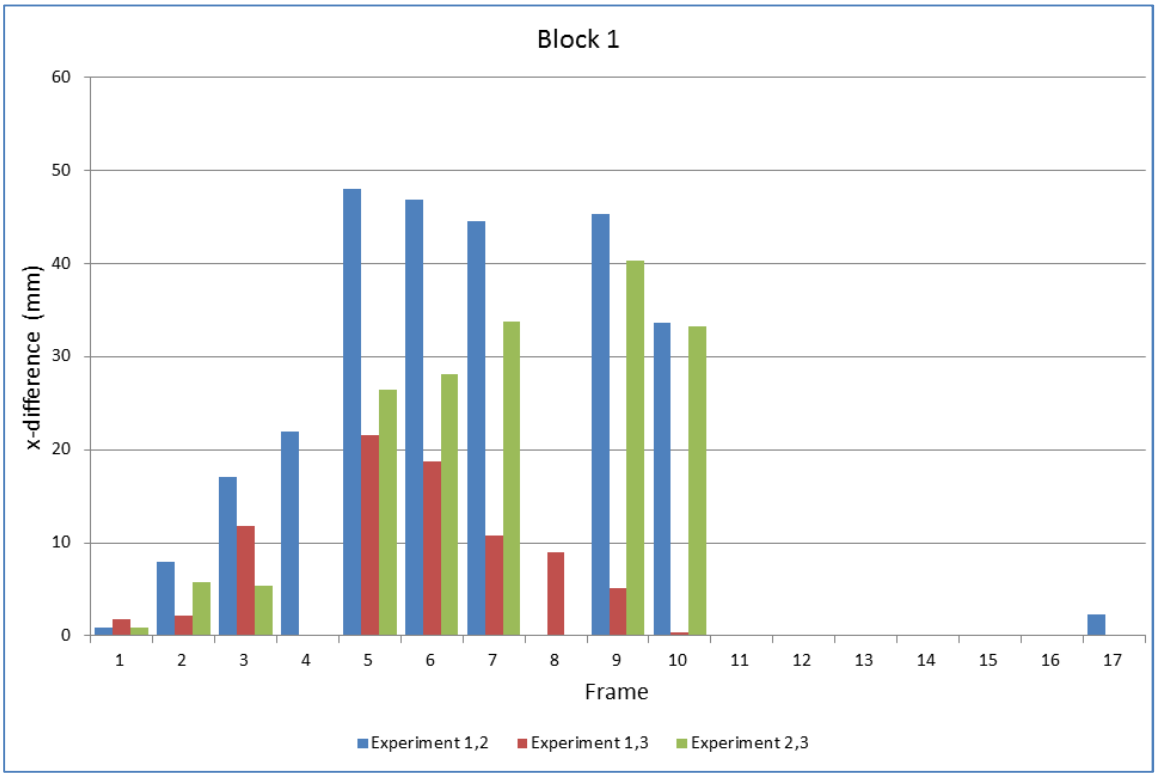
Block 3

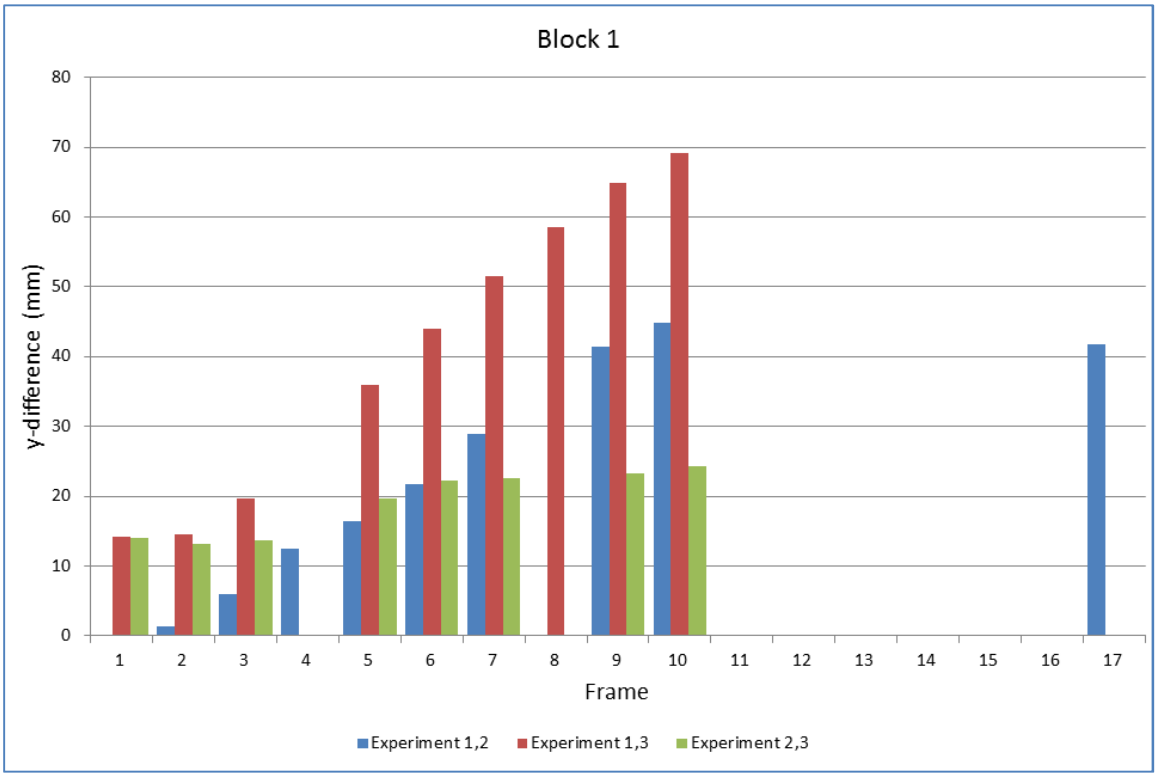


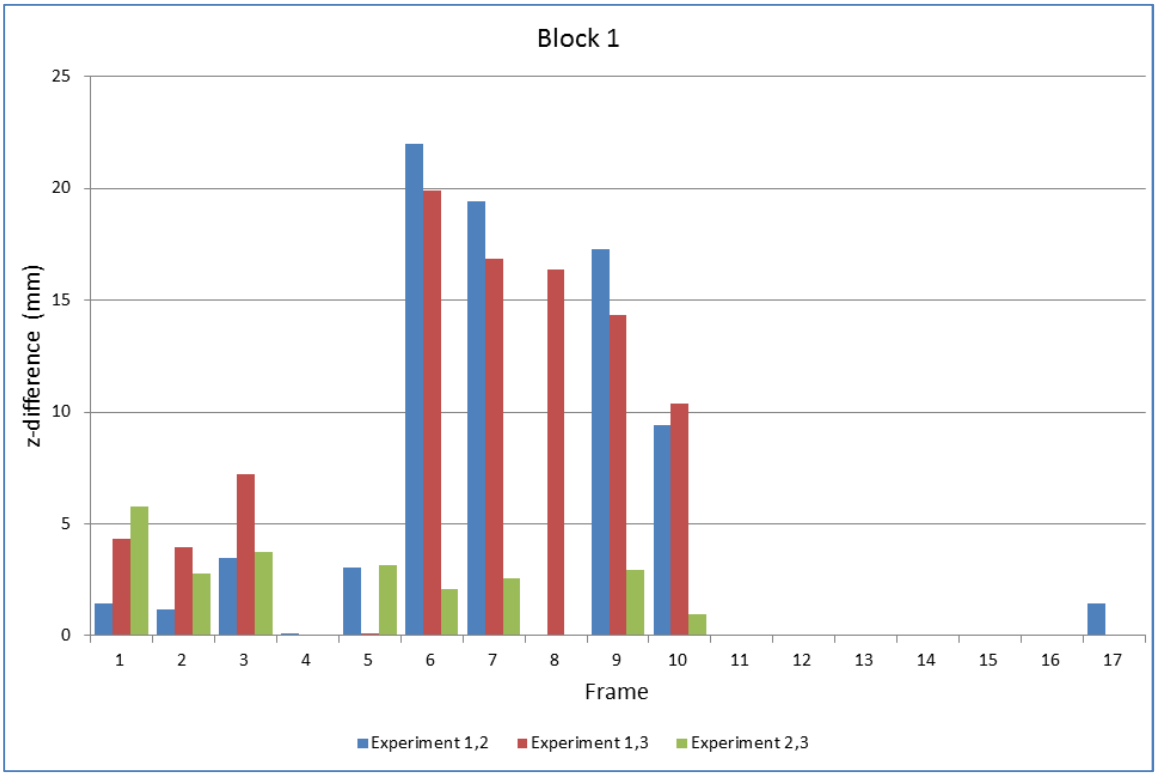
Block 4

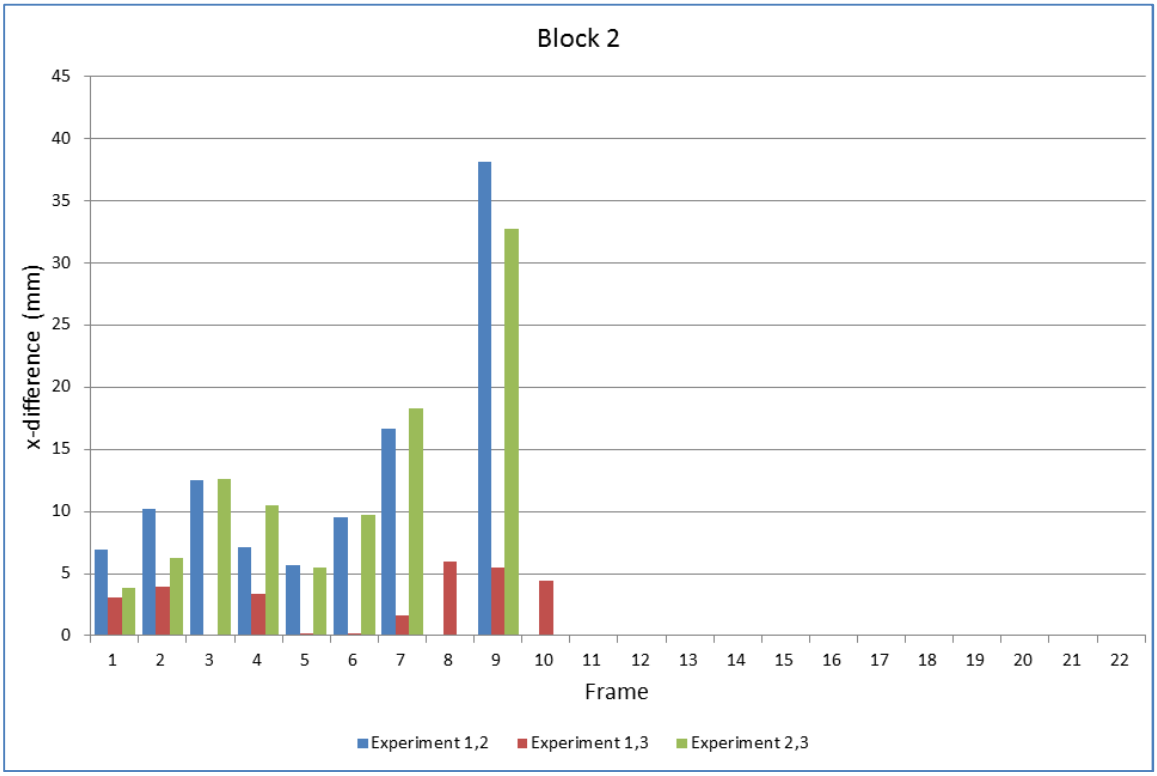


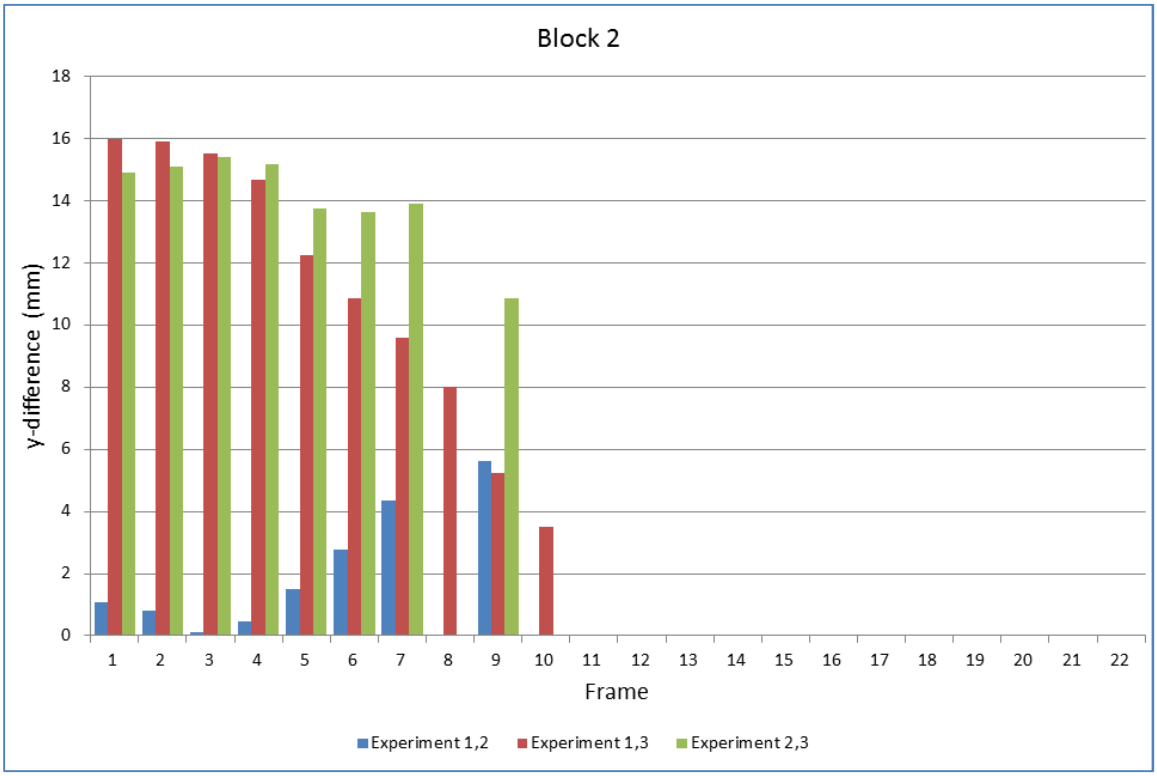
13 APPENDIX F: Statistical Dispersion Graphs

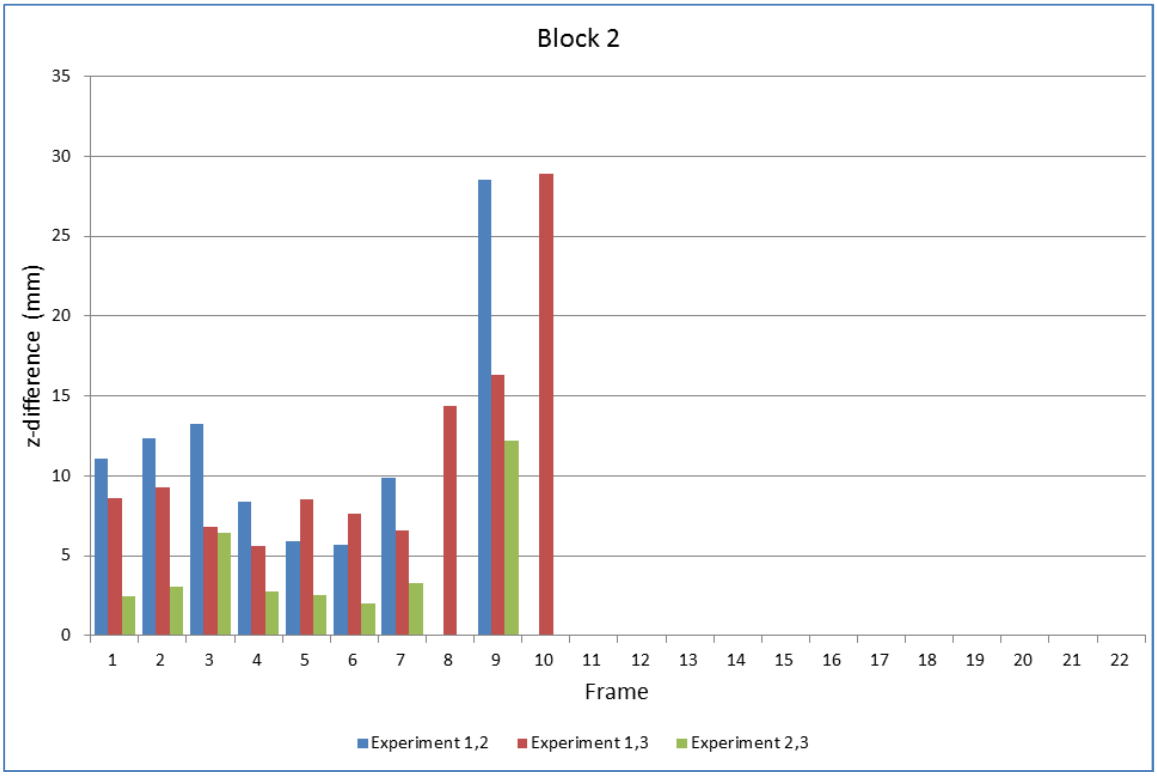


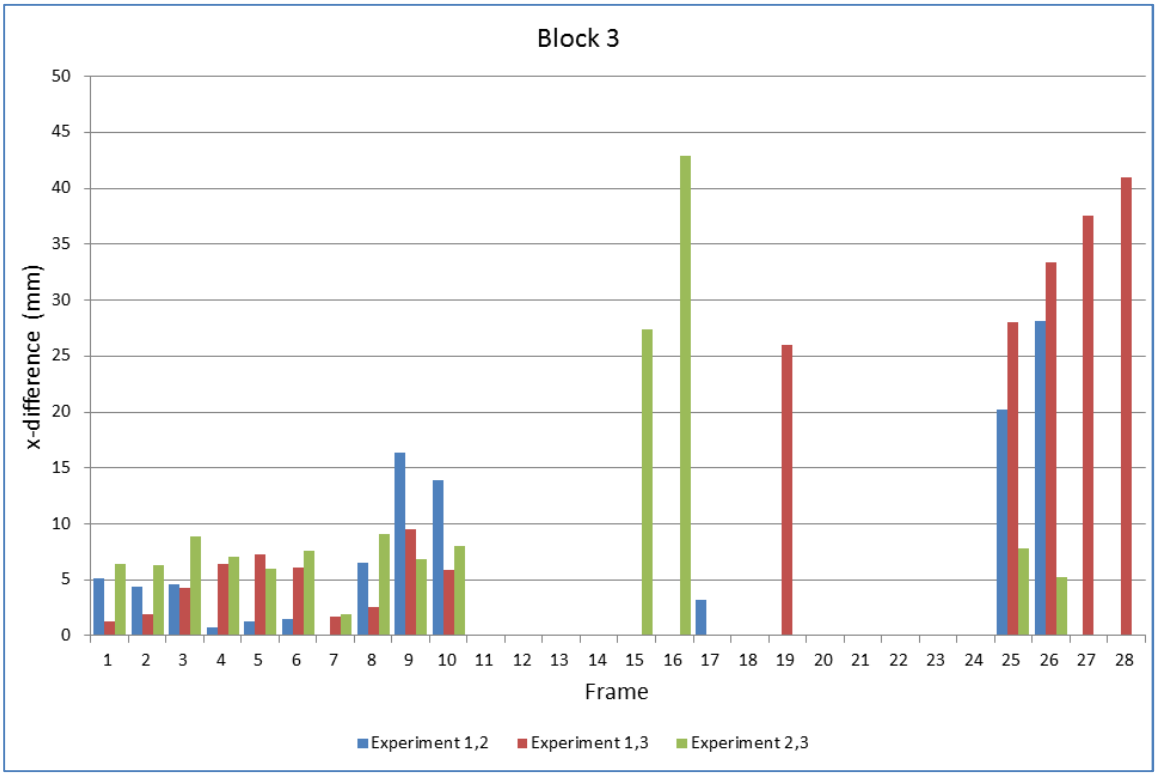


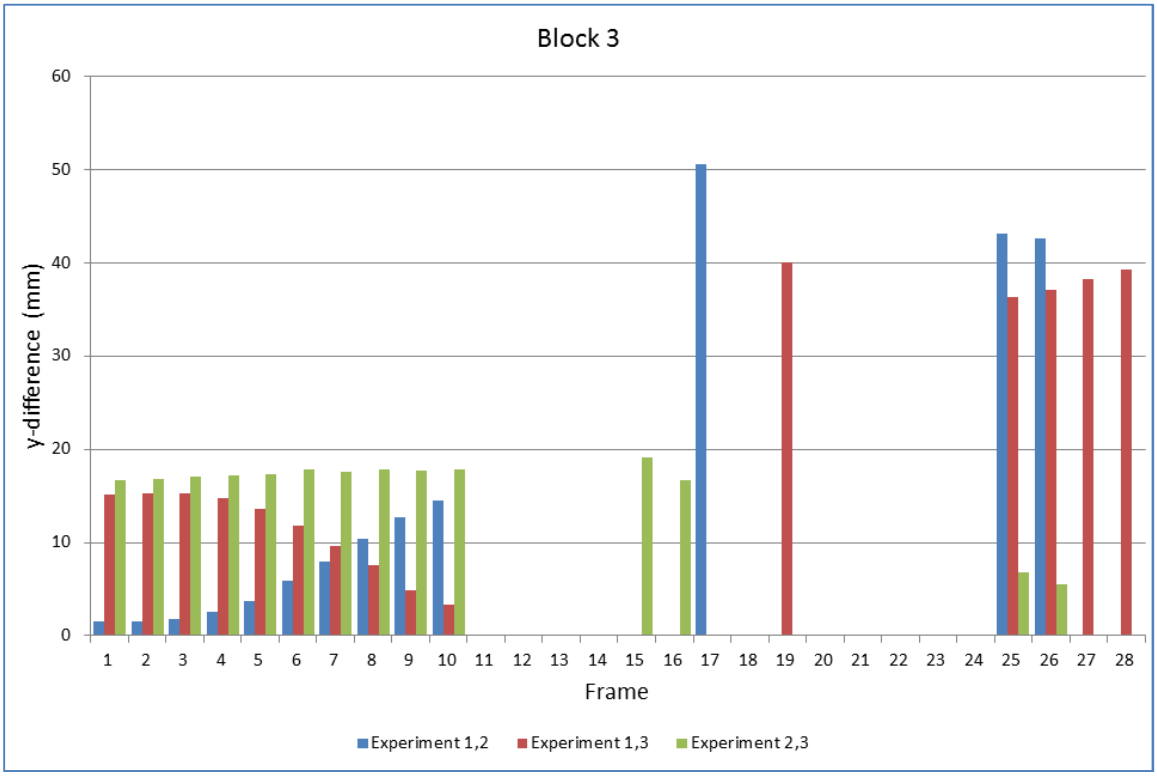

Block 1
z-difference (mm)
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3

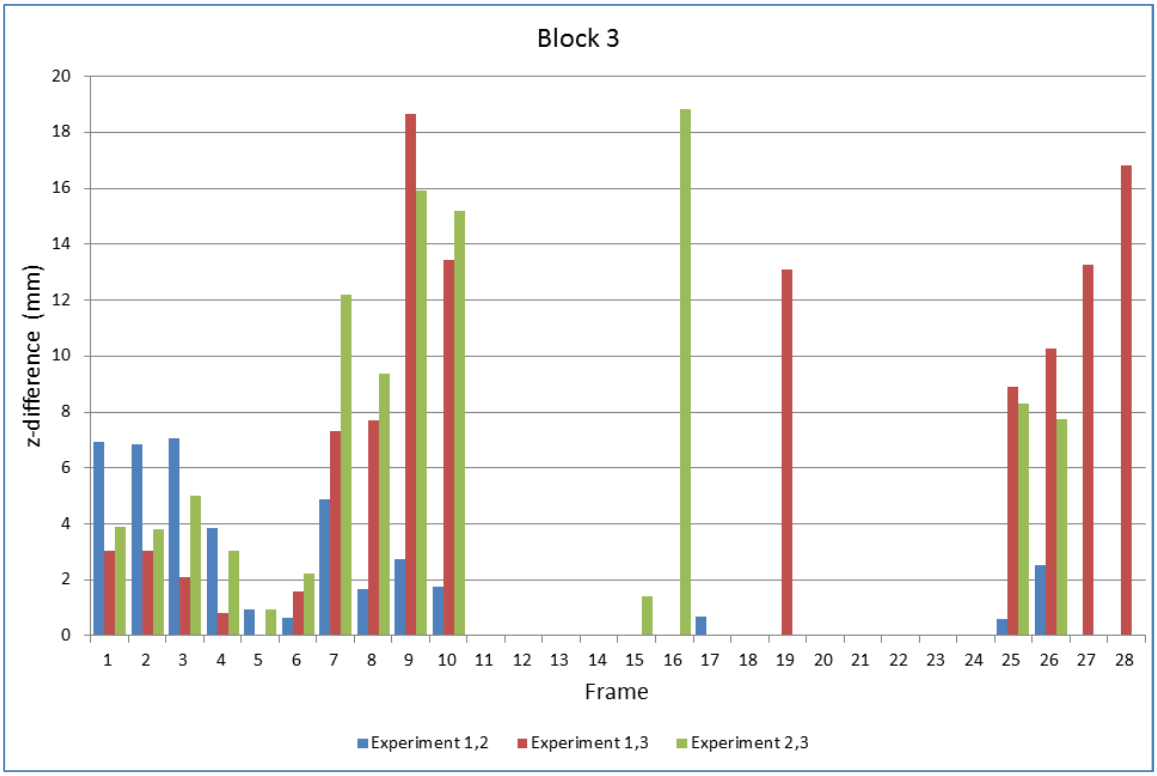

Block 2
x-difference (mm)
45
40
35
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3

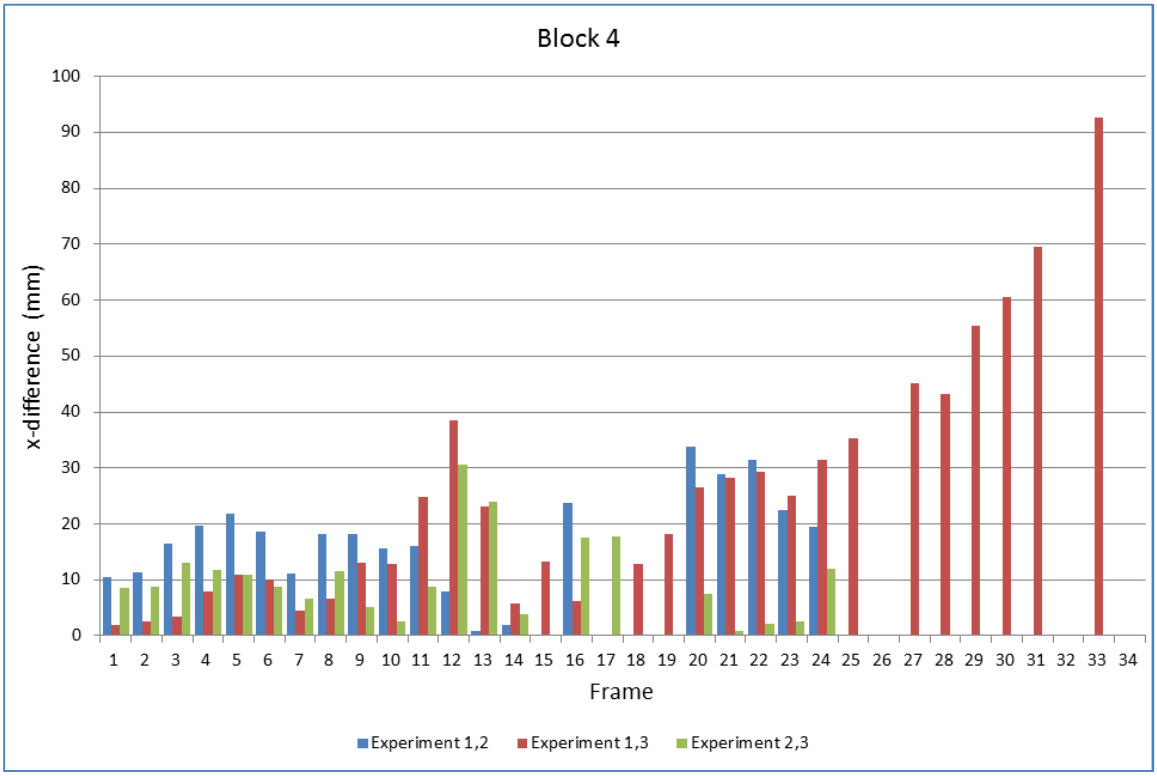

Block 2
y-difference (mm)
0
2
4
6
8
10
12
14
16
18
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3

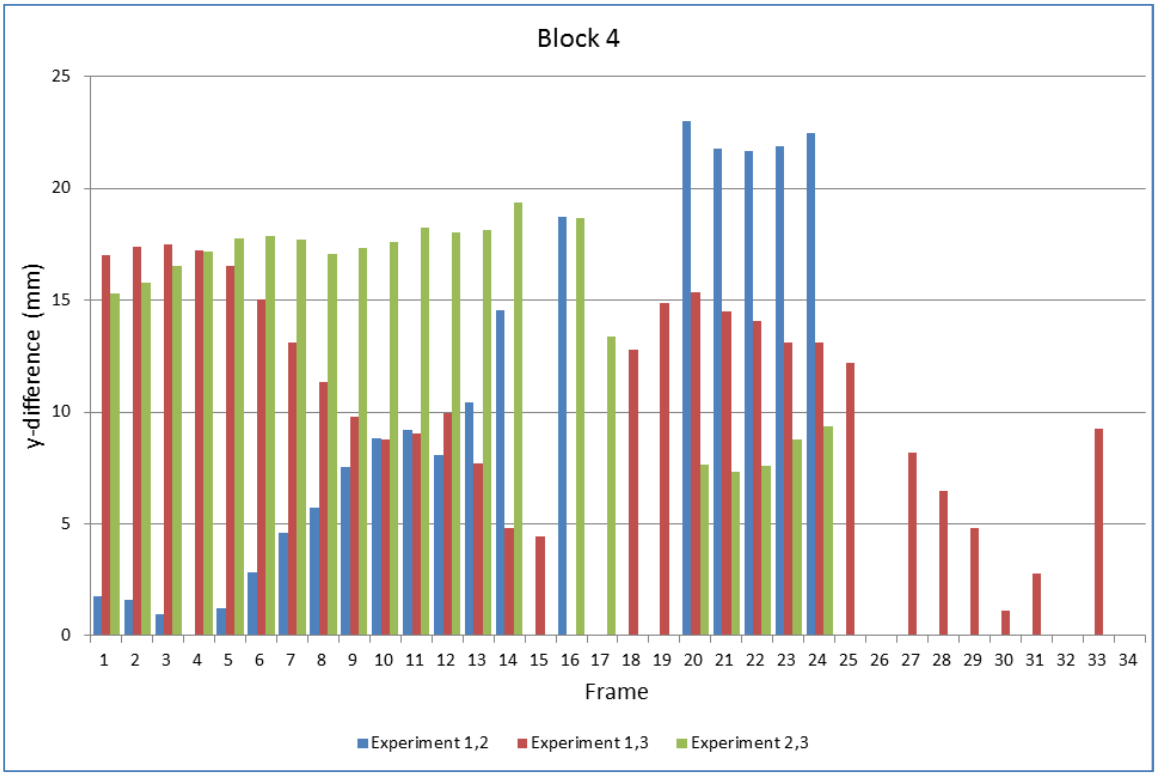

Block 2
z-difference (mm)
0
5
10
15
20
25
30
35
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3

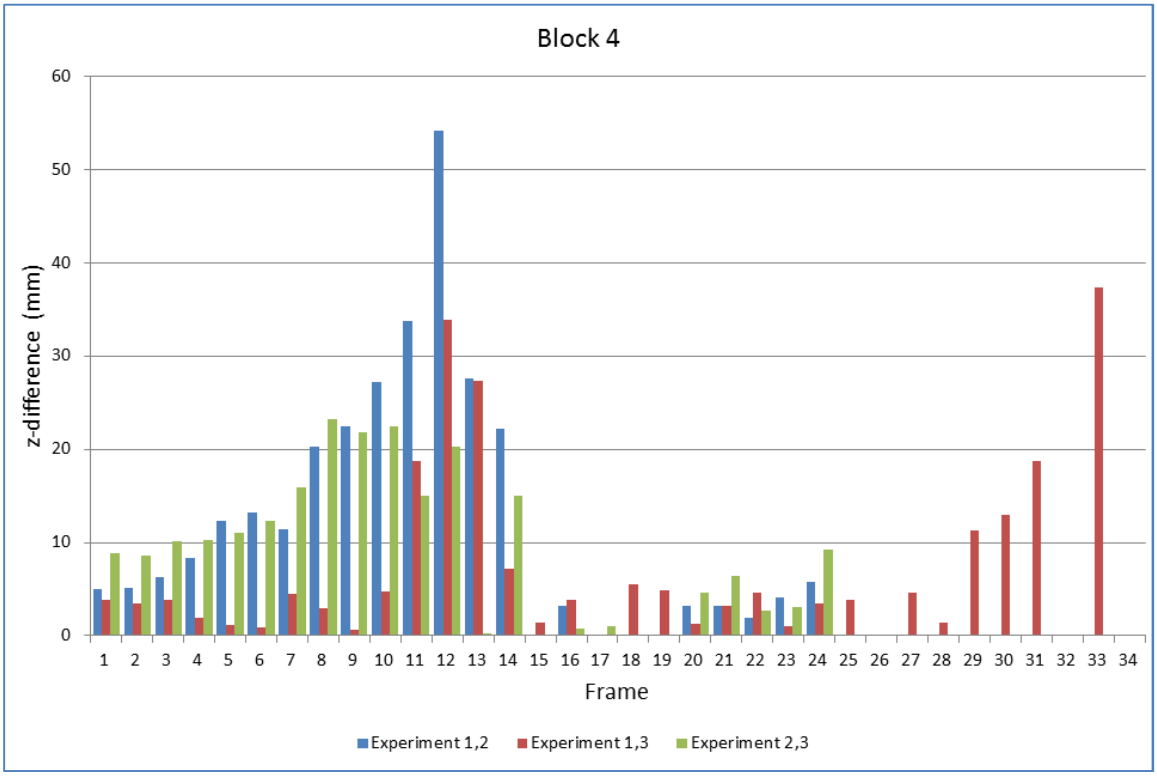

Block 3
x-difference (mm)
0
5
10
15
20
25
30
35
40
45
50
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3


Block 3
y-difference (mm)
0
10
20
30
40
50
60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3


Block 3
z-difference (mm)
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3


Block 4
x-difference (mm)
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3


Block 4
y-difference (mm)
0
5
10
15
20
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3


Block 4
z-difference (mm)
0
10
20
30
40
50
60
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
Frame
Experiment 1,2
Experiment 1,3
Experiment 2,3

14 APPENDIX G: Technical Data of Star-CCM+ Tutorial

Variable	**Value**
Maximum inner iterations	5
Advancing direction of waves	Positive x-direction
Vertical direction (used in wave model)	Positive z-direction
Current (used in wave model)	2.5 m/s in the positive x-direction
Wind speed (used in wave model)	2.5 m/s in the positive x-direction
Wave height	0.2 m
Wave length	6 m
Water depth (used in wave model)	10 m
Total simulation time	0.0 to 2.0 s
Release time	0.5 s
Ramp time	0.0 s
Time step	0.01 s
Moments of inertia (half model)	1000 kg-m^2 in all directions
Body mass (half model)	400 kg
6DOF motion solver	DFBI[1] Rotation and Translation
Degrees of freedom	2 (z-axis translation and y-axis rotation)
Turbulence model	k-epsilon
Free surface method	VOF
Air density	Constant
Time stepping method	Implicit unsteady

[1] Dynamic Fluid Body Interaction

15 APPENDIX H: Technical Data of Flow Channel Simulation

Variable	**Value**
Length of flow domain	9 m
Height of flow domain	1 m
Width of flow domain	0.75 m
Approximate mesh size	275022 cells
Distance from upstream boundary to platform	4 m
Number of blocks	4
Initial arrangement of blocks	Stacked vertically on top of one another
Block dimensions	0.050 m x 0.050 m x 0.050 m
Block density	631 kg/m^3
Coefficient of restitution (all materials)	0.5
Static friction coefficient (cubes; platform)	(0.3; 0.1)
Dynamic friction coefficient (cubes; platform)	(0.015; 0.0)
Method of coefficient combination	Averaging
Initial water column length	0.300 m
Initial water column height	0.200 m
Initial water column width	0.75 m (width of flow channel)
Time step	0.0025 s
Turbulence model	None (laminar flow assumed)
Free surface method	VOF
Air density	Constant (1.18415 kg/m^3)
Time stepping method	Implicit unsteady
Maximum inner iterations	5