

Numerical Modeling of Tsunami-Induced Hydrodynamic Forces on Free-Standing Structures using the SPH Method

Philippe St-Germain

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Civil Engineering

Under the auspices of the Ottawa-Carleton Institute for Civil Engineering

Academic advisors: Dr. Ioan Nistor and Dr. Ronald Townsend



University of Ottawa
Ottawa, Ontario, Canada
August 2012

© Philippe St-Germain, Ottawa, Canada 2012

Abstract

Tsunamis are among the most terrifying and complex physical phenomena potentially affecting almost all coastal regions of the Earth. Tsunami waves propagate in the ocean over thousands of kilometres away from their generating source at considerable speeds. Among several other tsunamis that occurred during the past decade, the 2004 Indian Ocean Tsunami and the 2011 Tohoku Tsunami in Japan, considered to be the deadliest and costliest natural disasters in the history of mankind, respectively, have hit wide stretches of densely populated coastal areas. During these major events, severe destruction of inland structures resulted from the action of extreme hydrodynamic forces induced by tsunami flooding. Subsequent field surveys in which researchers from the University of Ottawa participated ultimately revealed that, in contrast to seismic forces, such hydrodynamic forces are not taken into proper consideration when designing buildings for tsunami prone areas. In view of these limitations, a novel interdisciplinary hydraulic-structural engineering research program was initiated at the University of Ottawa, in cooperation with the Canadian Hydraulic Centre of the National Research Council, to help develop guidelines for the sound design of nearshore structures located in such areas.

The present study aims to simulate the physical laboratory experiments performed within the aforementioned research program using a single-phase three-dimensional weakly compressible Smoothed Particle Hydrodynamics (SPH) numerical model. These experiments consist in the violent impact of rapidly advancing tsunami-like hydraulic bores with individual slender structural elements. Such bores are emulated based on the classic dam-break problem. The quantitatively compared measurements include the time-history of the net base horizontal force and of the pressure distribution acting on columns of square and circular cross-sections, as well as flow characteristics such as bore-front velocity and water surface elevation. Good agreement was obtained. Results show that the magnitude and duration of the impulsive force at initial bore impact depend on the degree of entrapped air in the bore-front. The latter was found to increase considerably if

the bed of the experimental flume is covered with a thin water layer of even just a few millimetres.

In order to avoid large fluctuations in the pressure field and to obtain accurate simulations of the hydrodynamic forces, a Riemann solver-based formulation of the SPH method is utilized. However, this formulation induces excessive numerical diffusion, as sudden and large water surface deformations, such as splashing at initial bore impact, are less accurately reproduced. To investigate this particular issue, the small-scale physical experiment of Kleefsman *et al.* (2005) is also considered and modeled.

Lastly, taking full advantage of the validated numerical model to better understand the underlying flow dynamics, the influence of the experimental test geometry and of the bed condition (i.e. dry vs. wet) is investigated. Numerical results show that when a bore propagates over a wet bed, its front is both deeper and steeper and it also has a lower velocity compared to when it propagates over a dry bed. These differences significantly affect the pressure distributions and resulting hydrodynamic forces acting on impacted structures.

Remerciements

Premièrement, je tiens à remercier mon directeur de thèse Dr. Nistor, envers lequel je suis exceptionnellement reconnaissant. Grâce à ces innombrables encouragements et à son appréciation, celui-ci m’a toujours motivé à me surpasser. Depuis notre connaissance, il a toujours fait tout ce qui était en son pouvoir pour me pousser à des accomplissements que jamais je n’aurais pu envisager. Sa générosité m’a permis de vivre des expériences inoubliables pour lesquelles je ne pourrai assez le remercier. Tout ce que j’ai réalisé jusqu’à ce jour est en grande partie grâce à lui. Je remercie également mon codirecteur de thèse Dr. Townsend, qui m’a aussi largement supporté et qui, avec sa patience et ses connaissances en hydraulique, a grandement amélioré le contenu de cette thèse.

Je remercie tout le personnel de l’institut de recherche *IHCantabria* qui a fait de ma visite à l’Université de Cantabrie une expérience mémorable. Tout particulièrement Dr. Lomónaco Tonda qui, lors de nos nombreuses rencontres informelles au café du campus, a si gentiment et clairement partagé avec moi son savoir sur les vagues océaniques. D’ailleurs, je tiens à remercier Christian Klinghammer, technicien de laboratoire, qui a toujours été disponible pour répondre à mes questions; Iñaki Silanes Cristóbal, informaticien de génie, sans lequel je n’aurais jamais pu obtenir autant de bon résultats numériques; et Nabil Kakeh Burgada qui, avec son amitié et son sens de l’humour infatigable, a fait de mon séjour en Espagne une expérience des plus plaisantes.

J’éprouve une grande gratitude à l’égard de Dr. Shibayama, qui m’a si admirablement reçu à l’Université de Waseda. Je lui suis grandement reconnaissant pour le temps qu’il m’a accordé et le savoir sur les tsunamis qu’il a partagé avec moi et ce, comme si j’étais l’un de ses propres étudiants. Parmi ceux-ci, je tiens à remercier Sekine Yuki, Ohira Koichiro, Mikami Takahito et Ohtani Akira pour non seulement leur aide, mais surtout leur amitié. Je tiens particulièrement à remercier Takabatake Tomoyuki ainsi que Yasufumi Takayama qui, dès mon arrivée, m’ont fait découvrir le Japon avec une bonté incomparable.

Je n'aurais jamais pu concentrer autant d'efforts à mes études sans l'amour et le support inconditionnel de ma mère. En revanche, mes efforts auraient été vains sans la patience et la discipline au travail que mon père m'a transmises. Je remercie sincèrement mes parents à qui je dois tout.

Tous mes nombreux efforts ne sont cependant pas sans conséquences. C'est pourquoi je tiens à m'excuser devant tous mes amis et mes proches pour le temps que je n'ai pas passé avec eux.

Finalement, j'aimerais souligner tout particulièrement la compréhension et les sacrifices de ma douce complice Stéphanie, de loin ma particule préférée parmi toutes celles discutées dans cette thèse. Sans elle à mes côtés pour m'encourager et me supporter durant les moments difficiles, je n'aurais pu réaliser mes études avec tant de succès. Avec sa remarquable confiance envers moi et ses nombreux conseils, elle m'a permis de prendre conscience de mon potentiel et il est certain que sans elle, je ne serais pas la même personne aujourd'hui.

Acknowledgments

First, I would like to express my deep and sincere gratitude to my supervisor Dr. Nistor. With his endless encouragements and appreciation, he has always motivated me to reach higher goals. Since we ever met, he always did all that was in his power to help me succeed both academically and professionally. His exceptional kindness allowed me to experience once-in-a-lifetime opportunities for which I will ever be indebted. All that I have realised so far is in most part because of him. I also acknowledge the significant support of my co-supervisor Dr. Townsend. He has always been present at my presentations to help me improve, and with his patience and significant knowledge on hydraulics, has significantly increased the quality of this thesis.

I would like to thank all the personnel at the *IHCantabria* research institute that made my visit at the University of Cantabria such a memorable experience. Particularly Dr. Lomónaco Tonda who, during our numerous informal meetings over a good cup of coffee, shared with me a wealth of knowledge on ocean waves. Furthermore, I would like to thank Christian Klinghammer, laboratory technician who was always available to answer my questions, Iñaki Silanes Cristóbal, computer wizard whose help played a significant role in obtaining the numerical results presented in this thesis, and Nabil Kakeh Burgada, who with his friendship and remarkable sense of humour, has made my journey in Spain such a pleasant one.

I am enormously recognizant towards Dr. Shibayama at the Waseda University for being such outstanding host during my visit there. Although extremely busy with the media coverage associated with the first year anniversary of the great 2011 Tohoku Tsunami, he always took the time to sit down with me to share his valuable knowledge on the subject. He really made me feel as one of his own. Thanks are also due to Sekine Yuki, Ohira Koichiro, Mikami Takahito, Ohtani Akira, Yamao Sho, and all other members of the *Shibayama Lab* for not only their help, but more importantly their friendship. I am especially thankful towards Takabatake Tomoyuki as well as Yasufumi

Takayama, who as soon as I landed in Japan, took great care of me and showed me so much.

I never could have concentrated that many efforts in my undergraduate and graduate studies without the unconditional love and support of my mother. However, such efforts would have been in vain without the patience, discipline, and strong work ethics my dad passed down to me. I am forever thankful to my parents, to whom I owe everything.

Nevertheless, all my efforts did not come without consequences. This is why I would like to ask forgiveness to all of my friends and loved ones for the time I did not spend with them.

Last but not least, I thank my love Stéphanie for her unfailing comprehension and all the sacrifices she made for me. From all of the particles discussed in this thesis, she is by far my favourite one. Without her at my side encouraging me and supporting me during difficult times, I never could have completed my studies as well as I did. With and her remarkable confidence in me and with her never-ending advices, she made me realize my potential and it is certain that I would not be the same person without her today.

Agradecimientos

Primero, quiero expresar mi más sentido agradecimiento a mi supervisor, el Dr. Nistor. Con su interminable apoyo y aprecio, siempre me ha motivado a superarme. Desde que nos conocemos, él ha siempre hecho todo lo que estaba en su mano para ayudarme a alcanzar el éxito tanto académico como profesional. Su excepcional amabilidad me ha permitido vivir experiencias únicas en la vida por las que siempre estaré agradecido. Hasta ahora, la mayoría de mis éxitos son, en la mayor parte, gracias a él. También agradezco el gran apoyo de mi co-supervisor, el Dr. Townsend. Siempre ha estado presente en mis presentaciones para ayudarme a mejorar, y con su paciencia y sus enormes conocimientos hidráulicos, ha aumentado considerablemente la calidad de esta tesis.

Quiero agradecer a todo el personal del instituto de hidráulica ambiental *IHCantabria* que ha hecho de mi visita a la Universidad de Cantabria una experiencia memorable. En particular al Dr. Lomónaco Tonda que, durante nuestras reuniones informales en el café del campus, me ha enseñado muchísimas cosas sobre las olas oceánicas. Además, quiero agradecer a: Christian Klinghammer, técnico de laboratorio que siempre ha estado dispuesto a responder mis preguntas; Iñaki Silanes Cristóbal, virtuoso informático sin quien hubiera sido imposible obtener resultados numéricos tan buenos; y Nabil Kakeh Burgada que, con su amistad y su formidable sentido del humor, ha hecho que mi verano en España fuera increíblemente agradable.

Tengo una gratitud gigantesca hacia el Dr. Shibayama de la Universidad de Waseda por la hospitalidad tan buena que me ofreció durante mi visita allí. Incluso estando extremadamente ocupado debido a la cobertura mediática asociada con el primer aniversario del gran 2011 Tohoku tsunami, siempre ha encontrado tiempo para sentarse frente a mí a compartir su conocimiento sobre el tema. Realmente, me ha hecho sentir como alumno suyo. Muchísimas gracias también a Sekine Yuki, Ohira Koichiro, Mikami Takahito, Ohtani Akira, Yamao Sho y a todos los otros miembros del *Shibayama Lab* no

sólo por su ayuda, sino más importante, por su amistad. Especialmente, debo mucho a Takabatake Tomoyuki y a Yasufumi Takayama por descubrirme tantas cosas sobre Japón y hacer que mi viaje fuera inolvidable.

Yo no habría podido realizar tantos esfuerzos en mis estudios sin el apoyo y el amor incondicional de mi madre. Sin embargo, estos esfuerzos habrían sido en vano sin las nociones de paciencia, disciplina y trabajo duro, transmitidas por mi padre. Por tanto, siempre estaré agradecido sinceramente mis padres, a quienes debo todo.

No obstante, no todos mis esfuerzos quedan sin consecuencias. Es por eso que quiero disculparme con mis amigos y parientes por el tiempo que no he pasado con ellos.

Finalmente, quiero recalcar especialmente la gran comprensión y todos los sacrificios que mi vida Stéphanie a hecho por mí. Sin duda, de todas las partículas discutidas en esta tesis, ella es de lejos mi preferida. Sin ella a mi lado para alentarme y apoyarme durante los momentos difíciles, no habría podido completar con éxito mis estudios. Con su remarcable confianza en mí y sus consejos interminables, me ha permitido darme cuenta de mi potencial y de seguro que sin ella no sería la misma persona que soy a día de hoy.

謝辞

まず始めに、私の指導教員であるDr.Nistorに心からの感謝の意を表したいと思います。先生の温かい励ましがあったからこそ、こうして高い目標を達成することができました。お会いしてからずっと、先生は学術的な面と社会的な面の両方において私が成果を出せるようにと、全力を尽くしてくださいました。先生の多大なお力添えがあったからこそ、私は一生に一度しかできないであろう経験をすることができました。先生にさせていただいたことは、これから先もずっと忘れることはありません。ここまで私が出してきた成果は、先生がいなければ決して成し得なかったものばかりです。また、副指導教員Dr.Townsendにも大変お世話になりました。私がプレゼンテーションを行う際にはいつもご出席頂き、ご指導を頂きました。先生の深い知識のおかげで、本論文の質を飛躍的に向上させることができたと思っています。

Cantabria大学の訪問を忘れることのできないものとしてくれたIHCantabria研究所の全ての職員の方々にも、大変感謝しています。特に何どもコーヒーを飲みながら話し合いをさせて頂いたDr. Lomónaco Tonda は海岸工学の知識の塊のような方で、大変貴重な時間を一緒に過ごさせて頂きました。そして、研究所の技術者であり、いつも私の質問に答えてくださった Christian Klinghammer 氏、本論文の数値解析結果を得る際に大変お世話になった「コンピュータの魔術師」ことIñaki Silanes Cristóbal 氏と出会えたことで、スペインへの旅はかけがいのないものになりました。

早稲田大学の柴山先生には、私が日本滞在中に大変お世話になりました。2011年に起きた東北地方太平洋沖津波から一年が経ち、大変お忙しい中でも、先生はいつも私のために時間を割いて下さり、私の研究に知恵を貸してくださいました。1ヶ月強の短い間でしたが、先生は自分の生徒と同様に私のことを扱って

くださいました。関根佑貴君、大平幸一郎君、三上貴仁君、大谷彬君、山尾祥君、そして他の全ての柴山研究室のメンバーに、研究仲間としてだけでなく、友人としても感謝しています。そして特に高畠知行君、高山靖史君には、大変お世話になりました。日本に着いてすぐ、寮に連れて行ってくださっただけでなく、日本にある様々なものを私に紹介してくれました。

学部生から大学院生に至るまで、私がここまで努力し続けることができたのは、母の絶え間のない愛と支えのおかげです。しかし、忍耐の大切さや、自分を律すること、仕事に対する真摯な姿勢といった父の教えがなければ、その努力も意味のないものになっていただろうと思います。私を支え続けてくれた両親に深く感謝しています。

ただもちろん、ここまでたどり着くまでに犠牲にしてしまったものもありました。だからこそ、この場を借りて、時として一緒に時間を過ごすことができなかったことを、大切な友人達に謝りたいと思います。

最後になりますが、愛するStephanieの私に対する理解と私のために費やしてくれた時間、労力に対して感謝の言葉を捧げたいと思います。私にとって、この論文に書かれているあらゆる事象よりも、理論よりも、発見よりも、彼女は大切な存在です。苦しい時、彼女がそばにいて支えてくれなければ、決してこのような研究成果を残すことはできませんでした。彼女が私をずっと信じ続けてくれたこと、そしていつも温かい励ましをかけてくれたこと、彼女がいたからこそ、私は自分の可能性を信じることができました。彼女がいなければ、今の私はありません。これはどんな理論よりも明らかなことであると、私は思っています。

Table of Contents

Abstract.....	i
Remerciements	iii
Acknowledgments	v
Agradecimientos.....	vii
謝辭.....	ix
Table of Contents	xi
List of Figures.....	xvi
List of Tables	xxv
List of Symbols	xxvi
Chapter 1. Introduction	1
1.1. Objective	1
1.2. Scope	6
1.3. Contributions	7
1.4. Outline.....	8
Chapter 2. The SPH Method	10
2.1. SPH in Perspective	10
2.1.1 Numerical Simulations	10
2.1.2 Grid-Based Methods.....	11
2.1.3 Mesh-free Methods.....	12

2.2.	<i>Essential Formulation of SPH</i>	15
2.2.1	Function Approximation.....	15
2.2.2	Particle Approximation.....	20
2.2.3	Approximation Errors	23
2.2.4	Derivation of SPH Formulation for Fluid Motion	26
2.3.	<i>Numerical Aspects of SPH</i>	32
2.3.1	Smoothing Kernels	33
2.3.2	Compressibility	35
2.3.3	Viscosity and Turbulence	36
2.3.4	Time-Stepping	38
2.3.5	Correction Schemes	42
2.3.6	Boundary Condition.....	47
2.3.7	Execution Flow Chart	54
Chapter 3.	Literature review	56
3.1.	<i>Experimental Modeling of Tsunami Bores and Dam-Break Waves</i>	56
3.2.	<i>Influence of Bed Condition on Dam-Break Flow Characteristics</i>	61
3.3.	<i>Physical Modeling of Bore-Structure Interaction</i>	63
3.4.	<i>Numerical Modeling of Bore-Structure Interaction using SPH</i>	70
3.5.	<i>Conclusion</i>	77
Chapter 4.	Description of Physical and Numerical Experiments	78
4.1.	<i>Physical Models</i>	78
4.1.1	Yeh and Petroff [n.d.]	78
4.1.2	Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012]	79
4.1.3	Kleefsman <i>et al.</i> [2005]	84
4.2.	<i>Calibration of SPH-related Parameters</i>	84
4.2.1	Preliminary selection of parameter and simulation options.....	86
4.2.2	Particle Spacing	87
4.2.3	Slope Limiter	89
4.2.4	Smoothing Length	90

4.2.5	Speed of Sound.....	92
4.2.6	Additional Parameters	94
4.2.7	Selected Values.....	95
4.3.	<i>Computational Domains</i>	95
4.3.1	Yeh and Petroff [n.d.]	96
4.3.2	Nouri et al. [2010] and Al-Faesly et al. [2012].....	97
4.3.3	Kleefsman et al. [2005].....	102
Chapter 5. Post-Processing of Numerical Outputs.....		104
5.1.	<i>Repulsive Force of Boundary Particles</i>	104
5.2.	<i>Particle Sampling</i>	106
5.2.1	Resultant Force on Column	108
5.2.2	Water Surface Elevation	110
5.3.	<i>Performance and Adjustments</i>	111
5.3.1	Repulsive Force of Boundary Particles	111
5.3.2	Particle Sampling.....	115
5.4.	<i>Discussion</i>	125
Chapter 6. Comparison of Numerical with Physical Results		127
6.1.	<i>Experiments of Nouri et al. [2010] and Al-Faesly et al. [2012]</i>	127
6.1.1	Water Surface Elevation	128
6.1.2	Bore-Front Velocity.....	144
6.1.3	Pressure.....	146
6.1.4	Force	159
6.2.	<i>Experiment of Kleefsman et al. [2005]</i>	164
6.2.1	Water Surface Elevation	165
6.2.2	Pressure.....	168
6.3.	<i>Discussion</i>	174
Chapter 7. Results of Numerical Experiments		176
7.1.	<i>Influence of Reservoir Geometry</i>	176

7.1.1	Runup and Force.....	177
7.1.2	Water Surface Elevation and Velocity.....	177
7.1.3	Cross-Waves Formation	184
7.1.4	Discussion.....	186
7.2.	<i>Influence of Channel Width</i>	<i>186</i>
7.2.1	Runup and Force.....	186
7.2.2	Water Surface Elevation and Velocity.....	187
7.2.3	Bore-Structure Interaction	192
7.2.4	Discussion.....	199
7.3.	<i>Influence of Reservoir Size.....</i>	<i>201</i>
7.3.1	Runup and force.....	203
7.3.2	Water Surface Elevation and Velocity.....	204
7.3.3	Steady-State Flow Condition.....	207
7.3.4	Discussion.....	209
7.4.	<i>Influence of Bed Condition</i>	<i>210</i>
7.4.1	Bore Propagation	211
7.4.2	Bore Runup and Velocity Field	216
7.4.3	Pressure.....	223
7.4.4	Force	228
7.4.5	Discussion.....	230
Chapter 8. Conclusions and Recommendations		233
8.1.	<i>Conclusions.....</i>	<i>233</i>
8.2.	<i>Recommendations for Future Work.....</i>	<i>236</i>
References.....		238
Publications		248
Appendix A	Riemann Solver Description	249
Appendix B	Additional Results of Sensitivity Analysis	257
Appendix C	Post-Processing Code.....	261

Appendix D	Additional Comparison of Numerical with Physical Results.....	314
-------------------	--	------------

List of Figures

Figure 1.1. Hydraulic bore induced by the 2011 Tohoku Tsunami, Iwanuma city, Miyagi prefecture, Japan [Kyodo News, 2012].	2
Figure 1.2. Destruction of coastal structures by the 2011 Tohoku Tsunami: (a) Overturned caissons of the breakwater at Kamaishi city, Miyagi prefecture [Takahashi <i>et al.</i> , 2011]; (b) Concrete panels stripped from earthen sea dike at Tarou city, Miyagi prefecture [EERI, 2011].	3
Figure 1.3. Widespread destruction of inland structures by the 2011 Tohoku Tsunami in Onagawa city, Miyagi prefecture [Courtesy of Mikami Takahito].	3
Figure 1.4. Destruction of inland structures due to tsunami-induced hydrodynamic forces: (a) 2004 Indian Ocean Tsunami, Phuket city, Thailand [CAEE, 2005]; (b) 2011 Tohoku Tsunami, Onagawa city, Miyagi prefecture, Japan [Courtesy of Dr. Ioan Nistor].	4
Figure 1.5. Subduction zones around the Pacific Ocean and locations of several great tsunamigenic earthquakes (adapted from Atwater <i>et al.</i> [2005]).	5
Figure 2.1. Conceptualized computational nodes of (a) grid-based methods and (b) mesh-free methods.	13
Figure 2.2. Influence domain Ω with radius $2h$ of smoothing kernel W . r_{ij} is the distance separating particles i and j .	21
Figure 2.3. Cubic spline and quadratic smoothing kernels.	33
Figure 2.4. Absolute value of the first derivative of the cubic spline and quadratic smoothing kernels.	34
Figure 2.5. Truncated influence domain Ω with radius $2h$ of smoothing kernel W .	42
Figure 2.6. Central repulsive force $\mathbf{f}_{ik}$ on fluid particle i due to boundary particle k .	49
Figure 2.7. Normal repulsive force $\mathbf{f}_{ik}$ on fluid particle i due to boundary particle k .	50
Figure 2.8. Wall friction force $\mathbf{f}_{ik,wall\ visc}$ on fluid particle i due to boundary particle k .	53
Figure 2.9. Spatial arrangement of dynamic boundary particles where Δ is the initial inter-particle spacing (based on Gómez-Gesteira <i>et al.</i> [2010a]).	54
Figure 2.10. General execution flow chart of SPHysics code.	55
Figure 3.1. Types of inland intrusions of a tsunami [adapted from Takahashi <i>et al.</i> , 2012].	57
Figure 3.2. Schematic of the dam-break considered for the generation of tsunami-like bores.	59
Figure 3.3. Definition sketch for the propagation of dam-break waves according to the shallow water equations: (a) wet-bed condition; (b) dry-bed condition.	60
Figure 3.4. Influence of initial downstream water depth h_d on the bore-front velocity U (based on Leal <i>et al.</i> [2006]).	62

Figure 3.5. Comparison of experimental and analytical [Stoker, 1957] bore profiles for a nearly dry-bed condition (from Stansby <i>et al.</i> [1998] with $h_u = 0.100\text{m}$ and $h_d \approx 0.002\text{m}$).	64
Figure 3.6. Comparison of experimental and analytical [Stoker, 1957] bore profiles for a wet-bed condition (from Stansby <i>et al.</i> [1998] with $h_u = 0.100\text{m}$ and $h_d = 0.010\text{m}$).	65
Figure 3.7. Bore propagating on a perfectly dry bed ($h_u = 0.300\text{m}$ and $h_d = 0$): (a) and (b) 0 to 0.30m downstream of the gate; (c) to (f) 0.90 to 1.20m downstream of the gate (adapted from Lauber and Hager [1998], time T1 is not specified).	66
Figure 3.8. Bore propagating on a nearly dry bed ($h_u = 0.360\text{m}$ and $h_d \approx 0.002\text{m}$): (a) to (d) just downstream of the gate; (e) and (f) further downstream of the gate (adapted from Stansby <i>et al.</i> [1998], the actual distances where the images were taken is not specified).	67
Figure 3.9. Typical time-history of the pressure acting on structures due to the impact of a breaking tsunami wave (adapted from Arikawa [2011]).	68
Figure 3.10. Comparison of the experimental (a) bore profile, (b) runup, and (c) pressure head at an elevation of 0.018m above the channel's bottom on a vertical wall due to bores ($h_u = 0.5\text{m}$) propagating on dry ($h_d = 0$) and wet ($h_d = 0.003\text{m}$) beds. γ is the unit weight of water (adapted from Ramsden [1993]).	68
Figure 3.11. Comparison of the experimental (a) pressure at an elevation of 0.02m above the channel's bottom, and (b) net force acting streamwise on a square column due to bores ($h_u = 0.55\text{m}$) propagating on dry ($h_d = 0$) and wet ($h_d = 0.005\text{m}$) beds (experimental results of Al-Faesly <i>et al.</i> [2012]).	69
Figure 3.12. Experimental time-history of the net force acting in the streamwise direction on a square column due to bores generated by various upstream water depths h_u (adapted from Árnason [2005])	70
Figure 3.13. Comparison between experimental (circles) and numerical (solid line) time-history of the resulting force on a column (from Gómez-Gesteira and Dalrymple [2004]).	71
Figure 3.14. Comparison of bore profile for the (a) dry and (b) wet bed conditions moments before initial impact (from Gómez-Gesteira and Dalrymple [2004]).	71
Figure 3.15. Vertical distribution of the maximum force acting on the upstream face of a column for dry (crosses) and wet (dots) bed conditions (from Gómez-Gesteira and Dalrymple [2004]).	72
Figure 3.16. Numerical flow sequence of the experiment of Yeh and Petroff [n.d.] with $h_u = 0.30\text{m}$ and $h_d = 0.01\text{m}$ (adapted from Silvester and Cleary [2006]).	73
Figure 3.17. Comparison between experimental and numerical time-history of the resulting force on a column using both a repulsive and a dynamic boundary condition (from Silvester and Cleary [2006])	74
Figure 3.18. Experimental and numerical time-history of the pressure exerted on the upstream face of a small obstacle at an elevation of 0.021m above the experimental tank's bottom. The experimental and VOF results were obtained by Kleefsman <i>et al.</i> [2005] (from Lee <i>et al.</i> [2010]).	75

Figure 3.19. Comparison of the water splash 0.56 seconds after gate opening predicted by (a) WCSPH and (b) ISPH models (from Lee <i>et al.</i> [2010]).	75
Figure 3.20. Experimental and numerical time-history of the pressure exerted on upstream face of a small obstacle at an elevation of 0.021m above the experimental tank's bottom. The experimental results were obtained by Kleefsman <i>et al.</i> [2005] (from Crespo <i>et al.</i> [2011]).	76
Figure 3.21. Comparison of the (a) numerical and (b) experimental water splash 0.56 seconds after the opening of the gate. The experimental image was taken by Kleefsman <i>et al.</i> [2005] (from Crespo <i>et al.</i> [2011]).	76
Figure 4.1. Plan view of the experimental setup of Yeh and Petroff [n.d.]	79
Figure 4.2. Plan view of the experimental setup of Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012] shown with a square column.	80
Figure 4.3. Picture of the experimental setup of Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012] with a full reservoir and looking downstream (courtesy of Taofiq Al-Faesly).	80
Figure 4.4. Partially dry-bed condition observed is some of the experimental runs.	82
Figure 4.5. Schematic of experimental column model (square) and instrumentation used by Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012].	84
Figure 4.6. Plan view of the experimental setup of Kleefsman <i>et al.</i> [2005].	85
Figure 4.7. Test structure considered in the experiment of Kleefsman <i>et al.</i> [2005].	85
Figure 4.8. Time-history of the net base horizontal force acting on the column for various initial inter-particle spacing (Δ).	88
Figure 4.9. Time-history of the net base horizontal force acting on the column for various slope limiters (β).	90
Figure 4.10. Time-history of the net base horizontal force acting on the column for various smoothing lengths (h).	92
Figure 4.11. Time-history of the net base horizontal force acting on the column for various speed of sound at the reference water density (c_0).	93
Figure 4.12. Initial configuration of computational domain for simulating the experiment of Yeh and Petroff [n.d.].	97
Figure 4.13. . Initial configuration of computational domain for simulating the experiments of Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012] shown with a square column.	99
Figure 4.14. Initial configuration of computational domain for simulating the general experiment based on Al-Faesly <i>et al.</i> [2012] shown with a dry bed.	101
Figure 4.15. Initial configuration of computational domain for the experiments with wall test structure based on Al-Faesly <i>et al.</i> [2012] shown with a dry bed.	102
Figure 4.16. Initial configuration of computational domain for simulating the experiment of Kleefsman <i>et al.</i> [2005].	103

Figure 5.1. Total force per unit mass $\mathbf{f}_k$ on a boundary particle k due to neighbouring fluid particles (normal components shown only).	105
Figure 5.2. Plan view of circular column surrounded with spherical sampling volumes of radius $2h$	109
Figure 5.3. Force contribution dF of a single sampling volume acting on tributary area dA	110
Figure 5.4. Schematic of virtual wave gage for the determination of the water surface elevation.	111
Figure 5.5. Comparison of numerical and experimental time-history of the net force acting on the column obtained through the summation of boundary repulsive forces: (a) with different initial inter-particle spacings (simulations D1 and D2); (b) with various speed of sound reduction factor (κ) for simulation D1.	112
Figure 5.6. Numerical and experimental time-history of the wave-induced horizontal force acting on a caisson breakwater obtained through the summation of boundary repulsive forces by Rogers <i>et al.</i> [2010] with an initial particle spacing (Δ) of (a) 0.010m, and (b) 0.005m.	113
Figure 5.7. Region of unrealistic pressures and void developing beyond initial impact upstream of the square column in a plane close the flume's bottom (simulation D2-85).	116
Figure 5.8. Region of unrealistic pressures and void developing beyond initial impact upstream of the circular column in a plane close the flume's bottom (simulation D2-85-C).	117
Figure 5.9. Region of unrealistic pressures and void developing beyond initial impact upstream of the square column in a plane close the flume's bottom for a simulation with a higher resolution (simulation D3-HIGH).	118
Figure 5.10. Hydrostatic pressure distribution of fluid particles acting on the upstream wall of the water reservoir before the arrival of the negative dam-break wave for simulation D2-85.	119
Figure 5.11. Cross-section along the X-Z plane passing through the column's center showing the arrangement of sampling volumes for the determination of the pressure time-history: (a) with no spacing; (b) with a spacing of 0.06m from the column's edge (black lines represent solid boundaries).	120
Figure 5.12. Numerical pressure time-histories at transducers P5 to P45 obtained with no spacing (a), and spacings of (b) 0.03m, (c) 0.06m, and (d) 0.12m for simulation D2-85-C.	121
Figure 5.13. Numerical time-history of the pressure at corresponding transducers derived from the absolute maximums of the time-histories obtained with various spacings for simulation D2-85-C.	122
Figure 5.14. Pressure time-histories at transducer P5 for simulation D2-85-C obtain using various smoothing lengths (h).	123
Figure 5.15. Numerical time-history of the resulting force acting on the column for simulation D1 obtained by pressure integration for various distances separating the sampling volumes from the column's surface.	124
Figure 5.16. Numerical and experimental time-history of the net force acting on a square column for simulation D1.	125

Figure 6.1. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, (d) W4, (e) W5, (f) W6, (g) W7, and (h) W8 for $h_u = 0.85\text{m}$ and without a column (experimental run E-85-N and simulation D2-85-N).	129
Figure 6.2. Cross-waves occurring upstream of column in physical experiments (experiment E-85-C). ...	130
Figure 6.3. Qualitative comparison of the numerical (left, simulation D2-85) and experimental (right, Nouri <i>et al.</i> [2010]) water surface (a, b) 0.50, (c, d) 2.00, and (e, f) 4.00 seconds after gate opening ($h_u = 0.85\text{m}$). (g) and (h) shows the out-of-phase upstream propagation of an obstruction-induced wave at 6.25 and 7.50 seconds, respectively.	132
Figure 6.4. Obstruction-induced wave propagating upstream in the physical experiments ($h_u = 0.85\text{m}$, experimental run E-85-C).	134
Figure 6.5. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W7, (c) W8, (d) W9, (e) W11, and (f) W13 for $h_u = 0.55\text{m}$ (experimental run E-55-WET and simulation D2-55).....	135
Figure 6.6. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W7, (c) W8, (d) W9, (e) W11, and (f) W13 for $h_u = 0.85\text{m}$ (experimental run E-85 and simulation D2-85).	136
Figure 6.7. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, (d) W7, (e) W8, (f) W9, (g) W11, and (h) W13 for $h_u = 0.85\text{m}$ and with the circular column (experimental run E-85-C and simulation D2-85-C).....	137
Figure 6.8. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W8, (c) W9, (d) W11, and (e) W13 for $h_u = 1.15\text{m}$ (experimental run E-115 and simulation D2-115).	138
Figure 6.9. Qualitative comparison of numerical (left, simulation D2-115) and experimental (right, Nouri <i>et al.</i> [2010]) water surface profiles (a, b) 1.50, (c, d) 3.00, (e, f) 4.00, and (g, h) 7.00 seconds after gate opening. Experimental $h_u = 1.00\text{m}$ and numerical $h_u = 1.15\text{m}$	141
Figure 6.10. Numerical time-history of the water surface elevation at all wave gauges for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).	144
Figure 6.11. Experimental still frames (plan view) of the bore-front at the location of the column ($h_u = 0.85\text{m}$ and $h_d \approx 0.005\text{m}$, time T1 is unknown).	145
Figure 6.12. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.55\text{m}$ and experimental $h_d \approx 0$ (experimental run E-55-DRY and simulation D2-55).	147
Figure 6.13. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.55\text{m}$ and experimental $h_d \approx 0.005\text{m}$ (experiment E-55-WET and simulation D2-55).	148

Figure 6.14. Experimental still frames of the bore-front propagating over a “virtually” dry bed ($h_d \approx 0$) captured with the high-speed video camera ($h_u = 0.55\text{m}$).....	150
Figure 6.15. Experimental still frames of the bore-front propagating over a wet bed ($h_d \approx 0.005\text{m}$) captured with the high-speed video camera ($h_u = 0.55\text{m}$).....	150
Figure 6.16. Comparison of numerical time-history of pressures acting on the column at an elevation of 0.05m above the flume’s bottom with different sampling frequencies for $h_u = 0.85\text{m}$ (simulation D2-85).....	151
Figure 6.17. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume’s bottom for $h_u = 0.85\text{m}$ (experimental run E-85 and simulation D2-85).....	153
Figure 6.18. Comparison of numerical and experimental time-history of pressures acting on the circular column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume’s bottom for $h_u = 0.85\text{m}$ (experimental run E-85-C and simulation D2-85-C).....	154
Figure 6.19. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume’s bottom for $h_u = 1.15\text{m}$ (experimental run E-115 and simulation D2-115).....	155
Figure 6.20. Comparison of the numerical time-history of the pressure acting on the column at an elevation of 0.05m above the flume’s bottom obtained using the SPH method (simulation D2-115) and FVM with VOF ($h_u = 1.15\text{m}$ and $h_d = 0$). FVM results courtesy of Zhihui Ma.	157
Figure 6.21. Numerical time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume’s bottom for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).....	158
Figure 6.22. Comparison of numerical and experimental time-history of the net base horizontal force acting on the column for impoundment depths (h_u) of (a) 0.55m with a virtually dry bed ($h_d \approx 0$), (b) 0.55m with a wet bed ($h_d \approx 0.005\text{m}$), (c) 0.85m, and (d) 1.15m.	159
Figure 6.23. Still frames of the physical experiments [Nouri <i>et al.</i> , 2010] with impoundment depths (h_u) of (a) 0.50m, (b) 0.75m, (c) 1.00m showing the varying degree of turbulence and air entrainment at times corresponding to approximately 2 seconds after the runup force is reached.....	160
Figure 6.24. Comparison of numerical and experimental time-history of the net base horizontal force acting on the circular column for $h_u = 0.85\text{m}$. $h_d = 0$ for the numerical simulation performed using the SPH method (D2-85-C) while $h_d = 0.001\text{m}$ for the one performed using FVM with VOF. FVM results courtesy of Zhihui Ma.....	161
Figure 6.25. Comparison of the experimental rise time of the net base horizontal force acting on the column due to the impact of bores ($h_u = 0.55\text{m}$) propagating on dry ($h_d \approx 0$) and wet ($h_d \approx 0.005\text{m}$) beds (experimental runs E-55-DRY and E-55-WET).	162

Figure 6.26. Comparison of the experimental and numerical rise time of the net base horizontal force acting on the column due to the impact of a bore ($h_u = 0.55\text{m}$) propagating on a dry bed (experimental $h_d \approx 0$; experimental run E-55-DRY and simulation D2-55).....	163
Figure 6.27. Numerical time-history of the net base horizontal force acting on the column for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).....	164
Figure 6.28. Comparison to the numerical water splash at initial wave impact for inter-particle spacings Δ of (a) 0.0350, (b) 0.0200, and (c) 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.	166
Figure 6.29. Numerical time-history of the water surface elevation at wave gauges (a) W1, (b) W2, (c) W3, and (d) W4 for inter-particle spacings Δ of 0.0350, 0.0200, and 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.....	167
Figure 6.30. Comparison of numerical (simulation D5-HIGH) and experimental (Kleefsman <i>et al.</i> [2005]) time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, and (d) W4.	168
Figure 6.31. Qualitative comparison of numerical (left, simulation D5-HIGH) and experimental (right, Kleefsman <i>et al.</i> [2005]) water surface profiles.	171
Figure 6.32. Numerical time-history of the pressure at all pressure transducers for inter-particle spacings Δ of 0.0350, 0.0200, and 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.	172
Figure 6.33. Comparison of numerical (simulation D5-HIGH) and experimental (Kleefsman <i>et al.</i> [2005]) time-history of the pressure at all pressure transducers.	173
Figure 7.1. Time-history of (a) the bore runoff and (b) the net base horizontal force acting streamwise on the column for straight (simulation D3-L2) and bend (simulation D2-85) reservoirs.....	177
Figure 7.2. Water-surface profile along the centerline of the bended (simulation D2-85) and straight (simulation D3-L2) reservoirs (a), 1.00, (b) 2.00, (c) 2.88, (d) 5.00, and (e) 8.00 seconds after gate opening.....	178
Figure 7.3. Plan view of velocity field inside the bended reservoir (a) 1.00, (b) 2.00, (c) 2.88, and (d) 5.00 seconds after gate opening (simulation D2-85).....	181
Figure 7.4. Time-history of the depth-averaged flow velocity (a) -3.00m, (b) -0.25m, and (c) 3.27m from the gate for straight (simulation D3-L2) and bended (simulation D2-85) reservoirs.	182
Figure 7.5. Time-history of the discharge calculated 0.25m upstream from the gate for straight (simulation D3-L2) and bended (simulation D2-85) reservoirs	183
Figure 7.6. Effect of additional lateral inflow on the formation of diagonal cross-waves at the entrance of the test section ($h_u = 0.85\text{m}$). Numerical results obtained using FVM with VOF courtesy of Zhihui Ma.	185
Figure 7.7. Time-history of (a) the bore runoff and (b) the net base horizontal force acting streamwise on the column for channel widths (W) of 1.3 (simulation D3-L1) and 2.6m (simulation D3-WIDE)....	187

Figure 7.8. Cross-sectional view of bore-structure interaction just upstream of column (a, b) 2.00, (c, d) 3.00, (e, f) 3.75, (g, h) 4.50, and (i, j) 6.00 seconds after gate opening for channel widths (W) of 1.3 (left, simulation D3-L1) and 2.6m (right, simulation D3-WIDE).	189
Figure 7.9. Time-history of the water-surface elevations (left) and depth-averaged flow velocities (right) for channel widths (W) of 1.3 (simulation D3-L1) and 2.6m (simulation D3-WIDE) at locations (a, b) 1.77m, and (c, d) 3.77m from the gate.	190
Figure 7.10. Froude number 1.77 and 3.77m downstream of the gate for simulation D2-85.	190
Figure 7.11. Time-history of the water-surface elevation 1.77m downstream from the gate for numerical simulations (D3-L1 and D3-HIGH) with different initial inter-particle spacings Δ .	191
Figure 7.12. Cross-sectional view at $X = 4.92\text{m}$ demonstrating the reduction of the effective flow area caused by the void of fluid particles near solid boundaries for the initial inter-particle spacing Δ of (a) 0.035m (simulation D3-L1) and (b) 0.025m (simulation D3-HIGH).	192
Figure 7.13. Numerical time-history of the net base horizontal force acting on the column with points associated to Figure 7.14 (simulation D3-L1).	193
Figure 7.14. Velocity field upstream of column's center (a) 1.23, (b) 1.61, (c) 3.00, (d) 3.75, (e) 4.27, (f) 4.75, (g) 5.50, (h) 6.00, and (i) 7.00 seconds after gate opening for the channel width (W) of 1.3m (simulation D3-L1).	196
Figure 7.15. Still frames from the physical experiments of Árnason [2005] showing the interaction of a hydraulic bore ($h_u = 0.250\text{m}$) with a circular column.	196
Figure 7.16. Plan views of the X velocity field in a plane close to the channel's bottom at (a) 1.50, (b) 2.50, (c) 4.00, and (d) 6.00 seconds after gate opening for simulation D2-85.	198
Figure 7.17. Plan views of the Y velocity field in a plane close to the channel's bottom at (a) 1.50, (b) 2.50, (c) 4.00, and (d) 6.00 seconds after gate opening for simulation D2-85.	199
Figure 7.18. Plan view of velocity field around the column (a, b) 2.50 and (c, d) 3.00 seconds after gate opening for channel widths (W) of 1.3 (left, simulation D3-L1) and 2.6m (right, simulation D3-HIGH).	200
Figure 7.19. Time-history of (a) the bore runup and (b) the net base horizontal force acting streamwise on the column for reservoir lengths (L) of 6.0, 11.2 and 15.0m corresponding to simulations D3-L1, D3-L2, and D3-L3, respectively	203
Figure 7.20. Time-history of the water-surface elevations (left) and depth-averaged flow velocities (right) for reservoir lengths (L) of 6.0, 11.2 and 15.0m at locations of (a, b)	205
Figure 7.23. Difference in the water surface elevation at the front and back of the column at steady-state for $L = 15.0\text{m}$ (simulation D3-L3).	209
Figure 7.24. Comparison of numerical and analytical (SWE) bore profiles for the dry-bed condition (simulation D3-DRY).	212
Figure 7.25. Comparison of numerical and analytical (SWE) bore profiles for the wet-bed condition (simulation D3-WET).	213

Figure 7.26. Time-history of the bore-front velocity U for the dry- and wet-bed conditions until impact with structures (simulations D3-DRY and D3-WET).....	214
Figure 7.27. Side (top) and plan (bottom) views of bore-front for the dry- (left, simulation D3-DRY) and wet-bed (right, simulation D3-WET) conditions at a distance of 3.60m from the downstream end of the reservoir.	215
Figure 7.28. Comparison of bore-front profile for the dry- (simulation D3-DRY) and wet-bed (simulation D3-WET) conditions at a distance of 3.60m from the downstream end of the reservoir.....	216
Figure 7.29. Time-history of the bore runoff on the (a) column (simulations D3-DRY and D3-WET), and (b) wall for dry- and wet-conditions (simulations D4-DRY and D4-WET).	217
Figure 7.30. Side view of velocity field upstream of column's center for the dry-bed condition (a) 0.50 and (b) 1.00 seconds after initial impact (simulation D3-DRY).	218
Figure 7.31. Side view of velocity field upstream of column's center for the wet-bed condition at (a) 0.50 and (b) 1.00 seconds after initial impact (simulation D3-WET).	218
Figure 7.32. Plan view of velocity field around the column at times corresponding to 0.75 seconds after initial impact for (a) dry- and (b) wet-bed conditions corresponding to simulations D3-DRY and D3-WET, respectively.....	220
Figure 7.33. Side view of velocity field upstream of wall's center for the dry-bed condition (a) moments before impact, (b) at initial impact and at times corresponding to (c) 0.50, (d) 0.75, (e) 1.00, (f) 1.50 (g) 2.00, and (h) 2.50 seconds after initial impact (simulation D4-DRY).....	221
Figure 7.34. Side view of velocity field upstream of wall's center for the wet-bed condition (a) moments before impact, (b) at initial impact and at times corresponding to (c) 0.50, (d) 0.75, (e) 1.00, (f) 1.50 (g) 2.00, and (h) 2.50 seconds after initial impact (simulation D4-WET).	222
Figure 7.35. Time-history of pressures acting at the center of the upstream face of the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, and (f) 0.30m above the channel's bottom for the dry- (simulation D3-DRY) and wet-bed (simulation D3-WET) conditions.....	225
Figure 7.36. Time-history of pressures acting at the center of the wall at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, and (f) 0.30m above the channel's bottom for the dry- (simulation D4-DRY) and wet-bed (simulation D4-WET) conditions.	226
Figure 7.37. Plan views of the pressure field at initial impact with the column (left, simulations D3-DRY and D3-WET) and wall (right, simulations D4-DRY and D4-WET) for the dry- (top) and wet-bed (bottom) conditions.	227
Figure 7.38. Side views of the pressure field at times corresponding to 1.50 seconds after initial impact with the wall for the (a) dry- and (b) wet-bed conditions (simulations D4-DRY and D4-WET).....	228
Figure 7.39. Time-history of (a) the net base horizontal force acting on the column (simulations D3-DRY and D3-WET), and of (b) the net force per meter width of wall (simulations D4-DRY and D4-WET) for dry- and wet-bed conditions.	229

List of Tables

Table 4.1. Details of the experimental runs considered for quantitative comparison.	82
Table 4.2. SPH-related parameters selected for the base case of the sensitivity analysis.	87
Table 4.3. Total number of particles and computational times for various inter-particle spacings (Δ).	89
Table 4.4. Computational times for various slope limiters (β).	90
Table 4.5. Computational time for various smoothing lengths (h).	92
Table 4.6. Computational time for various speeds of sound at the reference water density (c_0).	94
Table 4.7. Final SPH-related parameters selected for numerical simulations	96
Table 4.8. Simulation details for the experiment of Yeh and Petroff [n.d.].	97
Table 4.9. Simulation details for the experiment of Nouri <i>et al.</i> [2010] and Al-Faesly <i>et al.</i> [2012].	99
Table 4.10. Simulation details for general experiments based on Al-Faesly <i>et al.</i> [2012].	101
Table 4.11. Simulation details for the experiment with wall test structure based on Al-Faesly <i>et al.</i> [2012]	102
Table 4.12. Simulation details for the experiment of Kleefsman <i>et al.</i> [2005].	103
Table 6.1. Comparison of numerical and experimental bore-front velocity.	145
Table 6.2. Numerical water density at initial impact and corresponding variation with respect to the reference density ($\rho_0 = 1000\text{kg/m}^3$).	156
Table 6.3. Numerical water density at the moment of maximum force due to runup and corresponding variation with respect to the reference density ($\rho_0 = 1000\text{kg/m}^3$).	157

List of Symbols

Roman letters

B	Number of neighbouring boundary particles in influence domain Ω
B_{object}	Number of boundary particles forming a solid object
c	Speed of sound in water (m/s)
c_0	Speed of sound in water at the reference density ρ_0 (m/s)
d	Still water level (m)
$f(\mathbf{x})$	Arbitrary scalar function at position $\mathbf{x}$
$\mathbf{f}(\mathbf{x})$	Arbitrary vector function at position $\mathbf{x}$
$\mathbf{f}_{ik}$	Force per unit mass on water particle i exerted by boundary particle k (N/kg)
$\mathbf{f}_{ki}$	Force per unit mass on boundary particle k exerted by boundary particle i (N/kg)
F_D	Form drag (N)
F_H	Resulting hydrostatic force (N)
F_X	Net base horizontal force acting streamwise on the column (N)
$\mathbf{F}$	Acceleration due to external forces in momentum equation (m/s^2 or N/kg)
$\mathbf{F}_{object}$	Resultant force vector acting on a solid object due to neighbouring water particles (N)
h	Smoothing length or water surface elevation (m)
h_u	Upstream water depth of dam-break problem or impoundment depth (m)
h_d	Downstream water depth of dam-break problem (m)
H	Wave height (m)
L	Length of straight dam-break reservoir (m)
m	Mass (kg)
M	Mach number

$\mathbf{n}$	Unit vector normal to the surface S or unit vector normal to a solid boundary at location of a boundary particle k
N	Number of neighbouring water particles in influence domain Ω
N_{volume}	Total number of water particles within a sampling volume v
P	Fluid pressure (Pa)
r_{ij}	Distance separating two particles i and j (m)
$\mathbf{r}_{ij}$	Vector separating two particles i and j (m)
S	Surface of influence domain Ω (m ²)
dS	Infinitesimal surface of influence domain Ω (m ²)
t	Time (seconds)
$\mathbf{t}$	Unit vector tangent to a solid boundary at location of boundary particle k
T_{sim}	Simulation time (seconds)
T_{comp}	Computational time (seconds)
u	X component of velocity vector $\mathbf{u}$ (m/s)
$\mathbf{u}$	Velocity vector (m/s)
U	Bore-front velocity (m/s)
v	Y component of velocity vector $\mathbf{u}$ or flow velocity (m/s)
v_a	Depth-averaged flow velocity (m/s)
ΔV	Finite volume (m ³)
w	Z component of velocity vector $\mathbf{u}$ (m/s)
W	Smoothing kernel (-) or flow width of test channel (m)
W_{ij}	Smoothing kernel evaluated for particles i and j
x	X component of position vector $\mathbf{x}$ (m)
$\mathbf{x}$	Position vector (m)
X	Distance from the gate positive downstream (m)
y	Y component of position vector $\mathbf{x}$ (m)
z	Z component of position vector $\mathbf{x}$ (m)

Greek letter symbols

α_{time}	Time step coefficient
α_{visc}	Artificial viscosity coefficient
α_{smooth}	Smoothing length coefficient
α_{sound}	Speed of sound coefficient
β	Slope limiter coefficient in MUSCL scheme
γ	Equation of state exponent
δ	Dirac delta function
Δ	Initial inter-particle spacing (m)
ε	XSPH coefficient
ν	Kinematic viscosity of laminar flow (m ² /s)
ν_{wall}	Wall viscosity (m ² /s)
ρ	Water density (kg/m ³)
ρ_0	Reference water density (kg/m ³)
τ	Shear stress tensor in SPS turbulence model (N/m ²)
Ω	Influence domain of smoothing kernel
$d\Omega$	Infinitesimal volume of influence domain Ω (m ³)

Mathematical operators

∇	Gradient
∇_i	Gradient taken with respect to the coordinates of particle i
$\nabla \cdot$	Divergence
∇^2	Laplace
$\cdot$	Vector dot product
$\otimes$	Tensor product
$ \ $	Absolute value
$\ \ \ $	Euclidean norm
D/Dt	Substantial derivative
$\partial/\partial a$	Partial derivative with respect to a variable a
$O(f(n))$	Order or “Big-Oh” of a function $f(x)$
$\langle f(x) \rangle$	Kernel approximation of a function $f(x)$

Subscripts

i	Represents a water particle i
j	Represents a water particle j
k	Represents a boundary particle k
v	Represents a sampling volume v

Chapter 1. Introduction

1.1. Objective

Large tsunami waves propagate in the ocean over thousands of kilometres away from their generating source at considerable speeds. They are among the most terrifying and complex physical phenomena potentially affecting almost all coastal regions of the Earth. Generally, when an offshore subduction earthquake of significant magnitude occurs, the quick uplift motion of the seabed vertically displaces a large volume of water, inducing a rapid, localized change in sea level. This results into one or several tsunami waves propagating in opposite directions, perpendicular to the fault line. In deep waters, these waves will have relatively small amplitudes (often less than 1m) but their wavelengths often approach hundreds of kilometres. As these waves propagate towards the shoreline into shallow coastal waters, due to the compressing effect of the up-sloping seabed and the decreasing water depth, waves experience deceleration and a reduction of their wavelength, as well as an increase in amplitude. In most cases, depending on the geographical features of the coastline, the front of the wave will break when its height becomes approximately equal to the local water depth. From this point, the broken wave will take the form of a hydraulic bore, a rapidly advancing body of water with a relatively uniform depth and a steep turbulent front, and continue to propagate onshore as seen in Figure 1.1. In addition, although most often triggered by subduction earthquakes as explained above, tsunamis can also be due to the sudden displacement of water induced by subaerial and submarine landslides, volcanic eruptions, and meteor impacts.

Within the past few years, although the phenomenon has been known for centuries in countries like Japan, several tsunamis around the globe brought to light the potential such events have for causing severe damage to inland buildings and endangering human life. Two events that stand out particularly among others are the 2004 Indian Ocean Tsunami and the 2011 Tohoku Tsunami, Japan. On December 26th of 2004, a Mw9.1 [USGS, 2012a] subduction earthquake off the northwest coast of Indonesia's Sumatra



Figure 1.1. Hydraulic bore induced by the 2011 Tohoku Tsunami, Iwanuma city, Miyagi prefecture, Japan [Kyodo News, 2012].

island triggered a tsunami that caused widespread destruction in Indonesia, Malaysia, Myanmar, Thailand, Bangladesh, Sri Lanka, India, and several East African countries. The death toll was estimated over 310,000 [CAEE, 2005]. In the second case, even though Japan had large coastal structures such as breakwaters and sea dikes specifically designed against tsunamis, these were no match for the unprecedented tsunami of March 11th 2011, as shown in Figure 1.2. Triggered by a Mw9.0 [USGS, 2012b] subduction earthquake located northeast of the Tohoku region, the tsunami resulted in widespread destruction (Figure 1.3), claimed over 20,000 lives, and caused economic losses estimated at over 217 billion US dollars [EERI, 2011]. In the aftermath of these events, as displayed in Figure 1.4, field surveys in which researchers from the University of Ottawa participated revealed that structures can severely be damaged or destroyed due to the action of extreme tsunami-induced hydrodynamic forces [Nistor *et al.*, 2005; Ghobarah *et al.*, 2006; Saatcioglu *et al.*, 2006; Tomita *et al.*, 2006; Yamamoto *et al.*, 2006; EERI, 2011]. Furthermore, after the latest 2011 Tohoku Tsunami is was observed that, although multiple-storey engineered concrete and steel buildings were previously believed to be tsunami-resistant [e.g. EERI, 2011; Takahashi *et al.*, 2011, Mikami *et al.*, 2012], as



(a)

(b)

Figure 1.2. Destruction of coastal structures by the 2011 Tohoku Tsunami: (a) Overturned caissons of the breakwater at Kamaishi city, Miyagi prefecture [Takahashi *et al.*, 2011]; (b) Concrete panels stripped from earthen sea dike at Tarou city, Miyagi prefecture [EERI, 2011].



Figure 1.3. Widespread destruction of inland structures by the 2011 Tohoku Tsunami in Onagawa city, Miyagi prefecture [Courtesy of Mikami Takahito].



Figure 1.4. Destruction of inland structures due to tsunami-induced hydrodynamic forces: (a) 2004 Indian Ocean Tsunami, Phuket city, Thailand [CAEE, 2005]; (b) 2011 Tohoku Tsunami, Onagawa city, Miyagi prefecture, Japan [Courtesy of Dr. Ioan Nistor].

evidence in Figure 1.3 and Figure 1.4-b, they can also be destroyed. Motivated by these field observations, an extensive literature review of current design codes demonstrated that, in contrast to seismic forces, such hydrodynamic forces are not taken into proper consideration when designing buildings for tsunami prone areas [Nistor *et al.*, 2009a].

From a Canadian perspective, history shows that the country's coastlines can also be affected by such catastrophic events. On November 18, 1929, a Mw7.2 earthquake occurred south of Newfoundland which triggered a submarine landslide that subsequently generated a tsunami that took 28 lives along the Burin peninsula of Newfoundland [Ruffman and Hann, 2006]. Another example is the tsunami that partially destroyed the town of Port Alberni, British-Columbia, during the night of March 28, 1964. This tsunami, which was generated by a Mw9.2 [USGS, 2011] subduction earthquake south of the Alaskan coast, reached Vancouver Island within approximately 4 hours. Although its waves were relatively small along the open coast of the island, the waves were significantly amplified at Port Alberni, by the fact that the town is located at the head of a contracting fjord valley. The damages resulting from this tsunami were estimated at 10 million dollars [White, 1966]. Even though the aforementioned events are of significance, what makes Canada particularly susceptible to potentially large tsunami waves is the proximity of the Cascadia subduction zone located just off the west coast of

Canada and of the United States (Figure 1.5). Recent geological studies in this area revealed that a *great* (magnitude 8 or higher) subduction earthquake of magnitude Mw9.0, followed by the generation of a large tsunami, occurred during the year of 1700 [Atwater *et al.*, 2005]. As tsunami waves can travel far away from their generating source, the findings of these studies were also supported by written Japanese records. These records indicate that a small tsunami (attenuated over the considerable distance it travelled) surprised the inhabitants of several villages, as they did not feel any preceding earthquake. It should be noted that, at this time the northwest of North America was not yet explored, consequently only the oral traditions of First Nations recall the event. One more event that further demonstrates the tsunami risk along the west coast of Canada is the 1960 Chile tsunami, generated by a Mw9.5 subduction earthquake, which travelled across the Pacific Ocean and took 138 lives in Japan [USGS, 2010]. This last example clearly shows that, depending on the orientation of the fault line, any tsunamigenic earthquake along the subduction zones surrounding the Pacific Ocean (also known as the “rim of fire”) can potentially affect the west coast of Canada.



Figure 1.5. Subduction zones around the Pacific Ocean and locations of several great tsunamigenic earthquakes (adapted from Atwater *et al.* [2005]).

Based on the damages observed following the 2011 Tohoku Tsunami (Figure 1.2), the extent of the protection provided by coastal structures, such as breakwaters and sea dikes is now questionable. On the other hand, properly designed vertical evacuation buildings could be the best approach for saving lives wherever higher grounds cannot be reached rapidly. With this in mind, a novel interdisciplinary hydraulic-structural engineering research program [Palermo *et al.*, 2008; Nistor *et al.*, 2009b, Nistor *et al.*, 2011] was initiated at the University of Ottawa in collaboration with the Canadian Hydraulic Center of the National Research Council (NRC-CHC). The goal of this comprehensive research program is to develop new recommendations and guidelines for the sound design of inland structures located in tsunami-prone areas. Although this ongoing research program comprises field surveys as well as the physical, numerical, and analytical modeling of both tsunami-induced debris impact and hydrodynamic forces on structures, this thesis focuses mainly on its numerical modeling component.

With the ever increasing computational capabilities of modern computers, numerical modeling has become a very valuable tool in the engineering field, from scientific investigation to the actual design process. However, before a new computer model can be applied with confidence in either research or design, it should be validated through physical experiments and/or field observations. Over the last few decades, numerical models relying on grid-based methods have evolved significantly and are now widely used. In the late seventies, a new breed of numerical models that apply mesh-free methods were introduced. However, due to their relatively high computing requirements, it was not until recent advances in computing technology that this type of numerical model began to be widely applied. The Smoothed Particle Hydrodynamics (SPH) method, considered to be one of the most promising mesh-free methods, is used in the present study to simulate the hydrodynamic forces acting on structures due to tsunami-induced hydraulic bores.

1.2. Scope

The primary goal of the present study was to numerically reproduce part of the ongoing physical experiments performed within the aforementioned research program. These large-scale experiments are concerned with the impact ‘mechanics’ of rapidly advancing

tsunami-like hydraulic bores on free-standing columns having both circular and square cross-sections. The hydraulic bores are generated in a rectangular-shaped channel by the sudden release of water from an upstream reservoir. This physical arrangement is based on the classic “dam-break” problem. The compared numerical and physical results include: (i) time-dependent pressure distributions exerted on the upstream face of the structures, (ii) time-histories of the net base horizontal force acting on the structures, (iii) flow velocities, and (iv) water surface elevations at various locations along the experimental channel.

The validated numerical model is applied to investigate the influence of the experimental test geometry on the resulting hydrodynamic forces. Due to both financial and space limitations, such tasks would be extremely difficult to perform physically.

The presence of a thin film of water, initially present over the channel’s floor in the physical experiments, was observed to have a significant influence on the magnitudes of the resulting hydrodynamic forces. To investigate further the impact of a wet-bed condition on system ‘dynamics’, additional numerical tests were performed that compared the characteristics of hydraulic bore impacts on free-standing square columns and vertical wall for both dry- and wet-bed conditions.

1.3. Contributions

The novelty of the present study resides in the numerical investigation, at a relatively large-scale and using the SPH method, of the interaction of a hydraulic bore with free-standing structures. Although this particular type of bore-structure interaction has been previously simulated using the SPH method, past studies only considered cases of limited dimensions. Hence, little information was provided that considered the time-history of hydrodynamic forces under the extended duration of a hydraulic bore similar to that of an actual tsunami.

Although the influence of the bed condition (i.e. dry or wet) on the propagation alone of hydraulic bores has been thoroughly investigated by others, little is known about the role of the bed condition in regards to the hydrodynamic forces resulting from the impact of these bores with structures. To date, studies considering not only both bed

conditions but also two fundamentally different structure types, such as slender columns and infinitely wide walls, have not been performed either experimentally or numerically.

In addition, as one of the very few Canadian studies considering the SPH method, one objective of this research is to provide a basic understanding of method and of its application for the simulation of hydrodynamic forces on structures.

Finally, the findings of this research can also be applied to other scenarios, such as (a) the rare but possible event of a dam or levee failure, where communities would be located in the path of the inundation wave, and (b) “green water” loading, which occurs as a large sea wave overtops onto the deck of a ship.

1.4. Outline

Chapter 1 provides an introduction to the study by explaining its objective, scope, and contributions to the related field of research.

In Chapter 2, the SPH method is first introduced in perspective within the general concepts of numerical simulations and a review of its general applications in the field of Computational Fluid Dynamics (CFD) is included. In an adapted notation utilized throughout this thesis, the most common formulation of the method is derived in detail. In addition, important numerical aspects are described in a concise manner, not only according to the *SPHysics* implementation [Gómez-Gesteira *et al.*, 2010a] that was used to perform the numerical simulations hereby, but also according to various studies. Observations related to these numerical aspects by other researchers are mentioned for later reference.

A detailed literature review with respect to bore-structure interaction is presented in Chapter 3. First, approaches for reproducing tsunami waves experimentally are explained and the results of studies demonstrating the influence of the bed condition on the propagation alone of hydraulic bores are then presented. Afterwards, existing results of current research on the impact of hydraulic bores with structures are reviewed. Lastly, previous numerical studies on such impacts performed using the SPH method are described. The main objectives of the literature review is to provide a background on concepts related to the dam-break scenario used to reproduce tsunami-like hydraulic bores, provide a starting point for the interpretation of the results presented in this thesis,

and demonstrate the novelty of the latter with respect to similar studies performed with the SPH method.

In Chapter 4, the several experimental setups considered in this study are detailed, along with the computational domains that were used to reproduce them numerically. This chapter also details the calibration of the numerical model based on a sensitivity analysis.

Chapter 5 discusses approaches for extracting specific results from the numerical simulations and these are evaluated based on a preliminary comparison with experimental data. The preferred approach considers the post-processing of raw numerical outputs by specifically implemented programs, and was used for obtaining the numerical results presented in the subsequent chapters of this thesis.

The core of this study is found in Chapter 6, where numerical results are compared quantitatively to experimental ones.

With the numerical model validated according to the observed agreement in the previous chapter, the results of additional numerical tests are presented in Chapter 7. The purpose of these tests is to fully exploit the numerical model in order to gain a better understanding of the underlying physical phenomena. The influences of the experimental test geometry and bed condition are investigated in detail.

Finally, research conclusions and recommendations for future work are presented in Chapter 8.

Chapter 2. The SPH Method

This chapter presents the general underlying theory of the SPH method. First, the method is introduced conceptually within a description of the fundamental concepts of numerical simulations along with the principal characteristics of grid-based and mesh-free methods (according to Liu and Liu [2003]). Secondly, the most common formulation of the method applied to the governing equations of fluid motion is derived in step-by-step detail following a general approach provided by Dalrymple [2007]. Finally, numerical aspects related to the method are described according to several sources. The notation used in this chapter was specially adapted by considering the advantages of several existing notations. The purpose of this is to not only enhance readability but also provide a constant notation for the remainder of the thesis as the content of this chapter will often be referred to.

2.1. SPH in Perspective

2.1.1 Numerical Simulations

Numerical simulations using computer models play a significant role in engineering as they allow relatively quick iterative design optimization and the behavioural evaluation of complex systems. Moreover, numerical simulations can either be an alternative or a complementary tool to scientific investigation, reducing the occurrence of expensive and time-consuming physical experiments in laboratories. Allowing measurements that are physically unfeasible, numerical simulations can provide additional insights into experimental results and even lead to the discovery of new phenomena. Nevertheless, it is only when their application to a specific type of problem has been validated against either sets of experimental results or theoretical solutions that numerical simulations should be considered as a practical design tool or as a reliable alternative to physical experiments.

Numerical simulations are obtained based on the following general procedure. Through laboratory experiments, physical phenomena are first observed, and mathematical models are then established with some possible simplifications and assumptions. These mathematical models are generally expressed as a set of governing equations and, in most circumstances, they are in the form of *partial differential equations* (PDE) with associated boundary and initial conditions. Since these equations often do not have analytical solutions, they require to be solved numerically. In order to do so, the geometry of the problem domain needs to be divided into discrete components over which the solution of the equations will be approximated. This is referred to as *domain discretization* and it consists in dividing a continuum problem domain into a finite number of components that form the *computational frame* for the approximation of the governing equations. This computational frame is traditionally a grid or mesh, which forms a lattice of points (or grid nodes), with defined connectivity that represents the geometry of the problem domain. It is at these points that field variables are to be evaluated. To solve the governing equations based on the computational frame, the integral or derivative operations within them must be converted from continuous form to discrete representation. This is referred to as *numerical discretization* which is achieved via a *function approximation theory*. After domain discretization and numerical discretization, the original governing equations are changed into a set of algebraic equations or *ordinary differential equations* (ODE) that can be solved using existing time-integration schemes.

2.1.2 Grid-Based Methods

There are two ways of describing the physical governing equations: the *Lagrangian* description and the *Eulerian* description. Within traditional grid-based methods, the Lagrangian description is typically represented by the *finite element method* (FEM) while the Eulerian description is typically represented by the *finite Difference method* (FDM) or the *finite volume method* (FVM). In grid-based methods following the Lagrangian description, the grid is attached onto the material over the entire computation process, and therefore it moves with the material. Since each grid node follows the path of the material at a grid point, relative movement of the connecting nodes may result in the

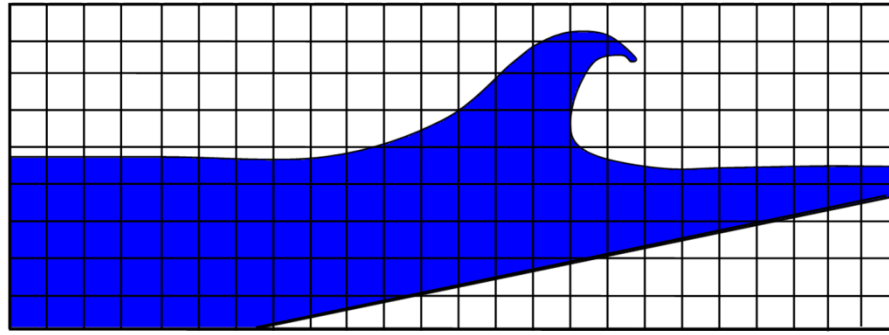
deformation of the grid cells. Mass, momentum and energy are transported with the movement of the grid cells and the mass within each cell remains fixed, as no mass flux crosses the cell boundaries. The numerical discretization is performed via the function approximation theory of *series representation*. Since the deformation of the grid can be problematic in simulations where the material deformation is large, Lagrangian grid-based methods are generally more popular for solving *computational solid mechanics* (CSM) problems. In the case of the grid-based methods following the Eulerian description, the grid is spatially fixed and the simulated medium moves across the cells of this grid (Figure 2.1-a). The flux of mass, momentum and energy across the grid cell boundaries are simulated in order to determine the distribution of mass, velocity, energy, etc. within the problem domain. The numerical discretization is performed via the function approximation theory of *differential representation*. Since the grid is fixed in space, large deformations in the simulated material do not cause any deformation in the grid itself. Therefore problematic issues related to highly-distorted grids (as in the case of Lagrangian grid-based methods) are avoided. For this reason, Eulerian grid-based methods are dominant in the area of *computational fluid dynamics* (CFD).

2.1.3 Mesh-free Methods

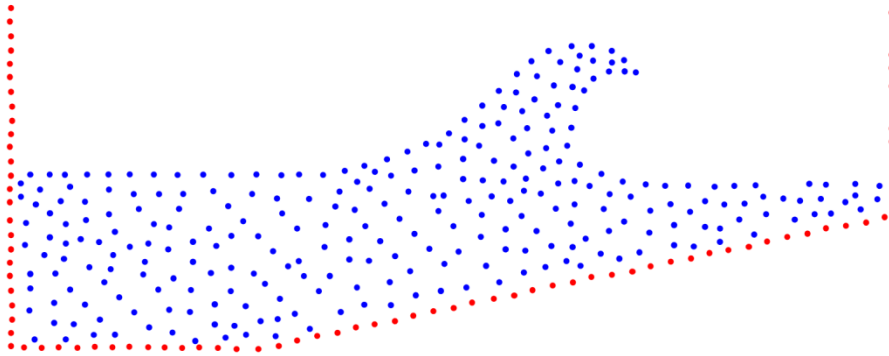
Recently, strong efforts have been geared towards a new generation of computational methods called *mesh-free methods*. The key role of mesh-free methods is to provide a numerical solution for PDEs with a set of arbitrarily distributed nodes, without using any grid that defines the connectivity between these nodes. Such methods can be advantageous compared to traditional grid-based ones when considering problems with a free-surface, deformable boundaries, moving interfaces (with respect to FDM), large deformations (with respect to FEM), and complex grid generation.

Mesh-free Particle Methods

Generally, *mesh-free particle methods* (MPM) refers to the class of mesh-free methods that employ a set of finite number of discrete nodes, or particles to represent the state of a system and to record the movement of this system. Each particle can either be directly associated with one discrete physical object as in the case of stars in astrophysical



(a)



(b)

Figure 2.1. Conceptualized computational nodes of (a) grid-based methods and (b) mesh-free methods.

problems, or be generated to represent a part of a continuum problem domain as in the case of a fluid particle within a water body (Figure 2.1-b). In MPMs, the domain discretization is often referred to as *particle representation*. Numerical discretization within the MPMs, often referred to as *particle approximation*, involves approximating the values of functions, derivatives and integrals at a certain particle using the information of the neighbouring particles which have an influence on this particle.

The SPH Method

The *Smoothed Particle Hydrodynamics* (SPH) method is a mesh-free, Lagrangian particle method in which the state of a system is represented by a set of arbitrarily distributed particles, which possess individual material properties and move according to governing conservation equations. The method was first introduced in the late seventies by Gingold and Monaghan [1977], and by Lucy [1977] independently, in order to model

astrophysical problems where each stellar object would be represented by a single particle. Since the collective movement of those stellar objects is similar to the movement of a liquid or gas flow, they could be modeled by the governing equations of the classical Newtonian hydrodynamics. Since its introduction, the method was applied to a wide variety of fields and the reader is referred to Liu and Liu [2003] for an exhaustive enumeration of these applications. Within the field of CFD, it was in 1994 that the method was first considered to simulate free-surface flows [Monaghan, 1994]. Afterwards, the method has been applied to a wide variety of hydraulic cases such as dam-breaks [Crespo *et al.*, 2008; Khayyer and Gotoh, 2010], hydraulic jumps [López *et al.*, 2010; Jonsson *et al.*, 2011], floating objects [Rogers *et al.*, 2008; Vandamme *et al.*, 2011], and flow through porous media [e.g. Morris *et al.*, 1999; Zhu *et al.*, 1999; Jiang *et al.*, 2007], etc.

In the field of Coastal Engineering particularly, the SPH method has been used for the simulation of wave propagation and breaking [Monaghan and Kos, 1999; Lo and Shao, 2002; Landrini *et al.*, 2007; Guilcher *et al.*, 2007; Dalrymple *et al.*, 2009], the interaction of waves with breakwaters [Gotoh *et al.*, 2004; Shao, 2005; Didier and Neves, 2010; Rogers *et al.*, 2010], wave transmission through porous rubble-mounds [Shao, 2010] and sediment transport [Zou and Dalrymple, 2006; Zou, 2007; Ulrich and Rung, 2010; Manenti *et al.*, 2012]. Since the role of air entrainment is known to be important in coastal processes, studies related to multi-phase flows have also been performed [Colagrossi and Landrini, 2005; Cuomo *et al.*, 2006; Hu and Adams, 2009, Gong *et al.*, 2010].

As explained in the following sections of this chapter, the discretization of the governing equations relies on approximating functions at one particle using a weighted average, or *smoothing*, of the properties of its neighbouring particles. Since the formulation of the SPH method is not affected by the arbitrariness of the spatial distribution of the particles, the method can handle problems with large deformations without any special free-surface treatment. This is considered to be its most attractive feature. However, as the neighbours of each particle have to be searched at every time step, the method is computationally expensive especially for 3D applications, although recent advances in computer technology have helped overcome this difficulty. For

instance, Maruzewski *et al.* [2010] studied the water entry of solid objects such as ship hulls using a parallel SPH model executing on as much as 8,192 processors, while Dalrymple *et al.* [2010] have taken advantage of the more affordable computational power of Graphic Processing Units (GPUs) to simulate the physical experiments of Drønen *et al.* [2002]. These experiments reproduced near-shore circulation patterns resulting from waves propagating and breaking over a bar and a rip channel. Nevertheless, depending on the problem to solve, the SPH method generally requires a high resolution (i.e. small spacing between adjacent particles) in order to achieve accurate results. Hence, modeling large computational domains using a high resolution can be challenging in terms of computational time, even with parallel computing. With this in mind, although significant work is still required, a few recent studies [Vacondio *et al.*, 2011; Omidvar *et al.*, 2012] have focused on implementing SPH models with a variable particle resolution in analogy to a variable grid size within conventional mesh-based methods.

2.2. Essential Formulation of SPH

In the SPH method, the numerical discretization results from the combination of two steps: the function approximation and the particle approximation.

2.2.1 Function Approximation

The function approximation theory considered in the SPH method is the *integral representation of functions*, which approximates a function $\mathbf{f}(\mathbf{x})$, in this case a vector function, using its information within a local *influence domain* Ω , such that:

$$[2.1] \quad \mathbf{f}(\mathbf{x}) = \int_{\Omega} \mathbf{f}(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\Omega$$

where $\mathbf{x}$ is the position vector of the location where $\mathbf{f}$ is being approximated, $\mathbf{x}'$ represents the position vector of surrounding locations where $\mathbf{f}$ is being evaluated in order

to be approximated at $\mathbf{x}$, $d\Omega$ is the infinitesimal volume at a location $\mathbf{x}'$, and δ is the Dirac delta function given by

$$[2.2] \quad \delta(\mathbf{x} - \mathbf{x}') = \begin{cases} 1, & \mathbf{x} = \mathbf{x}' \\ 0, & \mathbf{x} \neq \mathbf{x}' \end{cases}$$

When using the Dirac delta function, the integral representation of $\mathbf{f}(\mathbf{x})$ is exact as long as this last one is defined and is continuous in Ω . Replacing the Dirac function by a scalar smoothing function W , often referred to as the *smoothing kernel*, the integral representation of $\mathbf{f}(\mathbf{x})$ is given by

$$[2.3] \quad \langle \mathbf{f}(\mathbf{x}) \rangle = \int_{\Omega} \mathbf{f}(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\Omega$$

in which h is the *smoothing length* defining the influence domain of the smoothing kernel W . The function approximation given by Eq. 2.3 is often referred to as *kernel approximation*. In order for Eq. 2.3 to be valid, the smoothing kernel must satisfy the following conditions:

- Normalization condition:

$$[2.4] \quad \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\Omega = 1$$

- Delta function behaviour:

$$[2.5] \quad \lim_{h \rightarrow 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}')$$

- Compact support condition:

$$[2.6] \quad W(\mathbf{x} - \mathbf{x}', h) = 0 \quad \text{for} \quad \|\mathbf{x} - \mathbf{x}'\| \geq 2h$$

in which $2h$ is the radius of the effective (non-zero) influence domain associated with the smoothing kernel.

- Positivity condition:

$$[2.7] \quad W(\mathbf{x} - \mathbf{x}', h) > 0 \quad \text{for} \quad \|\mathbf{x} - \mathbf{x}'\| < 2h$$

- Monotonically decreasing:

$$[2.8] \quad W(\mathbf{x} - \mathbf{x}_1', h) > W(\mathbf{x} - \mathbf{x}_2', h) \quad \text{for} \quad \|\mathbf{x} - \mathbf{x}_1'\| < \|\mathbf{x} - \mathbf{x}_2'\|$$

- Symmetry condition:

$$[2.9] \quad W(\mathbf{x} - \mathbf{x}', h) = W(\mathbf{x}' - \mathbf{x}, h)$$

If we now consider the spatial derivative of the function $\mathbf{f}(\mathbf{x})$, it can be obtained by simply replacing the term $\mathbf{f}(\mathbf{x})$ of Eq. 2.3 with $\nabla \cdot \mathbf{f}(\mathbf{x})$,

$$[2.10] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}) \rangle = \int_{\Omega} [\nabla \cdot \mathbf{f}(\mathbf{x}')] W(\mathbf{x} - \mathbf{x}', h) d\Omega$$

For simplicity, let's consider

$$[2.11] \quad W = W(\mathbf{x} - \mathbf{x}', h)$$

In three-dimensional space,

$$\begin{aligned}
[2.12] \quad [\nabla \cdot \mathbf{f}(\mathbf{x}')]W &= \left[\frac{\partial f_x(\mathbf{x}')}{\partial x} + \frac{\partial f_y(\mathbf{x}')}{\partial y} + \frac{\partial f_z(\mathbf{x}')}{\partial z} \right] W \\
&= \left[\frac{W \partial f_x(\mathbf{x}')}{\partial x} + \frac{W \partial f_y(\mathbf{x}')}{\partial y} + \frac{W \partial f_z(\mathbf{x}')}{\partial z} \right] \\
&= \left[\frac{W \partial f_x(\mathbf{x}')}{\partial x} + \frac{W \partial f_y(\mathbf{x}')}{\partial y} + \frac{W \partial f_z(\mathbf{x}')}{\partial z} \right] \\
&\quad + \left[\frac{f_x(\mathbf{x}') \partial W}{\partial x} + \frac{f_y(\mathbf{x}') \partial W}{\partial y} + \frac{f_z(\mathbf{x}') \partial W}{\partial z} \right] \\
&\quad - \left[\frac{f_x(\mathbf{x}') \partial W}{\partial x} + \frac{f_y(\mathbf{x}') \partial W}{\partial y} + \frac{f_z(\mathbf{x}') \partial W}{\partial z} \right] \\
&= \left[\frac{\partial f_x(\mathbf{x}') W}{\partial x} + \frac{\partial f_y(\mathbf{x}') W}{\partial y} + \frac{\partial f_z(\mathbf{x}') W}{\partial z} \right] \\
&\quad - \left[\frac{f_x(\mathbf{x}') \partial W}{\partial x} + \frac{f_y(\mathbf{x}') \partial W}{\partial y} + \frac{f_z(\mathbf{x}') \partial W}{\partial z} \right] \\
&= \nabla \cdot [\mathbf{f}(\mathbf{x}') W] - \mathbf{f}(\mathbf{x}') \cdot \nabla W
\end{aligned}$$

Substituting the identity derived in Eq. 2.12 into Eq. 2.10 yields

$$[2.13] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}) \rangle = \int_{\Omega} \{ \nabla \cdot [\mathbf{f}(\mathbf{x}') W] - \mathbf{f}(\mathbf{x}') \cdot \nabla W \} d\Omega$$

which can be rewritten as

$$[2.14] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}) \rangle = \int_{\Omega} \nabla \cdot [\mathbf{f}(\mathbf{x}') W] d\Omega - \int_{\Omega} \mathbf{f}(\mathbf{x}') \cdot \nabla W d\Omega$$

Using the divergence theorem (Eq. 2.15), the first integral on the right hand side (RHS) of Eq. 2.14 can be converted into an integral over the surface S of the influence domain Ω .

$$[2.15] \quad \int_{\Omega} \nabla \cdot \mathbf{f} d\Omega = \int_S \mathbf{f} \cdot \mathbf{n} dS$$

where $\mathbf{n}$ is the unit vector normal to the surface S pointing outwards. Eq. 2.14 then becomes

$$[2.16] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}) \rangle = \int_S \mathbf{f}(\mathbf{x}') W \cdot \mathbf{n} dS - \int_{\Omega} \mathbf{f}(\mathbf{x}') \cdot \nabla W d\Omega$$

Due to the compact support condition (Eq. 2.6), the value of the smoothing kernel W is zero on the surface of its influence domain and therefore, the first integral on the RHS of Eq. 2.16 is zero. This results in the final form of the kernel approximation of the spatial derivative of a function $\mathbf{f}(\mathbf{x})$,

$$[2.17] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}) \rangle = - \int_{\Omega} \mathbf{f}(\mathbf{x}') \cdot \nabla W d\Omega$$

Note that, although a vector function ($\mathbf{f}$) is considered for demonstrative purposes, the integral representation of a scalar function (f) follows a similar approach, where the divergence operator is replaced by the gradient operator. In this case, the identity derived in Eq. 2.12 becomes

$$\begin{aligned}
[2.18] \quad [\nabla f(\mathbf{x}')]W &= \left[\frac{\partial f(\mathbf{x}')}{\partial x}, \frac{\partial f(\mathbf{x}')}{\partial y}, \frac{\partial f(\mathbf{x}')}{\partial z} \right] W \\
&= \left[\frac{W \partial f(\mathbf{x}')}{\partial x}, \frac{W \partial f(\mathbf{x}')}{\partial y}, \frac{W \partial f(\mathbf{x}')}{\partial z} \right] \\
&= \left[\frac{W \partial f(\mathbf{x}')}{\partial x}, \frac{W \partial f(\mathbf{x}')}{\partial y}, \frac{W \partial f(\mathbf{x}')}{\partial z} \right] \\
&\quad + \left[\frac{f(\mathbf{x}') \partial W}{\partial x}, \frac{f(\mathbf{x}') \partial W}{\partial y}, \frac{f(\mathbf{x}') \partial W}{\partial z} \right] \\
&\quad - \left[\frac{f(\mathbf{x}') \partial W}{\partial x}, \frac{f(\mathbf{x}') \partial W}{\partial y}, \frac{f(\mathbf{x}') \partial W}{\partial z} \right] \\
&= \left[\frac{W \partial f(\mathbf{x}')}{\partial x} + \frac{f(\mathbf{x}') \partial W}{\partial x}, \frac{W \partial f(\mathbf{x}')}{\partial y} + \frac{f(\mathbf{x}') \partial W}{\partial y}, \frac{W \partial f(\mathbf{x}')}{\partial z} + \frac{f(\mathbf{x}') \partial W}{\partial z} \right] \\
&\quad - \left[\frac{f(\mathbf{x}') \partial W}{\partial x}, \frac{f(\mathbf{x}') \partial W}{\partial y}, \frac{f(\mathbf{x}') \partial W}{\partial z} \right] \\
&= \left[\frac{\partial f(\mathbf{x}') W}{\partial x}, \frac{\partial f(\mathbf{x}') W}{\partial y}, \frac{\partial f(\mathbf{x}') W}{\partial z} \right] \\
&\quad - \left[\frac{f(\mathbf{x}') \partial W}{\partial x}, \frac{f(\mathbf{x}') \partial W}{\partial y}, \frac{f(\mathbf{x}') \partial W}{\partial z} \right] \\
&= \nabla[f(\mathbf{x}')W] - f(\mathbf{x}')\nabla W
\end{aligned}$$

and the following corollary of the divergence theorem is used instead.

$$[2.19] \quad \int_{\Omega} \nabla f d\Omega = \int_S f \mathbf{n} dS$$

2.2.2 Particle Approximation

The continuous integral representations expressed in Eq. 2.3 and Eq. 2.17 can be converted into discrete form through summations over all particles within the influence domain of the smoothing kernel. Figure 2.2 represents the influence domain of the smoothing kernel for a certain particle i located at $\mathbf{x}_i$.

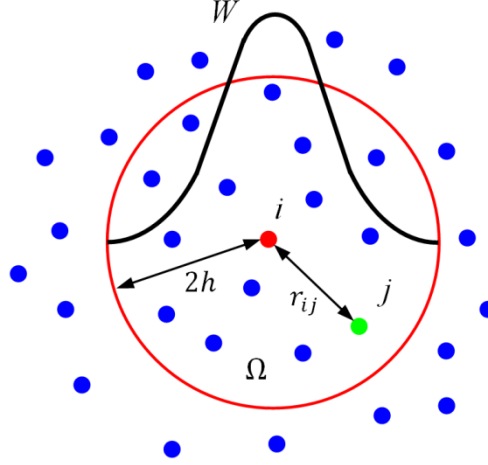


Figure 2.2. Influence domain Ω with radius $2h$ of smoothing kernel W . r_{ij} is the distance separating particles i and j .

The particle approximation is obtained by replacing the infinitesimal volume $d\Omega$ at the location of a neighbour particle j by the finite volume of this particle ΔV_j . Considering Eq. 2.3:

$$[2.20] \quad \langle f(x) \rangle = \int_{\Omega} f(x') W(x - x', h) d\Omega = \sum_{j=1}^N f(x_j) W(x_i - x_j, h) \Delta V_j$$

where ΔV_j is the incremental volume associated with particle j , and N is the total number of neighbouring particles within the influence domain Ω . Using the following relationship:

$$[2.21] \quad \Delta V_j = \frac{m_j}{\rho_j}$$

where m_j and ρ_j are the mass and density of a neighbouring particle j , respectively, the particle approximation of the function f at particle i is then

$$[2.22] \quad \langle \mathbf{f}(\mathbf{x}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{f}(\mathbf{x}_j) W(\mathbf{x}_i - \mathbf{x}_j, h)$$

For simplicity, let's consider

$$[2.23] \quad W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$$

Following the same approach, the spatial derivative of the function $\mathbf{f}$ at particle i can be expressed as

$$[2.24] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}_i) \rangle = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{f}(\mathbf{x}_j) \cdot \nabla_j W_{ij}$$

where ∇_j denotes the gradient taken with respect to the coordinates of particle j . Since the smoothing kernel is symmetric (Eq. 2.9),

$$[2.25] \quad \nabla_j W_{ij} = -\nabla_i W_{ij}$$

and Eq. 2.24 becomes

$$[2.26] \quad \langle \nabla \cdot \mathbf{f}(\mathbf{x}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{f}(\mathbf{x}_j) \cdot \nabla_i W_{ij}$$

with

$$[2.27] \quad \nabla_i W_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}}$$

where $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the distance separating particle j from particle i .

2.2.3 Approximation Errors

The errors related to the SPH method are categorized as either function approximation errors, or errors associated to the particle approximation. The overall error is the combination of the two. In a different way than traditional grid-based methods, there are two length scales affecting the errors in the SPH method. These intricately related scales are the smoothing length (h) and the inter-particle distance, or spacing, with the latter being irregular in most practical cases. [Note that, this section does not discuss the errors associated to time-integration schemes as these are independent of the numerical method].

Function Approximation Errors

These errors are related to how functions are approximated. In the SPH method, the function approximation theory utilized to perform the latter is the integral representation of functions (see Section 2.2.1).

As $h \rightarrow 0$, the smoothing kernel W further behaves like the Dirac delta function (Eq. 2.2) and the function approximation, or kernel approximation provided by Eq. 2.3 becomes exact. However, as this would require an infinite number of particles in the influence domain ($N \rightarrow \infty$), this is not practically feasible and errors associated with the calculation of W are introduced. The accuracy of the kernel approximation is generally considered as $O(h^2)$ [Monaghan, 1992]. Following Liu and Liu [2003], this can be demonstrated with Eq. 2.3 using a Taylor expansion of $f(\mathbf{x})$ around $\mathbf{x}$ as follows:

$$\begin{aligned}
[2.28] \quad \langle f(\mathbf{x}) \rangle &= \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\Omega \\
&= \int_{\Omega} \{f(\mathbf{x}) + f'(\mathbf{x})(\mathbf{x}' - \mathbf{x}) + R[(\mathbf{x}' - \mathbf{x})^2]\} W(\mathbf{x} - \mathbf{x}', h) d\Omega \\
&= \int_{\Omega} f(\mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\Omega + \int_{\Omega} f'(\mathbf{x})(\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\Omega \\
&\quad + \int_{\Omega} R(h^2) W(\mathbf{x} - \mathbf{x}', h) d\Omega \\
&= f(\mathbf{x}) \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\Omega + f'(\mathbf{x}) \int_{\Omega} (\mathbf{x}' - \mathbf{x}) W(\mathbf{x} - \mathbf{x}', h) d\Omega \\
&\quad + R(h^2) \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\Omega
\end{aligned}$$

where R stands for the residual. Since W is symmetric (i.e. even function) as stated by Eq. 2.9, $(\mathbf{x}' - \mathbf{x})W(\mathbf{x} - \mathbf{x}', h)d\Omega$ is an odd function and therefore its integral over the influence domain Ω is equal to zero. Considering this zero integral and the normalization condition (Eq. 2.4):

$$[2.29] \quad \langle f(\mathbf{x}) \rangle = f(\mathbf{x}) + R(h^2)$$

Using the same concept of *consistency* as in traditional grid-based methods, Liu and Liu also demonstrate that in order to reproduce a polynomial of degree n exactly, W must satisfy the following conditions:

$$\begin{aligned}
M_0 &= \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\Omega = 1 \\
M_1 &= \int_{\Omega} (\mathbf{x} - \mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\Omega = 0 \\
[2.30] \quad M_2 &= \int_{\Omega} (\mathbf{x} - \mathbf{x}')^2 W(\mathbf{x} - \mathbf{x}', h) d\Omega = 0 \\
&\quad \vdots \\
M_n &= \int_{\Omega} (\mathbf{x} - \mathbf{x}')^n W(\mathbf{x} - \mathbf{x}', h) d\Omega = 0
\end{aligned}$$

with

$$[2.31] \quad \left. \begin{aligned} W(\mathbf{x} - \mathbf{x}', h) &= 0 \\ W'(\mathbf{x} - \mathbf{x}', h) &= 0 \end{aligned} \right\} \quad \text{for} \quad \|\mathbf{x} - \mathbf{x}'\| \geq 2h$$

where M_k are the k -th moments of the smoothing kernel W . Eq. 2.31 states that both W and its first derivative should be equal to zero on the surface of Ω . Although high orders of consistency are theoretically possible with the integral representation of functions, they are not always advantageous. For instance, for the even moments ($k = 2, 4, 6 \dots$) to be zero, W has to be negative in some parts of Ω . This would make the positivity condition (Eq. 2.7) unsatisfied and negative values of W could result in unphysical solutions such as negative density values [Liu and Liu, 2003]. Hence, when applied to CFD, the consistency of the SPH method is generally first-order (i.e. C^1).

Particle Approximation Errors

These errors are associated with the discretization of the kernel approximation (Eq. 2.3) as the computational domain is represented with a discrete set of particles. In contrast to grid-based methods, the distance separating particles in SPH simulations is generally not constant as these ones move according to the governing equations. Hence, because of this rather random distribution of the computational nodes, traditional error estimates are not possible. For this reason, errors associated to the domain discretization are generally assessed based on numerical experiments and comparison with known analytical

solutions [Monaghan, 2005]. The following are some examples of such study. Interested in the stability of the SPH method, Swegle *et al.* [1995] performed von Neumann analyses within the perspective of computational solid mechanics (CSM). By simulating heat transfer in a homogenous slab, Cleary and Monaghan [1998] found that reducing h more slowly than the inter-particle spacing reduces the overall error. Considering the randomness of the particles and the ratio between h and the particle spacing, Graham and Hughes [2007] performed numerical experiments of an impulsive pressure-driven flow between parallel plates to assess the accuracy of SPH simulations of viscous diffusion.

2.2.4 Derivation of SPH Formulation for Fluid Motion

Based on the numerical discretization procedure described in the previous section, SPH formulations for PDEs can be derived in various ways. One approach introduced by Benz [1990], is to first multiply each term of the PDEs by the smoothing kernel and then integrate over the influence domain of the last one. In this study, we will consider the compressible Navier-Stokes equations in their Lagrangian form as the set of governing equations for the motion of the fluid. These PDEs will be transformed step-by-step into a discretized set of ODEs that can be solved via time integration, following the general approach provided by Dalrymple [2007].

Conservation of Mass

In the case of compressible flows, the conservation of mass in Lagrangian form is as follows:

$$[2.32] \quad \frac{1}{\rho} \frac{D\rho}{Dt} = -\nabla \cdot \mathbf{u}$$

with

$$[2.33] \quad \frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = \frac{\partial \rho}{\partial t} + u_x \frac{\partial \rho}{\partial x} + u_y \frac{\partial \rho}{\partial y} + u_z \frac{\partial \rho}{\partial z}$$

where ρ and $\mathbf{u}$ are the density and the velocity vector of the fluid, respectively. Multiplying both sides of Eq. 2.32 with the smoothing kernel $W = W(\mathbf{x} - \mathbf{x}', h)$ and integrating over the influence domain Ω gives:

$$[2.34] \quad \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) \left(\frac{1}{\rho} \frac{D\rho}{Dt} \right) d\Omega = - \int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) \nabla \cdot \mathbf{u} d\Omega$$

Using the same identity as demonstrated in Eq. 2.12, taking $\mathbf{f}$ as $\mathbf{u}$ and considering $W = W(\mathbf{x} - \mathbf{x}', h)$ for clarity,

$$[2.35] \quad \int_{\Omega} W \left(\frac{1}{\rho} \frac{D\rho}{Dt} \right) d\Omega = - \int_{\Omega} \{ \nabla \cdot [W\mathbf{u}] - \mathbf{u} \cdot \nabla W \} d\Omega$$

Using the divergence theorem (Eq. 2.15),

$$[2.36] \quad \int_{\Omega} W \left(\frac{1}{\rho} \frac{D\rho}{Dt} \right) d\Omega = - \int_S W\mathbf{u} \cdot \mathbf{n} dS + \int_{\Omega} \mathbf{u} \cdot \nabla W d\Omega$$

Due to the compact support condition (Eq. 2.6), the value of the smoothing kernel W is zero on the surface of its influence domain and therefore the first integral on the RHS of Eq. 2.36 is zero.

$$[2.37] \quad \int_{\Omega} W \left(\frac{1}{\rho} \frac{D\rho}{Dt} \right) d\Omega = \int_{\Omega} \mathbf{u} \cdot \nabla W d\Omega$$

Applying the concept of particle approximation to Eq. 2.37 gives:

$$[2.38] \quad \sum_{j=1}^N \frac{m_j}{\rho_j} \left(\frac{1}{\rho_i} \frac{D\rho_i}{Dt} \right) W(\mathbf{x}_i - \mathbf{x}_j, h) = \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_j W(\mathbf{x}_i - \mathbf{x}_j, h)$$

Because of the symmetry of the smoothing kernel (Eq. 2.9 and Eq. 2.25) and taking $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$ for simplicity, Eq. 2.38 can be rewritten as

$$[2.39] \quad \sum_{j=1}^N \frac{m_j}{\rho_j} \left(\frac{1}{\rho_i} \frac{D\rho_i}{Dt} \right) W_{ij} = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_i W_{ij}$$

Since $D\rho_i/Dt$ and the density (ρ_i) at particle i are constant with respect to the summation over the neighbouring particles $j = 1, 2, \dots, N$, they can be excluded from the summation of the left hand side (LHS) of Eq. 2.39,

$$[2.40] \quad \left(\frac{1}{\rho_i} \frac{D\rho_i}{Dt} \right) \sum_{j=1}^N \frac{m_j}{\rho_j} W_{ij} = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_i W_{ij}$$

Due to the normalization condition (Eq. 2.4),

$$[2.41] \quad \frac{1}{\rho_i} \frac{D\rho_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_i W_{ij}$$

Since the gradient of a constant is equal to zero, we can deduce with Eq. 2.26 that

$$[2.42] \quad \sum_{j=1}^N \frac{m_j}{\rho_j} \nabla_i W_{ij} = 0$$

Multiplying this zero sum by $\mathbf{u}_i$ and adding it to the RHS of Eq. 2.41 gives:

$$[2.43] \quad \frac{1}{\rho_i} \frac{D\rho_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_i W_{ij} + \mathbf{u}_i \cdot \sum_{j=1}^N \frac{m_j}{\rho_j} \nabla_i W_{ij}$$

As before, since the velocity at particle i ($\mathbf{u}_i$) is constant with respect to the summation over the neighbouring particles $j = 1, 2, \dots, N$, it can be included in the sum of the second term on the RHS of Eq. 2.43.

$$[2.44] \quad \frac{1}{\rho_i} \frac{D\rho_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_j \cdot \nabla_i W_{ij} + \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{u}_i \cdot \nabla_i W_{ij}$$

Rearranging Eq. 2.44 yields the final form of the conservation of mass equation in SPH notation

$$[2.45] \quad \frac{D\rho_i}{Dt} = \rho_i \sum_{j=1}^N \frac{m_j}{\rho_j} (\mathbf{u}_i - \mathbf{u}_j) \cdot \nabla_i W_{ij}$$

Conservation of Momentum

The conservation of momentum in Lagrangian form (without viscous diffusion terms) is expressed as follows:

$$[2.46] \quad \frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla P + \mathbf{F}$$

with

$$[2.47] \quad \frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \rho = \frac{\partial \mathbf{u}}{\partial t} + u_x \frac{\partial \mathbf{u}}{\partial x} + u_y \frac{\partial \mathbf{u}}{\partial y} + u_z \frac{\partial \mathbf{u}}{\partial z}$$

where $\mathbf{F}$ is the acceleration due to external forces and P is the fluid pressure. [Note that, for simplicity, the viscous diffusion terms are omitted, as they depend on the various viscosity and turbulence model selected]. Multiplying both sides of Eq. 2.46 with the smoothing kernel $W = W(\mathbf{x} - \mathbf{x}', h)$ and integrating over the influence domain Ω gives:

$$[2.48] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = \int_{\Omega} W \left(-\frac{1}{\rho} \nabla P + \mathbf{F} \right) d\Omega$$

$$[2.49] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = - \int_{\Omega} W \left(\frac{1}{\rho} \nabla P \right) d\Omega + \int_{\Omega} W \mathbf{F} d\Omega$$

Considering that the integration of the acceleration due to external forces ($\mathbf{F}$) will be accounted for in its final formulation, Eq. 2.49 is simplified as follows:

$$[2.50] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = - \int_{\Omega} W \left(\frac{1}{\rho} \nabla P \right) d\Omega + \mathbf{F}$$

Using the identity demonstrated in Eq. 2.18, taking f to be the scalar function P ,

$$[2.51] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = - \int_{\Omega} \frac{1}{\rho} \{ \nabla [WP] - P \nabla W \} d\Omega + \mathbf{F}$$

$$[2.52] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = - \int_{\Omega} \frac{1}{\rho} \nabla [WP] d\Omega + \int_{\Omega} \frac{1}{\rho} P \nabla W d\Omega + \mathbf{F}$$

Using the divergence theorem (Eq. 2.19),

$$[2.53] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = - \int_S \frac{1}{\rho} W P \mathbf{n} dS + \int_{\Omega} \frac{1}{\rho} P \nabla W d\Omega + \mathbf{F}$$

Due to the compact support condition (Eq. 2.6), the value of the smoothing kernel W is zero on the surface of its influence domain and therefore the first integral on the RHS of Eq. 2.53 is zero.

$$[2.54] \quad \int_{\Omega} W \left(\frac{D\mathbf{u}}{Dt} \right) d\Omega = \int_{\Omega} \frac{1}{\rho} P \nabla W d\Omega + \mathbf{F}$$

Applying the concept of particle approximation, Eq. 2.54 becomes

$$[2.55] \quad \sum_{j=1}^N \frac{m_j}{\rho_j} \left(\frac{D\mathbf{u}_i}{Dt} \right) W(\mathbf{x}_i - \mathbf{x}_j, h) = \sum_{j=1}^N \frac{1}{\rho_i} \frac{m_j}{\rho_j} P_j \nabla_j W(\mathbf{x}_i - \mathbf{x}_j, h) + \mathbf{F}$$

Since $D\mathbf{u}_i/Dt$ is constant with respect to the summation over the neighbouring particles $j = 1, 2, \dots, N$, it can be excluded from the summation of the LHS of Eq. 2.55, and again taking $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h)$ for simplicity,

$$[2.56] \quad \frac{D\mathbf{u}_i}{Dt} \sum_{j=1}^N \frac{m_j}{\rho_j} W_{ij} = \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_j \nabla_j W_{ij} + \mathbf{F}$$

and due to the normalization condition (Eq. 2.4),

$$[2.57] \quad \frac{D\mathbf{u}_i}{Dt} = \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_j \nabla_j W_{ij} + \mathbf{F}$$

Because of the symmetry of the smoothing kernel (Eq. 2.9 and Eq. 2.25), Eq. 2.57 can be rewritten as:

$$[2.58] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_j \nabla_i W_{ij} + \mathbf{F}$$

Following a similar approach as before, since the gradient of a constant is equal to zero, we can deduce with Eq. 2.26 that

$$[2.59] \quad \sum_{j=1}^N \frac{m_j}{\rho_j} \nabla_i W_{ij} = 0$$

Multiplying this zero sum by P_i and $1/\rho_i$, and subtracting it to the LHS of Eq. 2.58 gives:

$$[2.60] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_j \nabla_i W_{ij} - P_i \frac{1}{\rho_i} \sum_{j=1}^N \frac{m_j}{\rho_j} \nabla_i W_{ij} + \mathbf{F}$$

Now since the pressure (P_i) and density (ρ_i) at particle i are constant with respect to the summation over the neighbouring particles $j = 1, 2, \dots, N$, they can be included in the sum of the second term on the RHS of Eq. 2.60.

$$[2.61] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_j \nabla_i W_{ij} - \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} P_i \nabla_i W_{ij} + \mathbf{F}$$

Rearranging Eq. 2.61 yields the final form of the conservation of momentum equation in SPH notation:

$$[2.62] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N m_j \frac{P_i + P_j}{\rho_i \rho_j} \nabla_i W_{ij} + \mathbf{F}$$

2.3. Numerical Aspects of SPH

In this section, the main numerical aspects of the SPH method are presented in order to better understand their influence on the outcome of the simulations. Although there is a wide variety of numerical schemes, the ones presented here are those that are implemented in SPHysics [Gómez-Gesteira *et al.*, 2010a]. Note that, when a reference is made to the actual SPHysics source code, the related information is presented as coded. This is because information presented in the user manual [Gómez-Gesteira *et al.*, 2010a] can sometimes differ from the actual code or omit some details.

2.3.1 Smoothing Kernels

There is a wide array of smoothing kernels available and the reader is referred to Liu and Liu [2003] and Gómez-Gesteira *et al.* [2010b] for more details. Two of the most popular ones are presented below.

Monaghan and Lattanzio [1985] proposed a smoothing kernel based on *cubic spline* functions, which is expressed as follows:

$$[2.63] \quad W(\mathbf{x}_i - \mathbf{x}_j, h) = \alpha_D \times \begin{cases} \frac{2}{3} - q^2 + \frac{1}{2}q^3 & 0 \leq q < 1 \\ \frac{1}{6}(2 - q)^3 & 1 \leq q < 2 \\ 0 & q \geq 2 \end{cases}$$

with

$$[2.64] \quad q = r_{ij}/h$$

where $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$; $\alpha_D = 15/(7\pi h^2)$ and $\alpha_D = 3/(2\pi h^3)$ in two- and three-dimensional space, respectively. The cubic spline smoothing kernel and its first derivative with respect to q are shown in Figure 2.3 and Figure 2.4, respectively.

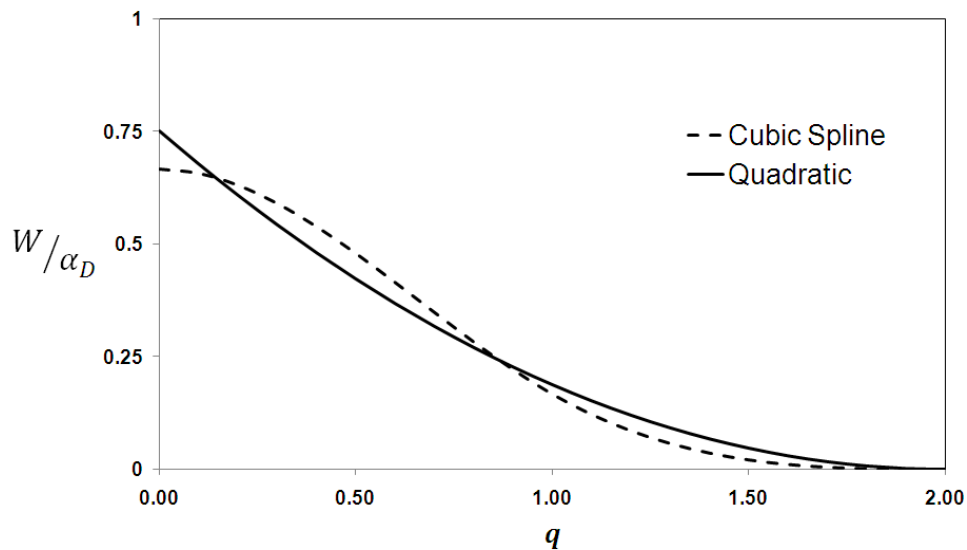


Figure 2.3. Cubic spline and quadratic smoothing kernels.

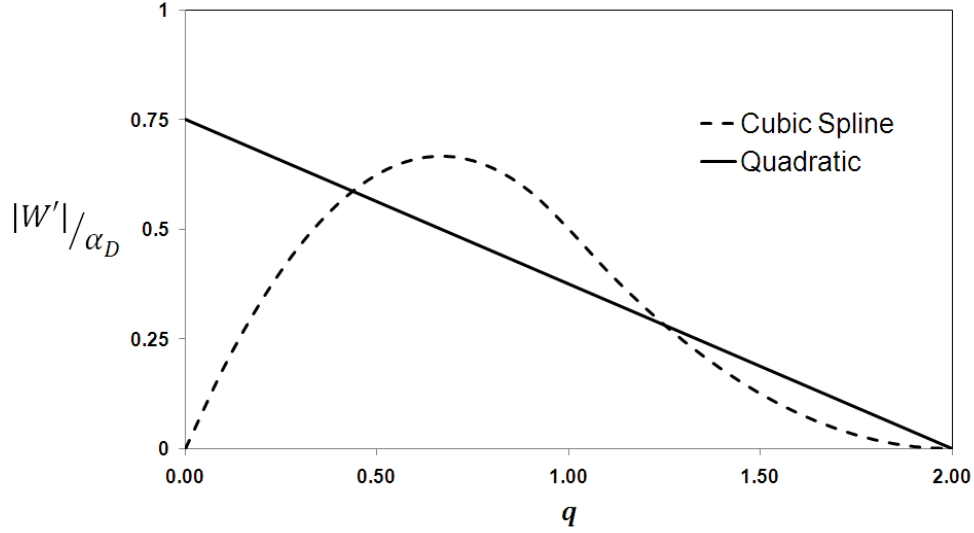


Figure 2.4. Absolute value of the first derivative of the cubic spline and quadratic smoothing kernels.

Johnson *et al.* [1996] proposed the *quadratic* smoothing kernel described as follows

$$[2.65] \quad W_{ij} = \alpha_D \left(\frac{3}{16} q^2 - \frac{3}{4} q + \frac{3}{4} \right) \quad 0 \leq q \leq 2$$

where $\alpha_D = 2/(\pi h^2)$ and $\alpha_D = 5/(4\pi h^3)$ in two- and three-dimensional space, respectively. The quadratic smoothing kernel and its first derivative with respect to q are shown in Figure 2.3 and Figure 2.4, respectively.

Unlike the cubic spline smoothing kernel, the absolute value of the first derivative of the quadratic smoothing kernel always increases as the distance between particles decreases (Figure 2.4), and therefore relieving the problem of *tensile instability* [Monaghan, 2000]. This so-called tensile instability manifests itself as particles clump unrealistically with each other, and it is due to the insufficient pressure responsible for keeping them apart. [Note that, the pressure term in the discretized momentum equation (Eq. 2.58) is multiplied by the first derivative of the smoothing kernel]. However, Monaghan [2000] proposed a correction for the tensile instability when using kernels of higher order such as the cubic spline smoothing kernel. Lastly, some researchers have

found that the accuracy of the SPH function approximation (Eq. 2.3) increases with the order of the smoothing kernel [Capone *et al.*, 2007], although this is accompanied with a longer computational time as a side effect.

2.3.2 Compressibility

With the compressible Navier-Stokes equation of continuity considered in this study (Eq. 2.32), the fluid is treated as slightly compressible. This approach, often referred to as *weakly compressible* SPH (WCSPH), allows the pressure terms of the discretized momentum equation (Eq. 2.62) to be calculated directly from the density of the fluid particles via an equation of state, rather than an additional PDE (i.e. a Poisson equation) that would have to be solved iteratively. In addition, solving a Poisson equation would require tracking the free-surface, which is not straightforward in most cases, especially the ones involving wave breaking [Gómez-Gesteira *et al.*, 2010b]. Originally given by Batchelor [1967] and adapted by Monaghan [1994], the equation of state used in this study is expressed as follows:

$$[2.66] \quad P_i = \frac{c_0^2 \rho_0}{\gamma} \left[\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right]$$

where $\gamma = 7$ is a constant, $\rho_0 = 1000 \text{ kg/m}^3$ is the reference water density, and c_0 is the speed of sound in water at the reference density. According to Dalrymple [2007], a value of $\gamma = 7$ fits oceanic data well, but Morris *et al.* [1997] suggested that γ is best taken as unity for low Reynolds number flows. Within this approach, the speed of sound at the reference density (c_0) is known to be a key factor that deserves careful consideration, and if the actual speed of sound is employed (e.g. 1480 m/s for water under standard pressure and temperature), the simulated fluid tends to become incompressible [Liu and Liu, 2003]. Generally, this speed of sound is considerably reduced in order to obtain reasonable time steps based on the Courant condition (see Section 2.3.4). However, the density fluctuations ($|\Delta \rho_i|/\rho_0$) should remain within 1% in order to obtain accurate results and, based on Eq. 2.67 below, this would imply that

c_0 remains about ten times greater than the maximum expected flow velocity [Monaghan, 1994, 2005].

$$[2.67] \quad \frac{|\Delta\rho_i|}{\rho_0} \approx \frac{\|\mathbf{u}_i\|^2}{c_0^2} = M^2$$

where M is the Mach number. Lastly, the speed of sound at a certain particle i is given by the derivative of the pressure with respect to density [e.g. Gómez-Gesteira *et al.*, 2010b]

$$[2.68] \quad c_i^2 = \frac{\partial P_i}{\partial \rho_i}$$

and substituting Eq. 2.66 into Eq. 2.68 followed by taking the derivative, yields

$$[2.69] \quad c_i^2 = c_0^2 \left(\frac{\rho_i}{\rho_0} \right)^{\gamma-1}$$

2.3.3 Viscosity and Turbulence

Two of the most frequently used formulations for the viscous diffusion terms of the momentum equation are the *artificial viscosity* [Monaghan, 1992] and the *laminar viscosity with Sub-Particle Scale (SPS) turbulence* [Lo and Shao, 2002; Dalrymple and Rogers, 2006].

In order to dissipate energy and to prevent fluid particles from moving chaotically and from colliding with each other, an artificial viscosity term Π_{ij} is added to the pressure terms of the momentum equation

$$[2.70] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N m_j \left(\frac{P_i + P_j}{\rho_i \rho_j} + \Pi_{ij} \right) \nabla_i W_{ij} + \mathbf{F}$$

$$[2.71] \quad \Pi_{ij} = \begin{cases} \frac{-\alpha_{visc} \bar{c}_{ij} \mu_{ij}}{\bar{\rho}_{ij}} & (\mathbf{u}_i - \mathbf{u}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) < 0 \\ 0 & (\mathbf{u}_i - \mathbf{u}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) \geq 0 \end{cases}$$

with

$$[2.72] \quad \mu_{ij} = \frac{h(\mathbf{u}_i - \mathbf{u}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^2 + 0.01h^2}$$

$$[2.73] \quad \bar{c}_{ij} = \frac{1}{2}(c_i + c_j)$$

$$[2.74] \quad \bar{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j)$$

where α_{visc} is a free parameter referred to as the artificial viscosity coefficient. It was noted that, in many cases where the particle resolution is high, this artificial viscosity is too dissipative [Dalrymple and Rogers, 2006].

In grid-based numerical methods, one popular approach for modeling turbulence is *Large Eddy Simulation* (LES). In this approach, the larger eddies which contain most of the turbulent kinetic energy are resolved directly by the grid while small eddies are modeled or represented by a *Sub-Grid Scale* (SGS) turbulence model. This is an alternative to computationally-expensive *Direct Numerical Simulation* (DNS) in which all scales of turbulence are modeled explicitly by ensuring a very fine grid resolution. For a detailed explanation of turbulence and LES, the reader is referred to Pope [2000]. Following the rationale of LES, Lo and Shao [2002] introduced a laminar viscosity with Sub-Particle Scale (SPS) turbulence formulation for modeling viscosity and turbulence within the SPH method. For this formulation, the momentum equation including the viscous diffusion terms is expressed as follows:

$$[2.75] \quad \frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \mathbf{u} + \frac{1}{\rho} \nabla \boldsymbol{\tau}$$

where ν is the kinematic viscosity of laminar flow ($10^{-6} \text{ m}^2/\text{s}$) and $\boldsymbol{\tau}$ is the SPS stress tensor. Discretized in SPH notation, the above equation becomes:

$$[2.76] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N m_j \left(\frac{P_i + P_j}{\rho_i \rho_j} \right) \nabla_i W_{ij} + \sum_{j=1}^N m_j \left(\frac{4\nu(\mathbf{x}_i - \mathbf{x}_j) \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)r_{ij}^2} \right) (\mathbf{u}_i - \mathbf{u}_j) \\ + \sum_{j=1}^N m_j \left(\frac{\boldsymbol{\tau}_i}{\rho_i^2} + \frac{\boldsymbol{\tau}_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} + \mathbf{F}$$

where $\boldsymbol{\tau}_i$ and $\boldsymbol{\tau}_j$ are the SPS shear stress tensors for particles i and j , respectively. For further details on the viscous and turbulence modeling considered in this study, the reader is referred to Gómez-Gesteira *et al.* [2010a].

2.3.4 Time-Stepping

As explained in Section 2.1.1, the original governing equations in PDE form are converted into ODE form through numerical discretization such that they can be solved using any stable time stepping algorithm for ODEs. Although there are a variety of ways to march the solution of the discretized governing equations ahead in time, two will be considered: a variation of the *predictor-corrector* algorithm [Monaghan, 1989] and the *symplectic* algorithm [Leimkuhler *et al.*, 1996]. Before describing these second order algorithms, the determination of the time step is explained.

The variable value of the time step (Δt) takes into account the Courant condition, the viscous diffusion, and the magnitude of the internal and external forces affecting each particle, such that

$$[2.77] \quad \Delta t = \alpha_{time} \times \min(\Delta t_f, \Delta t_{cu})$$

where Δt_f is the time step due to the internal and external force condition, Δt_{cu} is the time step due to the combined Courant and viscosity condition, and α_{time} is a free parameter usually taken as 0.3 and referred to as the time step coefficient. Throughout the simulation, a new time step value is determined at the end of each iteration. The internal

and external force condition ensures that particles do not get too close to each other during the integration of their movement [Morris *et al.*, 1997] and it is expressed as:

$$[2.78] \quad \Delta t_f = \min_i \sqrt{\frac{h}{\|\mathbf{f}_i\|}}$$

where h is the smoothing length, $\mathbf{f}_i$ is the total internal and external forces per unit mass associated with particle i (i.e. the value of the right hand side of the momentum equation), and the *min* operator takes into account all particles used in the simulation. On the other hand, the Courant condition states that the time step should be less than the time required for a sound wave, or any wave, to travel between adjacent particles, and the viscosity condition states that the time step should be less than the time scale of the viscous phenomenon. According to Monaghan and Kos [1999], the combined Courant and viscosity condition is as follows:

$$[2.79] \quad \Delta t_{cu} = \min_i \left(\frac{h}{c_i + \max_j \left| \frac{h \mathbf{u}_{ij} \cdot \mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|^2} \right|} \right)$$

where $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ is the difference in velocity between particles i and j , c_i is the speed of sound at particle i calculated according to Eq. 2.69, $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ is the vector between particles i and j , and the *min* operator takes into account all particles in the simulation, while the *max* operator takes into account all neighbouring particles j of a particle i .

The predictor-corrector algorithm implemented in SPHysics is as follows [Gómez-Gesteira *et al.*, 2010a]. The derivative of the density $D\rho_i^n/Dt$ and the acceleration $D\mathbf{u}_i^n/Dt$ of a particle i are first calculated using the mass and momentum conservation equations (Eq. 2.45 and Eq. 2.62), respectively. The values of the density,

velocity and position of the particle are then evaluated at half the time step as follows, respectively:

$$[2.80] \quad \rho_i^{n+1/2} = \rho_i^n + \frac{\Delta t}{2} \frac{D\rho_i^n}{Dt}$$

$$[2.81] \quad \mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n + \frac{\Delta t}{2} \frac{D\mathbf{u}_i^n}{Dt}$$

$$[2.82] \quad \mathbf{x}_i^{n+1/2} = \mathbf{x}_i^n + \frac{\Delta t}{2} \mathbf{u}_i^n$$

The value of pressure at half the time step $P_i^{n+1/2}$ is then calculated using the equation of state (Eq. 2.66) with $\rho_i^{n+1/2}$. This is known as the *predictor* stage. Afterwards, the derivative of the density $D\rho_i^{n+1/2}/Dt$ and the acceleration $D\mathbf{u}_i^{n+1/2}/Dt$ are re-evaluated at half the time step and the values of the density, velocity and position of the particle at half the time step are then corrected as follows, respectively:

$$[2.83] \quad \rho_i^{n+1/2} = \rho_i^n + \frac{\Delta t}{2} \frac{D\rho_i^{n+1/2}}{Dt}$$

$$[2.84] \quad \mathbf{u}_i^{n+1/2} = \mathbf{u}_i^n + \frac{\Delta t}{2} \frac{D\mathbf{u}_i^{n+1/2}}{Dt}$$

$$[2.85] \quad \mathbf{x}_i^{n+1/2} = \mathbf{x}_i^n + \frac{\Delta t}{2} \mathbf{u}_i^{n+1/2}$$

This is known as the *corrector* stage. Finally, the values of the density, velocity and position of the particle are evaluated at the end of the time step according to the following, respectively:

$$[2.86] \quad \rho_i^{n+1} = 2\rho_i^{n+1/2} - \rho_i^n$$

$$[2.87] \quad \mathbf{u}_i^{n+1} = 2\mathbf{u}_i^{n+1/2} - \mathbf{u}_i^n$$

$$[2.88] \quad \mathbf{x}_i^{n+1} = 2\mathbf{x}_i^{n+1/2} - \mathbf{x}_i^n$$

Lastly, the pressure at the end of the time step P_i^{n+1} is calculated using the equation of state (Eq. 2.66) with ρ_i^{n+1} .

The symplectic algorithm is similar to the predictor-corrector algorithm with the exceptions that the values are not corrected at half the time step and the final evaluation at the end of the time step is different. So, just like the predictor-corrector algorithm, the derivative of the density $D\rho_i^n/Dt$ and the acceleration $D\mathbf{u}_i^n/Dt$ of a particle i are first calculated using the mass and momentum conservation equations (Eq. 2.45 and Eq. 2.62), respectively. The values of the density $\rho_i^{n+1/2}$, velocity $\mathbf{u}_i^{n+1/2}$ and position $\mathbf{x}_i^{n+1/2}$ of the particle are then evaluated at half the time step according to Eq. 2.80, Eq. 2.81 and Eq. 2.82, respectively. The value of pressure at half the time step $P_i^{n+1/2}$ is then calculated using the equation of state (Eq. 2.66) with $\rho_i^{n+1/2}$. According to the implementation of SPHysics (i.e. the open source code), the values of the density, velocity and position of the particle are evaluated at the end of the time step according to the following, respectively:

$$[2.89] \quad \rho_i^{n+1} = \rho_i^n \times \frac{2 + \varepsilon_\rho \Delta t}{2 - \varepsilon_\rho \Delta t}$$

$$[2.90] \quad \mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t \frac{D\mathbf{u}_i^{n+1/2}}{Dt}$$

$$[2.91] \quad \mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \frac{\Delta t}{2} \left(\mathbf{u}_i^n + \mathbf{u}_i^{n+1/2} \right)$$

with

$$[2.92] \quad \varepsilon_\rho = \frac{1}{\rho_i^{n+1/2}} \frac{D\rho_i^{n+1/2}}{Dt}$$

Lastly, the pressure at the end of the time step P_i^{n+1} is calculated using the equation of state (Eq. 2.66) with ρ_i^{n+1} .

2.3.5 Correction Schemes

Several correction schemes are often required to keep SPH simulations accurate and more stable. Some of these are presented in the following subsections.

Kernel Renormalization

In the simulation of free-surface flows, the influence domain of the smoothing kernel of a particle can be truncated at a boundary of the computational domain or at the free-surface, as shown in Figure 2.5.

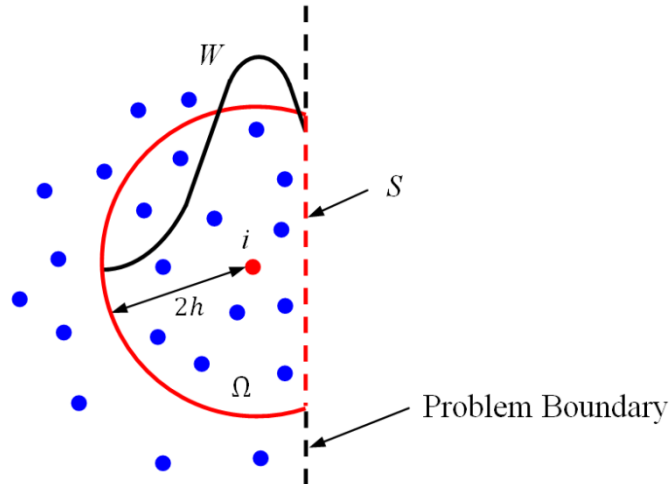


Figure 2.5. Truncated influence domain Ω with radius $2h$ of smoothing kernel W .

In this particular case, the normalization condition (Eq. 2.4) is not fulfilled, and the surface integral on the RHS of Eq. 2.16 is not equal to zero since the value of the smoothing kernel W is not zero on its entire surface S . Consequently, this makes Eq. 2.17 invalid. If not considered, this situation can introduce errors in the simulations, essentially at the problem boundaries and at the free-surface. The following are two correction schemes to handle this situation.

To correct the evaluation of a vector function $\mathbf{f}(\mathbf{x}_i)$, or scalar variable, Bonet and Lok [1999] proposed the following relationship:

$$[2.93] \quad \langle \mathbf{f}(\mathbf{x}_i) \rangle = \sum_{j=1}^N \frac{m_j}{\rho_j} \mathbf{f}(\mathbf{x}_j) W_{ij} \bigg/ \sum_{j=1}^N \frac{m_j}{\rho_j} W_{ij}$$

which is in fact Eq. 2.22 divided by the equation of the normalization condition (Eq. 2.4) in discretized form. If the influence domain of the smoothing kernel is not truncated, Eq. 2.93 is equal to Eq. 2.22, since the normalization condition is satisfied (i.e. Eq. 2.4 is equal to one).

The second correction scheme consists in the correction of the gradient of the smoothing kernel. According to Gómez-Gesteira *et al.* [2010a], the corrected gradient of the smoothing kernel $\widetilde{\nabla} W_{ij}$ for a particular particle i can be obtained as follows:

$$[2.94] \quad \widetilde{\nabla} W_{ij} = L_i \nabla W_{ij}$$

where the matrix L_i is given by

$$[2.95] \quad L_i = M_i^{-1}$$

with

$$[2.96] \quad M_i = \sum_{j=1}^N \frac{m_j}{\rho_j} \nabla W_{ij} \otimes (\mathbf{x}_i - \mathbf{x}_j)$$

To apply this correction, the corrected gradient of the smoothing kernel (given by Eq. 2.94) is substituted in the conservation equations (Eq. 2.45 and Eq. 2.62). One disadvantage of this correction method is that the matrix M_i must be evaluated and inverted for each particle in the simulation prior to the evaluation of the conservation equations. This requires performing an additional loop over each particle and their associated neighbouring particles, which increases computational time.

Density Filters and Riemann Solver

While the kinematics of SPH simulations are generally realistic, the pressure field of the particles can exhibit large pressure oscillations, which can be particularly important near the free-surface and near the boundaries of the problem domain [Gómez-Gesteira *et al.*, 2010b]. Since an accurate pressure field is essential for predicting the forces acting on obstacles, various schemes have been developed to address this issue.

One straightforward approach is to apply a filter to the density of the particles (or *density filter*) and re-assign an “averaged” density to each of these particles [Colagrossi and Landrini, 2003]. Recall that the pressure field is calculated directly from the density field via the equation of state (Eq. 2.66) and hence, a smoother density field will result in a smoother pressure field. Based on Eq. 2.22, with the density as the scalar function to be approximated, two orders of correction can be used within this approach:

1 - Zero-Order (Shepard Filter)

$$[2.97] \quad \rho_i^{new} = \sum_{j=1}^N \frac{m_j}{\rho_j} \rho_j \tilde{W}_{ij} = \sum_{j=1}^N m_j \tilde{W}_{ij}$$

with

$$[2.98] \quad \tilde{W}_{ij} = W_{ij} / \sum_{j=1}^N \frac{m_j}{\rho_j} W_{ij}$$

2 - First Order (Moving-Least-Squares)

$$[2.99] \quad \rho_i^{new} = \sum_{j=1}^N \frac{m_j}{\rho_j} \rho_j W_{ij}^{MLS} = \sum_{j=1}^N m_j W_{ij}^{MLS}$$

with (in 3D)

$$[2.100] \quad W_{ij}^{MLS} = [\Psi_0 + \Psi_1(x_i - x_j) + \Psi_2(y_i - y_j) + \Psi_3(z_i - z_j)]W_{ij}$$

where

$$[2.101] \quad \Psi = \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix} = [A_{ij}]^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$[2.102] \quad [A_{ij}] = \sum_{j=1}^N [\tilde{A}_{ij}] W_{ij}$$

$$[2.103] \quad [\tilde{A}_{ij}] = \begin{bmatrix} 1 & (x_i - x_j) & (y_i - y_j) & (z_i - z_j) \\ (x_i - x_j) & (x_i - x_j)^2 & (x_i - x_j)(y_i - y_j) & (x_i - x_j)(z_i - z_j) \\ (y_i - y_j) & (x_i - x_j)(y_i - y_j) & (y_i - y_j)^2 & (y_i - y_j)(z_i - z_j) \\ (z_i - z_j) & (x_i - x_j)(z_i - z_j) & (y_i - y_j)(z_i - z_j) & (z_i - z_j)^2 \end{bmatrix}$$

where x , y and, z are the components in the X, Y and Z direction of a position vector $\mathbf{x}$, respectively. Lastly, the above density filters are generally applied every 20 to 50 time steps, as required [Gómez-Gesteira *et al.*, 2010b].

Another correction scheme for removing large pressure fluctuations or discontinuities between two adjacent particles is to consider a Riemann problem as they interact with one another. By using this approach, the WCSPH model becomes *Godunov-*

based and, following the work of Parshikov [1999] and Cha and Whitworth [2003], the discretized conservation equations (Eq. 2.45 and Eq. 2.62) are reformulated as follows:

$$[2.104] \quad \frac{D\rho_i}{Dt} = -2\rho_i \sum_{j=1}^N \frac{m_j}{\rho_j} (\mathbf{u}_{ij}^* - \mathbf{u}_i) \cdot \nabla_i W_{ij}$$

$$[2.105] \quad \frac{D\mathbf{u}_i}{Dt} = -2 \sum_{j=1}^N m_j \left(\frac{P_{ij}^*}{\rho_i \rho_j} \right) \nabla_i W_{ij} + \mathbf{F}$$

where $\mathbf{u}_{ij}^*$ and P_{ij}^* are the solution of the Riemann problem between particles i and j for the velocity and pressure, respectively. Another Riemann-adapted SPH formulation of the conservation equations is provided by Vila [1999]. Within SPHysics [Gómez-Gesteira *et al.*, 2010a], the Riemann problem is solved using a Harten – Lax – van Leer – Contact (HLLC) approximate *Riemann solver* [Toro *et al.*, 1994] using a Monotone Upwind-centred Scheme for Conservation Laws (MUSCL) [Toro, 2001] with a general β -limiter [Hirsch, 1998]. A detailed description of this Riemann solver is provided in Appendix A. In their SPH simulations of the interaction of waves with a partially-submerged cylinder, Omidvar *et al.* [2012] observed that the numerical results are very sensitive to the value of the slope limiter (β -limiter). As observed in other numerical methods that utilize MUSCL-based β -limiters, such as FVM, a low value (i.e. $\beta = 1$), also referred to as a *Minmod* limiter, will give diffusive results or decay whereas a higher value (i.e. $\beta = 2$), also referred to as a *Superbee* limiter, can lead to unrealistic increases in wave amplitude. Based on this, Omidvar *et al.* [2012] implemented a sponge layer for the dissipation of waves, in order to avoid unwanted reflection at the limits of their computational domain by gradually decreasing the value of the slope limiter to zero in this region. In addition, as the particle resolution increases (i.e. the number of particles increases and the distance separating them decreases), the smoothing length (h) will approach zero and the value of the β -limiter should approach unity. Lastly, according to Dr. Benedict Rogers, one of the developers of SPHysics, [e-mail conversation, 2012] this Godunov-based scheme can

lead to a high level of “numerical viscosity” if the particle resolution is not sufficiently high (with respect to a low β -limiter value).

Particle Movement

In order to keep particles moving in an orderly manner, Monaghan [1989, 1992] proposed a correction for updating the position of a particle i . This correction is commonly referred to as the *XSPH* correction and is expressed as:

$$[2.106] \quad (\mathbf{u}_i)_{XSPH} = \mathbf{u}_i + \varepsilon \sum_{j=1}^N \frac{m_j}{\frac{1}{2}(\rho_i + \rho_j)} (\mathbf{u}_i - \mathbf{u}_j) W_{ij}$$

where ε is a coefficient, hardcoded as 0.5 within the implementation of SPHysics. With this correction, instead of moving according to its own velocity only, a particle’s movement is influenced by the average velocity of its neighbouring particles. It is this corrected velocity that is used when updating the position $\mathbf{x}_i$ of the particle within the time-stepping algorithm. Although the rationale for the selection of ε in Eq. 2.106 is not well explained in the literature, Monaghan [1994] selected the value $\varepsilon = 0.5$ for the simulation of free-surface flows, whereas Liu and Liu [2003] mention that $\varepsilon = 0.3$ is a good choice for the simulation of incompressible flows. Surprisingly, Didier and Neves [2010] reported that the use of the XSPH correction within the SPHysics implementation resulted in unphysical propagation behaviour as they simulated the interaction of waves with coastal structures.

2.3.6 Boundary Condition

Due to its origin in astrophysical problems, where boundaries are generally non-existent, the formulation of the SPH method can be problematic when applied to bounded problems. Take for example the issue related to the truncated smoothing kernel discussed in Section 2.3.5. In addition, the behaviour of water particles at boundaries needs to be defined artificially where it can be challenging to simulate realistic velocity and pressure fields. Although several types and variations of boundary treatment have been formulated

(the reader is referred to Gómez-Gesteira *et al.* [2010b] for an extensive review of these boundary conditions), they generally consist in modeling the boundaries of the computational domain as discrete particles that interact in a specific way with the fluid particles. These particles are often referred to as *boundary particles* and it is worth mentioning that these particles are not necessarily fixed, as their motion can be imposed (e.g. wavemaker), or be governed by laws of vector mechanics (e.g. floating objects), as described in Monaghan [2005]. The two types of boundary condition implemented in SPHysics are described below.

Repulsive Boundary Particles

According to this boundary treatment, the boundary particles placed in a single row (in 2-D) or in a layer (in 3-D) will exert a *repulsive* force on the fluid particles (in analogy to the Lennard-Jones forces between atoms in molecular dynamics). Such an idea was first introduced by Monaghan [1994], where the magnitude of the central repulsive force per unit mass on fluid particle i due to boundary particle k is given by

$$[2.107] \quad f_{ik} = \frac{D}{r_{ik}} \left[\left(\frac{\Delta}{r_{ik}} \right)^{12} - \left(\frac{\Delta}{r_{ik}} \right)^6 \right]$$

where r_{ik} is the distance separating the fluid particle i and the boundary particle k , Δ is the initial inter-particle spacing, and D is a coefficient based on the particular physical configuration. According to Monaghan [1994], for problems involving dam-breaks, bores or weirs with fluid depth H , D is typically taken as gH/h , but the results are insensitive to its exact value if sufficiently large. The central repulsive force due to a single boundary particle (described by Eq. 2.107) is depicted in Figure 2.6. A drawback of the central repulsive force given by Eq. 2.107 is that the large variation in the total force acting on a fluid particle moving parallel to the boundary causes large disturbances in the flow [Issa, 2004]. However, a smaller spacing between the boundary particles can mitigate these disturbances. According to Monaghan and Kos [1999], these disturbances in the flow are equivalent to a corrugated boundary with ripples on the scale of the spacing between the boundary particles.

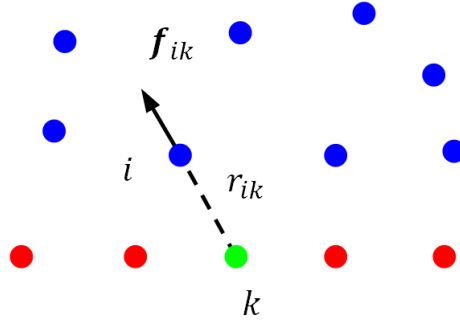


Figure 2.6. Central repulsive force $\mathbf{f}_{ik}$ on fluid particle i due to boundary particle k .

In order to address this issue, Monaghan and Kos [1999] suggest an approach in which an interpolation procedure would be used so that the resultant force acting on a fluid particle due to its neighbouring boundary particles would be normal to the boundary. In other words, the force per unit mass $\mathbf{f}_{ik}$ on a fluid particle i due to a boundary particle k is computed using the components of their separation along the normal and the tangent directions, such that:

$$[2.108] \quad \mathbf{f}_{ik} = \mathbf{n}R(|\mathbf{n} \cdot \mathbf{r}_{ik}|)P(|\mathbf{t} \cdot \mathbf{r}_{ik}|)$$

where $\mathbf{n}$ is the unit vector normal to the boundary and $\mathbf{t}$ is the unit vector tangent to the boundary at the location of boundary particle k , $\mathbf{r}_{ik}$ is the vector separating the fluid particle i and the fluid particle k , $R(|\mathbf{n} \cdot \mathbf{r}_{ik}|)$ is a function designed to increase as the normal distance separating the fluid particle i and the boundary particle k decreases, and $P(|\mathbf{t} \cdot \mathbf{r}_{ik}|)$ is a function designed to ensure that, as the fluid particle i moves between two adjacent boundary particles, the force contributions of the boundary particles will combine to make the boundary force constant if the fluid particle moves parallel to the solid boundary. The normal repulsive force described by Eq. 2.108 due to a single boundary particle is depicted in Figure 2.7.

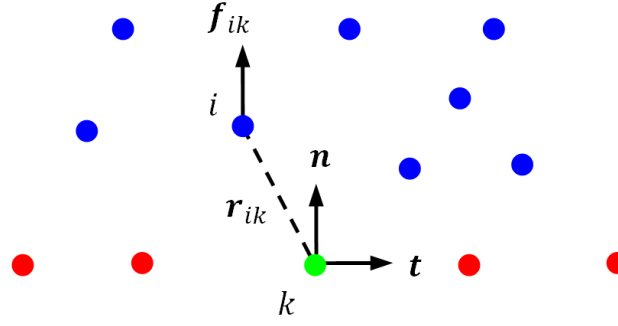


Figure 2.7. Normal repulsive force $\mathbf{f}_{ik}$ on fluid particle i due to boundary particle k .

In Eq. 2.108, the functions $R(|\mathbf{n} \cdot \mathbf{r}_{ik}|)$ and $P(|\mathbf{t} \cdot \mathbf{r}_{ik}|)$ are described as follows:

$$[2.109] \quad R(|\mathbf{n} \cdot \mathbf{r}_{ik}|) = \begin{cases} A \frac{1}{\sqrt{\varphi}} (1 - \varphi) & \varphi < 1 \\ 0 & \varphi \geq 1 \end{cases}$$

$$[2.110] \quad P(|\mathbf{t} \cdot \mathbf{r}_{ik}|) = \frac{1}{2} \left[1 + \cos \left(\frac{\pi |\mathbf{t} \cdot \mathbf{r}_{ik}|}{\Delta} \right) \right]$$

with

$$[2.111] \quad A = \frac{1}{h} \{ 0.01 c_i^2 + \xi c_i [(\mathbf{u}_i - \mathbf{u}_k) \cdot \mathbf{n}] \}$$

and

$$[2.112] \quad \varphi = \frac{|\mathbf{n} \cdot \mathbf{r}_{ik}|}{2\Delta}$$

where c_i is the speed of sound of particle i (as calculated by Eq. 2.69), ξ is a constant equal to 1 if particles i and k are approaching and equal to zero otherwise, $\mathbf{u}_i$ and $\mathbf{u}_k$ are the velocity vector of the fluid particle i and boundary particle k , respectively.

This approach was later modified by Rogers and Dalrymple [2008] for the numerical simulation of tsunami waves generated by landslides. Their modification consisted in the multiplication of an additional term that would adjust the magnitude of the normal repulsive force according to the local water depth and velocity of the fluid particle normal to the boundary. With this modification, Eq. 2.108 is rewritten as follows:

$$[2.113] \quad \mathbf{f}_{ik} = \mathbf{n}R(|\mathbf{n} \cdot \mathbf{r}_{ik}|)P(|\mathbf{t} \cdot \mathbf{r}_{ik}|)\varepsilon(z, u_{\perp})$$

with

$$[2.114] \quad \varepsilon(z, u_{\perp}) = \varepsilon(z) + \varepsilon(u_{\perp})$$

and

$$[2.115] \quad \varepsilon(z) = \begin{cases} 0.02 & z \geq 0 \\ |z/d| + 0.02 & 0 > z \geq -d \\ 1 & |z/d| > 1 \end{cases}$$

and

$$[2.116] \quad \varepsilon(u_{\perp}) = \begin{cases} 0 & u_{\perp} \geq 0 \\ \frac{|20u_{\perp}|}{c_0} & |20u_{\perp}| < c_0 \\ 1 & |20u_{\perp}| \geq c_0 \end{cases}$$

where z is the elevation above the still water level d , c_0 is the speed of sound in water at the reference density (as discussed in Section 2.3.2), and the velocity of the fluid particle normal the boundary $u_{\perp}$ is given by

$$[2.117] \quad u_{\perp} = (\mathbf{u}_i - \mathbf{u}_k) \cdot \mathbf{n}$$

This latter version of the repulsive boundary treatment (Eq. 2.113) is the one coded in the open source SPHysics implementation [Gómez-Gesteira *et al.*, 2010a]. It should be noticed here that this version seems to be highly adapted to problems pertaining to ocean waves, since it involves the concept of still water level. In SPHysics, this still water level (d) is an input parameter and it is not clear how it should be selected when simulating problems not related to ocean waves (e.g. dam-break).

It can be seen (Figure 2.7) that the repulsive force acting on a fluid particle i , due to a boundary particle k , has no tangential component with respect to the boundary. According to this, the repulsive force given by Eq. 2.108 or Eq. 2.113 corresponds to the *slip* condition, where the velocities of fluid particles at the boundary are not necessarily zero. In order to promote a *no-slip* condition, an additional tangential force component should be introduced. When using the laminar viscosity with the Sub-Particle Scale (SPS) turbulence model, as previously described in Section 2.3.3, this additional force component (shown in Figure 2.8), is expressed as follows (according to the SPHysics source code):

$$[2.118] \quad \mathbf{f}_{ik,wall\,visc} = -v_{wall} \frac{m_i (\mathbf{x}_i - \mathbf{x}_k) \cdot \nabla_i W_{ik}}{\rho_i |\mathbf{x}_i - \mathbf{x}_k|^2} (\mathbf{u}_i - \mathbf{u}_k)$$

where v_{wall} is a free parameter and is referred to as the *wall viscosity* within the SPHysics implementation. Nevertheless, a closer inspection of Eq. 2.118 reveals that this force is not solely tangential to the solid boundary but is in fact aligned with but opposite to the vector $(\mathbf{u}_i - \mathbf{u}_k)$.

Finally, the repulsive boundary treatment is incorporated in the momentum equation (Eq. 2.62) as follows

$$[2.119] \quad \frac{D\mathbf{u}_i}{Dt} = - \sum_{j=1}^N m_j \left(\frac{P_i + P_j}{\rho_i \rho_j} \right) \nabla_i W_{ij} + \sum_{k=1}^B \mathbf{f}_{ik} + \sum_{k=1}^B \mathbf{f}_{ik,wall\,visc} + \mathbf{F}$$

where B is the total number of neighbouring boundary particles within the influence domain of fluid particle i .

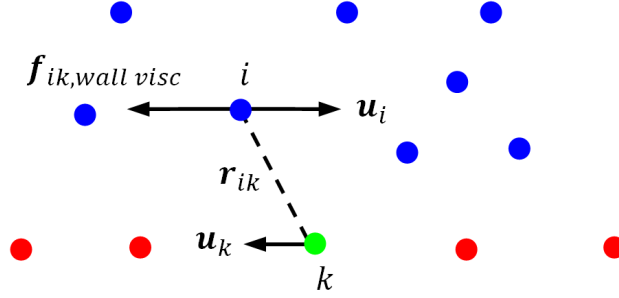


Figure 2.8. Wall friction force $f_{ik,wall\ visc}$ on fluid particle i due to boundary particle k .

Dynamic Boundary Particles

Within this approach, boundary particles are treated as regular fluid particles and they follow the same equations as these ones [Dalrymple and Knio, 2001]. Thus, these *dynamic* boundary particles follow the continuity equation (Eq. 2.45), the momentum equation (Eq. 2.62), and the equation of state (Eq. 2.69), and they share the same physical properties as the regular fluid particles. However, as they constitute solid boundaries, these particles remain fixed or move according to some externally imposed motion (e.g. moving objects such as wavemakers). The repulsion mechanism of this boundary condition works in the same way as if any other fluid particles would get closer to one another. When a fluid particle approaches a boundary, the density of the particles forming this one will increase according to the continuity equation (Eq. 2.45). Subsequently the pressure of the boundary particles will also increase according to the equation of state (Eq. 2.66), and because of the pressure term of the momentum equation (Eq. 2.62), the fluid particles will be repulsed by the boundary particles and hence kept within the computational domain. In order to further prevent the penetration of regular fluid particles through the boundary dynamic particles, these last ones are placed in a staggered manner as shown in Figure 2.9.

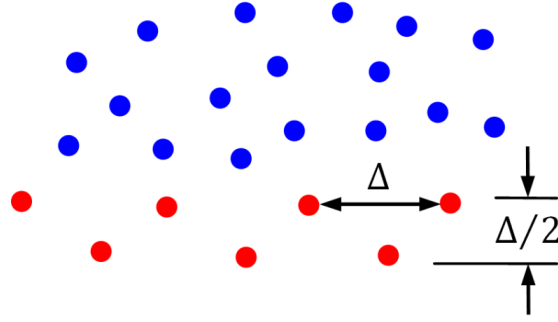


Figure 2.9. Spatial arrangement of dynamic boundary particles where Δ is the initial inter-particle spacing (based on Gómez-Gesteira *et al.* [2010a]).

Finally, the advantages of the dynamic boundary condition are its computational simplicity. This is because the properties of the boundary particles are calculated in the same loops as the fluid particles. Also, it represents a no-slip condition as the velocity of boundary particles, which contributes in the calculation of the velocity of nearby fluid particles, is set to zero.

2.3.7 Execution Flow Chart

To better understand the SPH formulation presented in this chapter, Figure 2.10 shows a flow chart for the general execution of SPHysics code. The predictor-corrector time-stepping algorithm (Section 2.3.4) is considered and t_{max} stands for the end time of the simulation.

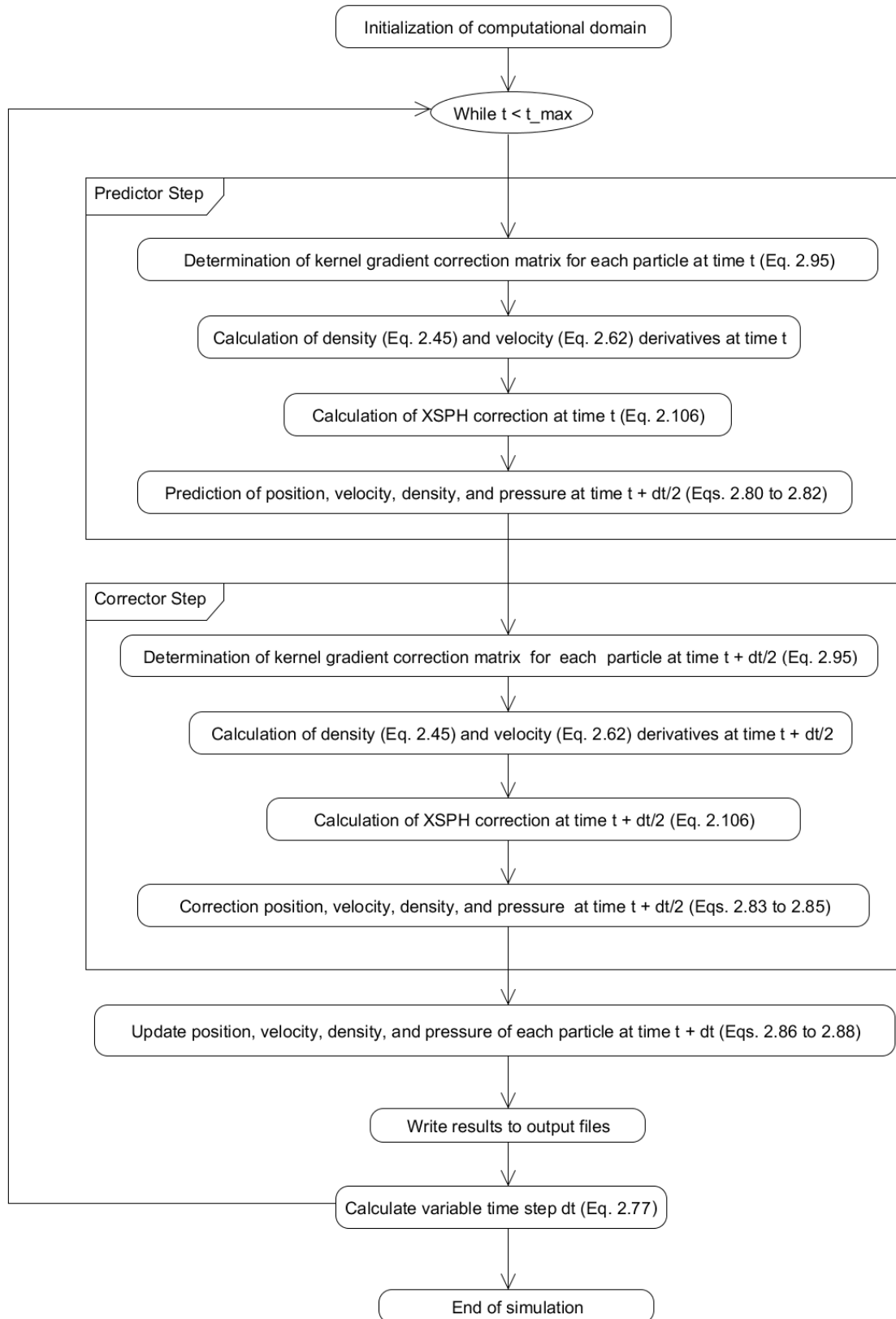


Figure 2.10. General execution flow chart of SPHysics code.

Chapter 3. Literature review

The literature review presented in this chapter focuses on past research contributions dealing with bore-structure interaction. First, studies that demonstrate the possible approaches for reproducing tsunami waves experimentally are reviewed. The different behaviours of tsunami waves at the shoreline are also discussed in order to compare the applicability of these approaches. Next, the results of experimental studies which investigate the influence of the bed condition on the propagation of hydraulic bores are presented. Then, results of current experimental-based research, concerned with the impact of hydraulic bores on structures, are reviewed and lastly, related numerical studies on such impacts, performed using the SPH method, are described. The main objectives of the literature review are to: (i) provide a background on concepts related to the dam-break scenario used to reproduce tsunami-like hydraulic bores, (ii) provide a starting point for the interpretation of the results presented in this thesis, and (iii) demonstrate the novelty of the latter with respect to similar studies performed using the SPH method.

3.1. Experimental Modeling of Tsunami Bores and Dam-Break Waves

Before describing the approaches for simulating them experimentally, it is worth explaining how actual tsunami waves behave as they approach a coastline. Taking the 2011 Tohoku Tsunami as an example, as it approached a large section of the north-east Japanese coastline the observed behaviour of its waves varied considerably, depending on local geographical features. The affected areas can generally be categorized as either *coastal rias* (Figure 1.3), which are lower river valleys previously submerged by the sea, where the tsunami height reached over 15m with a maximum runup of 38.2m, or *coastal plains* (Figure 1.1) characterized by the presence of sandy beaches, where the tsunami inundation waves traveled several kilometres inland with depths ranging from 5 to 10m [Mikami *et al.*, 2012]. Based on these differences in the coastline, Takahashi *et al.* [2011]

classified the inland intrusions of the 2011 Tohoku Tsunami into the following four types (Figure 3.1).

- *Breaking Wave* type: The tsunami wave breaks near the shoreline of a typical mild-sloping sandy beach, runs up the sand dune, and propagates several kilometers over low-lying land (Figure 3.1-a).
- *High Runup* type: The tsunami wave breaks inland on a relatively steep slope on which it rapidly reaches significant elevations afterwards (Figure 3.1-b).
- *Slowly-Varying* type: Because of the presence of steep cliffs with deep sea fronts, the tsunami wave does not break. The water surface simply moves up and down relatively smoothly (Figure 3.1-c).
- *Overtopping* type: In ports, where the water is deep in order to allow safe navigation, the tsunami wave overtops the quaywalls without breaking and intrudes inland at particularly high velocities (Figure 3.1-d).

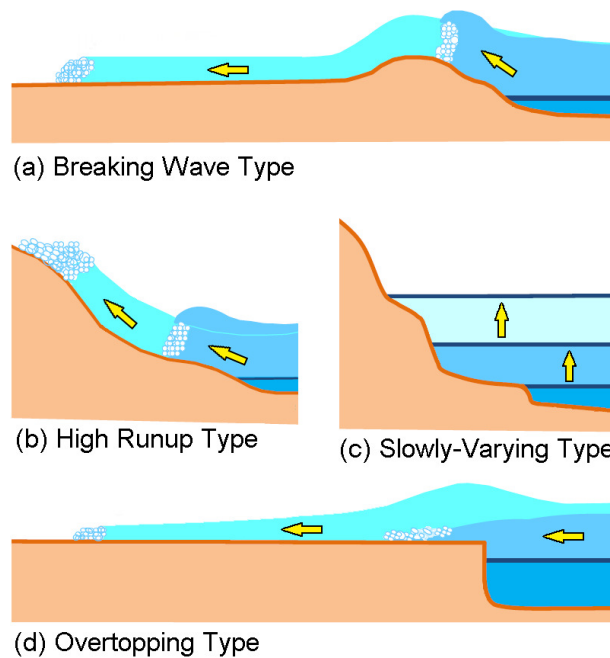


Figure 3.1. Types of inland intrusions of a tsunami [adapted from Takahashi *et al.*, 2012].

One conventional approach for reproducing tsunami waves experimentally is to generate a solitary wave inside a flume or wave basin. A benchmark example is the physical experiment of Briggs *et al.* [1995] – well known as the *conical island* experiment – and inspired by the tsunami of December 12, 1992, which caused over 750 deaths on Babi Island, Indonesia. In this experiment, solitary waves with heights (H) varying from 0.016 to 0.046m were generated by the single stroke of a piston-type wavemaker inside a 25.0m long $\times$ 30.0m wide $\times$ 0.6m deep wave basin, with still water depths (d) of 0.32 and 0.42m. The purpose of this experiment was to investigate the runup behaviour of a tsunami wave as it refracts around a conical island and converges on the backside of the latter. With a different approach, Monaghan and Kos [1999] studied (both experimentally and numerically using the SPH method) the runup of a solitary wave on a typical Cretan beach. The goal of their study was to investigate the effects of the tsunami generated by the volcanic eruption that occurred on the island of Santorini in the Aegean Sea (~1500 BC). In their physical experiment, a solitary wave ($H = 0.088\text{m}$) was generated by dropping a weighted box into still water ($d = 0.21\text{m}$) at the upstream end of a 9.00m long $\times$ 0.40m wide wave flume. Regardless of how a tsunami-like solitary wave is generated experimentally, the transformation it will undergo as it reaches and propagates over the modeled coastline will mainly depend on the geometry of the latter.

One relatively simple approach for reproducing a tsunami-induced hydraulic bore experimentally is to generate a dam-break wave by suddenly releasing an impounded volume of water from an upstream reservoir (depth h_u) into a straight channel (depth h_d), as conceptualized in Figure 3.2. The water is usually released suddenly by means of a vertical gate, after which the dam-break wave propagates downstream while a negative wave propagates upstream. Unless mentioned otherwise, the time origin considered throughout this thesis corresponds to the sudden release of the water in the reservoir. Chanson [2006] analysed visual images of an actual tsunami bore resulting from the 2004 Indian Ocean Tsunami and demonstrated that the flow characteristics of the analyzed bore were very similar to those of a hydraulic bore corresponding to a dam-break wave. It should be noted that the analysis he performed was for a tsunami bore propagating over a dry surface within a coastal plain area, i.e. corresponding to an inland intrusion of the

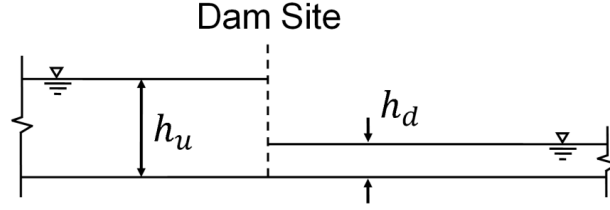


Figure 3.2. Schematic of the dam-break considered for the generation of tsunami-like bores.

Breaking Wave type described previously (Figure 3.1-a). This type of inland intrusion can clearly be observed in Figure 1.1 as the first wave of the 2011 Tohoku Tsunami breaks on the shoreline and subsequently propagates over the coastal plain in the form of a hydraulic bore.

For solving the dam-break problem, considering both the *dry*- ($h_d = 0$) and *wet*- ($h_d > 0$) bed conditions, Stoker [1957] proposed an analytical solution of the shallow water equations (SWE) using the method of characteristics. The SWE are a simplified form of the incompressible Navier-Stokes equations, for which viscosity and turbulence are neglected and for which hydrostatic pressure and uniform fluid velocity over depth are assumed. This analytical solution will subsequently be used in this thesis for comparisons with numerical results. Figure 3.3 shows a definition sketch for the propagation of dam-break waves according to the SWE, where, in the case of the wet-bed condition (Figure 3.3-a), four different zones can be indentified in the fluid at any time t after the instantaneous removal of the “dam”. Zone d is the quiet downstream region which is terminated on the upstream by the shock wave (i.e. the bore-front), zone u is the undisturbed upstream region, zone 2 is a zone of constant state in which the water is not at rest, and zone 3 is a rarefaction wave (negative wave) that connects zones u and 2. When simplified, based on Árnason [2005], Stoker’s analytical solution reads

$$[3.1] \quad 2 \frac{c_u}{c_d} = \frac{v_2}{c_d} + 2 \frac{c_2}{c_d}$$

with

$$[3.2] \quad \frac{v_2}{c_d} = \frac{U}{c_d} - \frac{c_d}{4U} \left(1 + \sqrt{1 + 8 \left(\frac{U}{c_d} \right)^2} \right)$$

and

$$[3.3] \quad \frac{c_2}{c_d} = \sqrt{\frac{1}{2} \left(\sqrt{1 + 8 \left(\frac{U}{c_d} \right)^2} - 1 \right)}$$

where $c = \sqrt{gh}$ and v is the fluid velocity uniform with depth. Solving Eq. 3.1 iteratively yields the constant velocity U of the bore-front as well as the constant flow depth $h_2 = c_2^2/g$ of the latter.

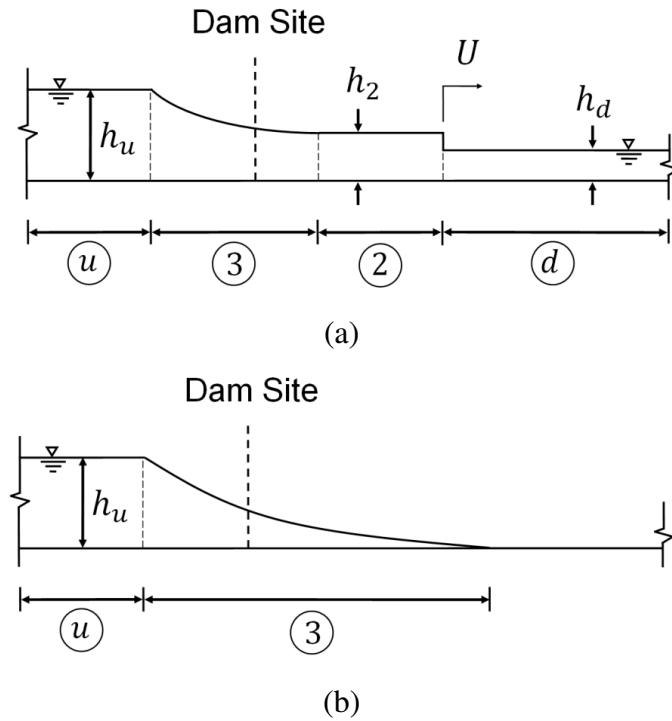


Figure 3.3. Definition sketch for the propagation of dam-break waves according to the shallow water equations: (a) wet-bed condition; (b) dry-bed condition.

The location of the bore-front can then be calculated as Ut , and the water surface elevation of zone 3 ($h_3 = c_3^2/g$), along with its location-dependant velocity can be evaluated using the following equations, respectively

$$[3.4] \quad c_3^2 = \frac{1}{9} \left(2c_u - \frac{x}{t} \right)^2$$

$$[3.5] \quad v_3 = \frac{2}{3} \left(c_u - \frac{x}{t} \right)$$

where x is the distance from the dam location and positive in the downstream direction. Lastly, in the case of the dry-bed condition (Figure 3.3-b), zones d and 2 are simply non-existent and all that is required is to determine zone 3 using Eq. 3.4 and Eq. 3.5.

3.2. Influence of Bed Condition on Dam-Break Flow Characteristics

Several researchers have performed physical experiments in order to investigate the influence of h_d on the propagation of a bore generated by a dam-break. Leal *et al.* [2006] performed a comprehensive compilation of experimental results which demonstrated that the bore-front velocity U drastically decreases initially as h_d increases from the dry-bed condition ($h_d = 0$) to a wet-bed condition ($h_d > 0$). Also in agreement with Stoker's analytical solution of the SWE [Stoker, 1957] (and as shown in Figure 3.4), U remains relatively constant as h_d increases further. In addition, Stansby *et al.* [1998] performed small-scale experiments which demonstrated that bores propagating over a wet bed tend to be deeper and have a steeper front compared to bores propagating on a “nearly” dry bed. Their experiments were conducted in a horizontal, 15.24m long $\times$ 0.40m wide $\times$ 0.40m high rectangular flume with Perspex acrylic sidewalls. The location of the vertical gate used for impounding water was located 9.76m downstream of the upstream end of the flume. Two scales were considered: upstream depths (h_u) of 0.100 and 0.360m along with downstream depths (h_d) of $0.1h_u$. Due to leakage across their gate apparatus, a thin film of water with a depth of 1-3mm covered the downstream section of the flume.

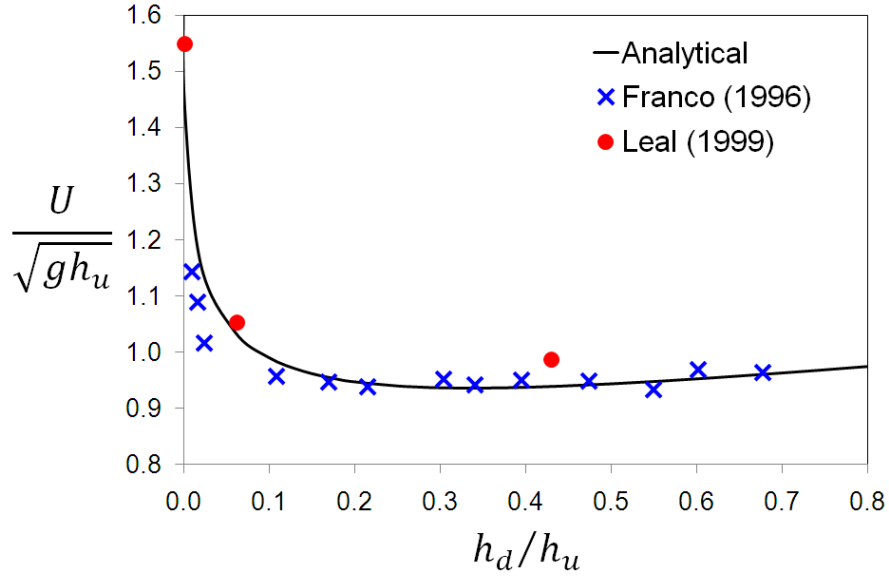


Figure 3.4. Influence of initial downstream water depth h_d on the bore-front velocity U (based on Leal *et al.* [2006]).

In view of this leakage, their “nearly” dry bed actually became a wet-bed condition with, $h_d \approx 0.002\text{m}$ on average. The previously-mentioned influence of the bed condition on the depth and steepness of the bore-front can be seen in Figure 3.5 and Figure 3.6. These figures show the propagation of bores propagating on a nearly dry bed and a wet bed, respectively.

On the other hand, Lauber and Hager [1998] studied the mechanics of dam-break waves propagating on a “perfectly” dry bed ($h_d = 0$) inside a smooth rectangular horizontal channel 14.00m long $\times$ 0.50m wide $\times$ 0.95m deep ($h_u = 0.300\text{m}$). By comparing the profile and propagation of the bore-fronts in Figure 3.7 and Figure 3.8, the differences induced by a downstream water layer of even a few millimetres can be noticed. The still images in Figure 3.7 correspond to the perfectly dry-bed experiment of Lauber and Hager [1998] ($h_u = 0.300\text{m}$ and $h_d = 0$), while the ones in Figure 3.8 correspond to one of the nearly dry-bed experiments of Stansby *et al.* [1998] ($h_u = 0.360\text{m}$ and $h_d \approx 0.002\text{m}$). It can be seen in Figure 3.7-a that when the bed is perfectly dry, the tip of the bore is reasonably similar to the one predicted by the analytical solution of the shallow water equations (Figure 3.3-b). However, Figure 3.8-a shows that, in the case of the nearly dry bed, the tip of the bore initially breaks and as the bore further

propagate downstream (Figure 3.8-e and Figure 3.8-f), the more stabilized tip seems steeper and deeper compared to the one seen in Figure 3.7-f.

Lastly, according to János *et al.* [2004], these drastic differences in the bore-front can be attributed to the resistance the downstream still water imposes to the propagation of the bore.

3.3. Physical Modeling of Bore-Structure Interaction

Following the approach of generating solitary waves within a wave flume mentioned in Section 3.1, Arikawa [2011] investigated at a relatively large scale the impulsive loading of breaking tsunami waves on reinforced concrete walls as well as debris impact loads. His experiments were performed in a 184m long $\times$ 3.5m wide $\times$ 12m deep wave flume equipped with a wave generator capable of strokes up to 14m that could generate waves up to 2.5m in height. His experimental setup was arranged such that the generated solitary waves would break directly in front of the test structures, and so the resulting pressures and forces would be the highest. It should be noted that such wave impacts would generally occur during inland intrusions corresponding to the *High Runup* type described previously in Section 3.1 (Figure 3.1-b). Figure 3.9 shows the typical time-history of the pressure acting on a structure due to the impact of a breaking tsunami wave. According to Arikawa, the peak in the time-history of the pressure occurring at initial breaking-wave impact with the structure is referred as “maximum impulsive bore pressure”. The magnitude of the latter depends on the profile of the impacting wave along with the elevation on the structure at which this pressure is measured. Furthermore, as the period of a tsunami wave is usually longer than several minutes, the duration of the inundation flow around a structure can be significant. The corresponding portion of the exerted pressure beyond initial impact is relatively constant and is referred as “sustainable pressure”. It should be noted that, as the test structures considered by Arikawa expanded almost entirely across the flume’s width, this sustainable pressure could perhaps result from the reflection of the wave instead. Finally, other researchers that have also studied tsunami-induced forces on structures by generating solitary waves within a laboratory flume include Thusyanthan and Madabhushi [2008] and Lukkunaprasit *et al.* [2009].

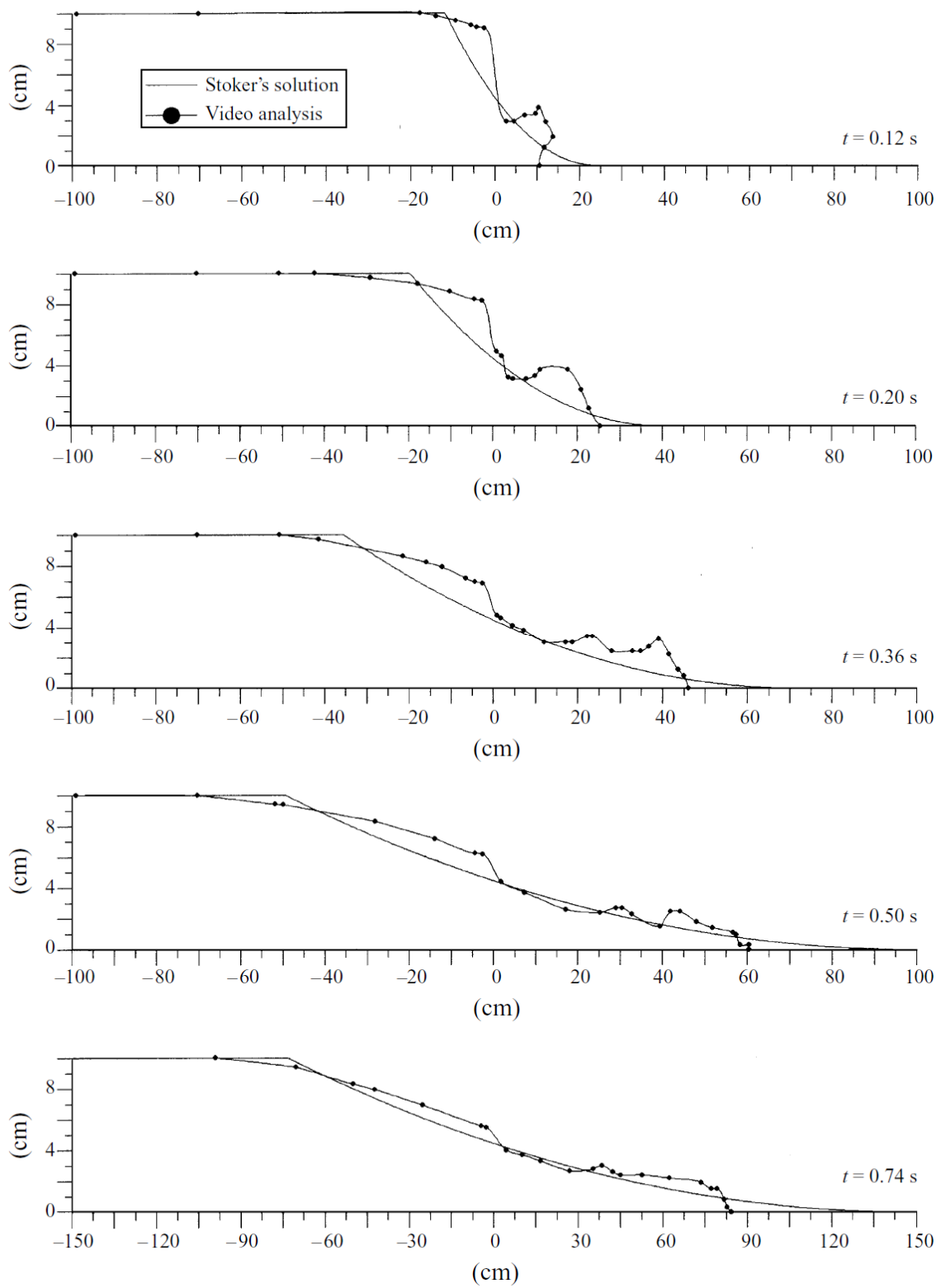


Figure 3.5. Comparison of experimental and analytical [Stoker, 1957] bore profiles for a nearly dry-bed condition (from Stansby *et al.* [1998] with $h_u = 0.100$ m and $h_d \approx 0.002$ m).

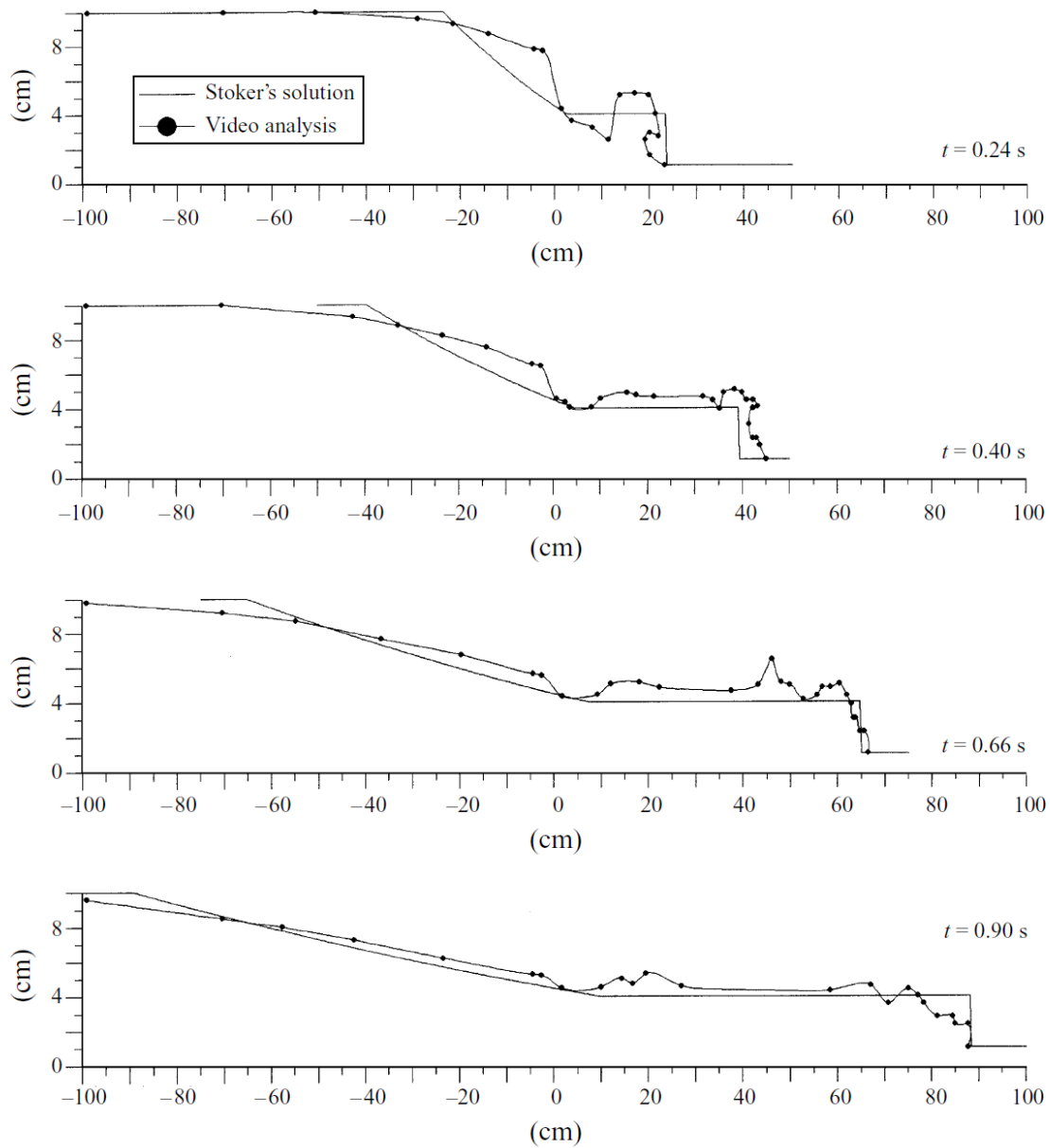


Figure 3.6. Comparison of experimental and analytical [Stoker, 1957] bore profiles for a wet-bed condition (from Stansby *et al.* [1998] with $h_u = 0.100\text{m}$ and $h_d = 0.010\text{m}$).

As part of several physical experiments, Ramsden [1993] investigated the impact with vertical walls of bores propagating over both dry and wet beds. His experiments were carried out in a 36.60m long $\times$ 0.40m wide $\times$ 0.61m deep horizontal channel in which bores were generated by releasing water from a reservoir with $h_u \approx 0.50\text{m}$ (dam-break approach mentioned in Section 3.1). In the case of the bore propagating over a wet

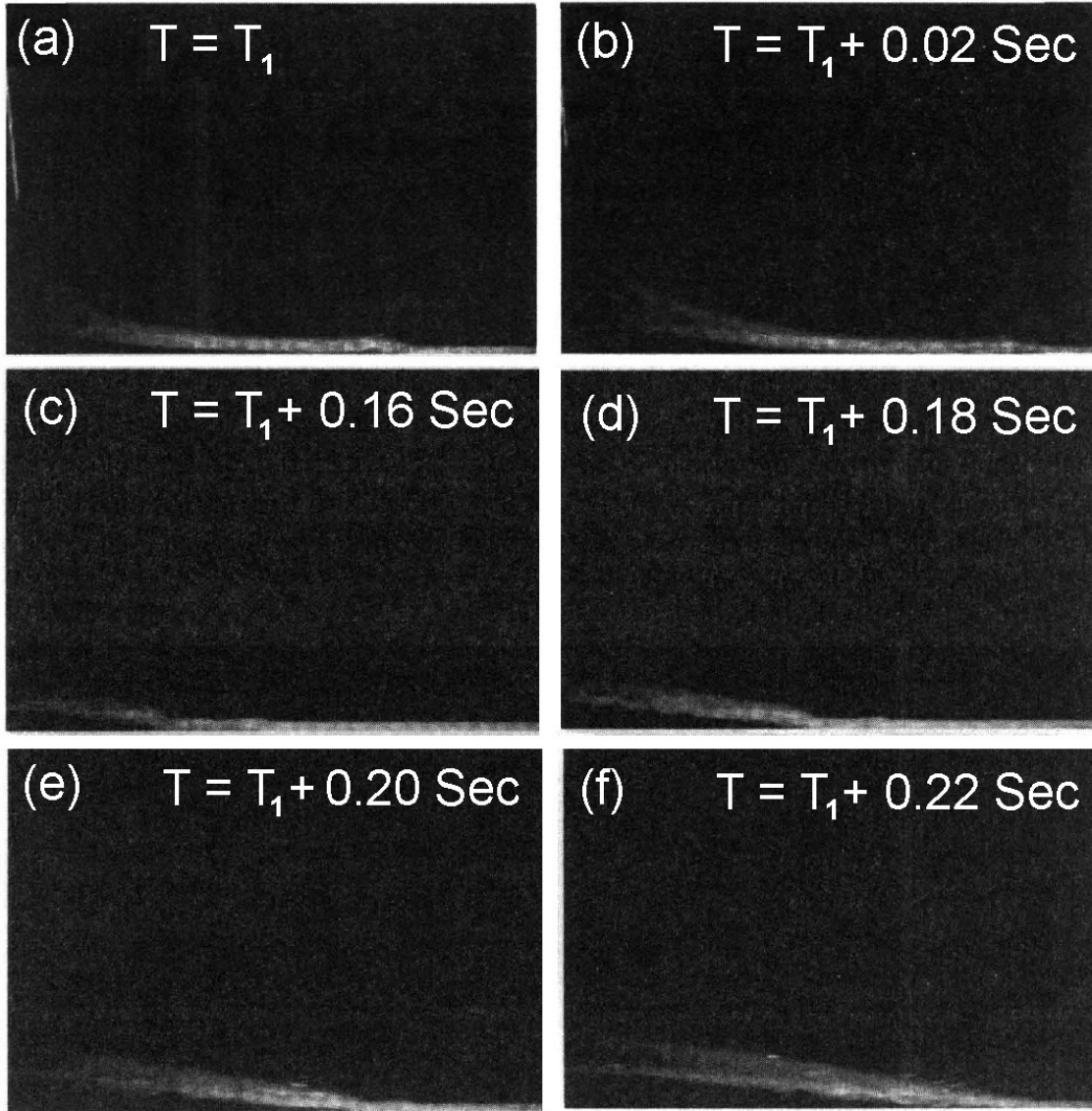


Figure 3.7. Bore propagating on a perfectly dry bed ($h_u = 0.300\text{m}$ and $h_d = 0$): (a) and (b) 0 to 0.30m downstream of the gate; (c) to (f) 0.90 to 1.20m downstream of the gate (adapted from Lauber and Hager [1998], time T_1 is not specified).

bed, an initial $h_d \approx 0.003\text{m}$ was considered. From his results, Ramsden observed that the initial impact pressure acting on a vertical wall is higher in the case of a bore propagating over a wet bed compared to a bore propagating over a dry bed (Figure 3.10-c). This was attributed to the steeper front of the bore advancing on the wet bed (Figure 3.10-a) at the sampling elevation of 0.018m above the channel's bottom (the lowest elevation at which pressures were measured).

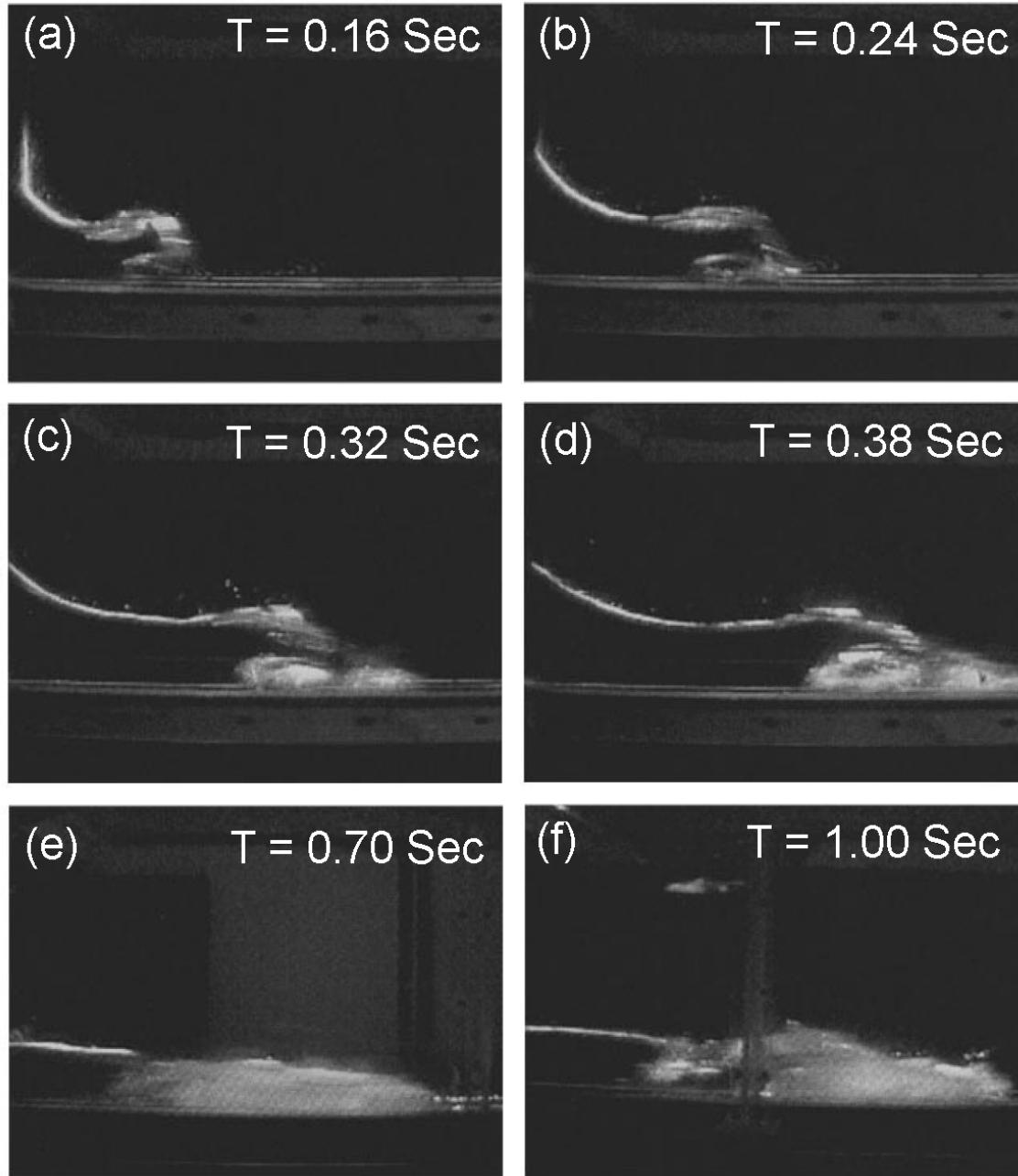


Figure 3.8. Bore propagating on a nearly dry bed ($h_u = 0.360\text{m}$ and $h_d \approx 0.002\text{m}$): (a) to (d) just downstream of the gate; (e) and (f) further downstream of the gate (adapted from Stansby *et al.* [1998], the actual distances where the images were taken is not specified).

On the other hand, although the impacted structure was a square column with a side dimension of 0.30m , the results of the physical experiments performed within the current research program introduced in Chapter 1 [Nistor *et al.*, 2011] exhibit a different

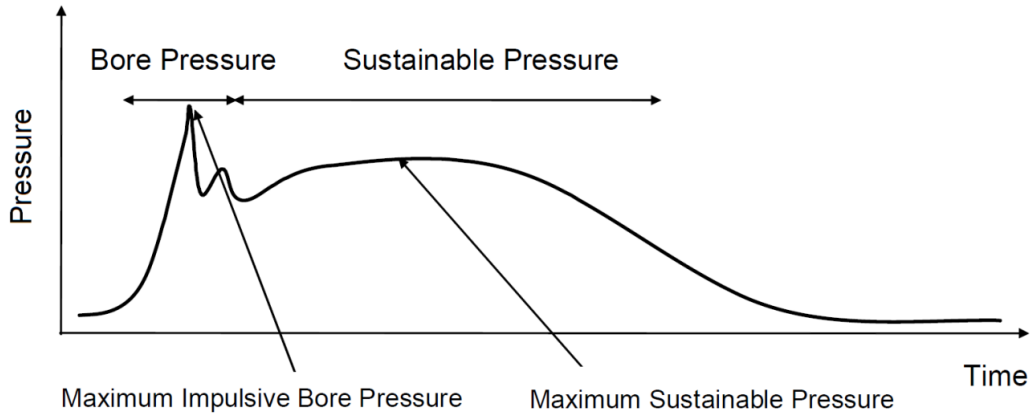


Figure 3.9. Typical time-history of the pressure acting on structures due to the impact of a breaking tsunami wave (adapted from Arikawa [2011]).

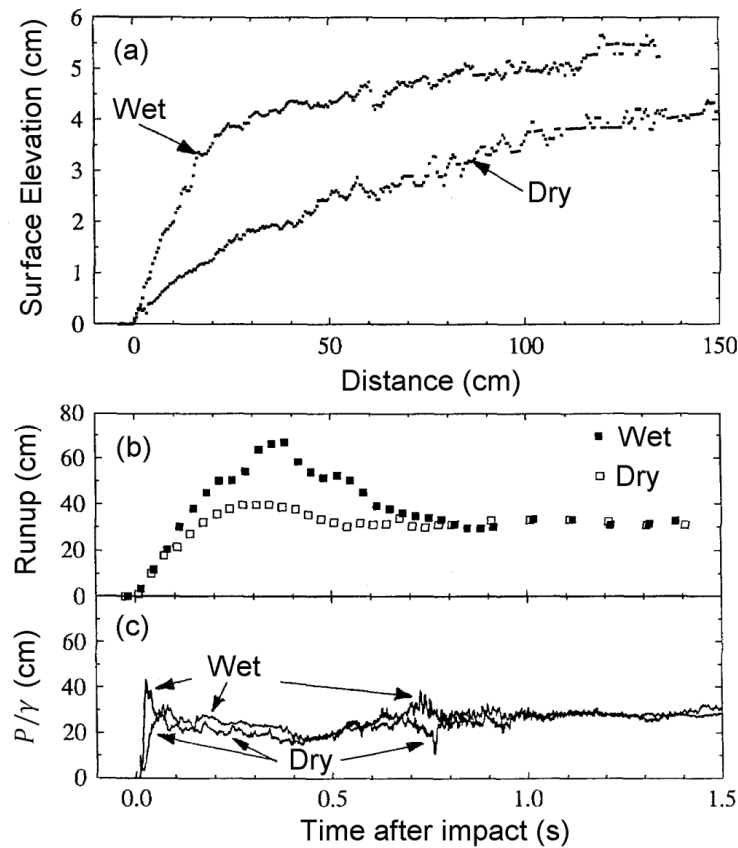


Figure 3.10. Comparison of the experimental (a) bore profile, (b) runup, and (c) pressure head at an elevation of 0.018m above the channel's bottom on a vertical wall due to bores ($h_u = 0.5\text{m}$) propagating on dry ($h_d = 0$) and wet ($h_d = 0.003\text{m}$) beds. γ is the unit weight of water (adapted from Ramsden [1993]).

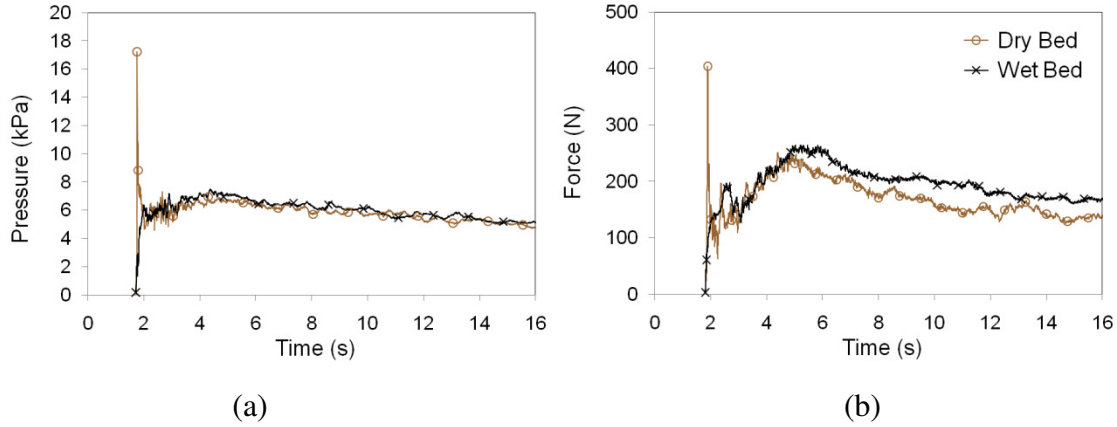


Figure 3.11. Comparison of the experimental (a) pressure at an elevation of 0.02m above the channel's bottom, and (b) net force acting streamwise on a square column due to bores ($h_u = 0.55\text{m}$) propagating on dry ($h_d = 0$) and wet ($h_d = 0.005\text{m}$) beds (experimental results of Al-Faesly *et al.* [2012]).

behaviour. Essentially, the initial impact pressure is considerably higher when the bore is propagating over a dry bed compared to when it is propagating over a wet bed. This was observed for the $h_u = 0.55, 0.85$, and 1.15m that were considered along with an $h_d \approx 0.005\text{m}$, which resulted from the difficulty in completely drying the channel between each run. A complete description of the experimental setup is provided in Section 4.1.2 of Chapter 4. Figure 3.11 shows the time-history of the pressure acting at an elevation of 0.020m above the channel's bottom and the net force acting streamwise on the square column for $h_u = 0.55\text{m}$, for both dry- and wet-bed conditions.

Finally, Árnason [2005] also studied the interaction of bores with free-standing structures of various cross-sections. His experiments were conducted in a 16.60m long $\times$ 0.60m wide $\times$ 0.45m deep channel in which bores propagating over a wet bed ($h_d = 0.02\text{m}$) were generated using reservoir depths of up to $h_u = 0.30\text{m}$. For a square column with a side dimension of 0.12m, Figure 3.12 shows that the initial impact force does not significantly overshoot the later quasi-steady hydrodynamic force. As indicated in Figure 3.11-b, this behaviour was also observed in the current research program for the case of a wet-bed condition.

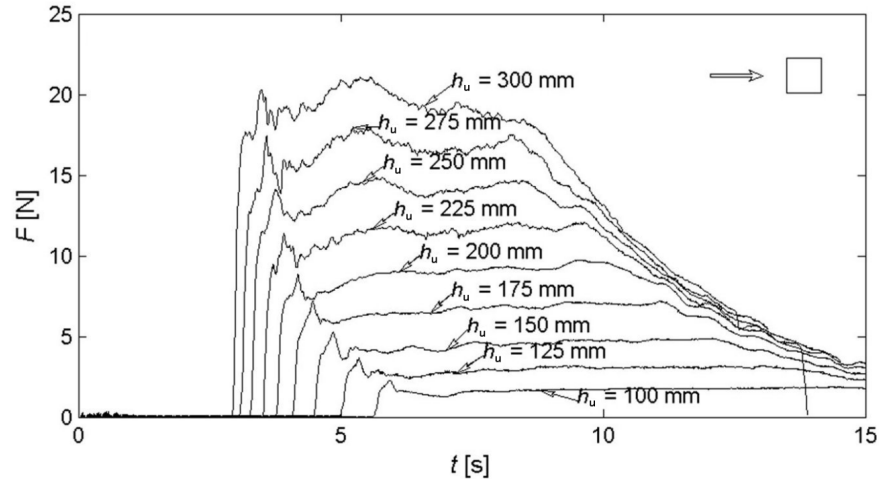


Figure 3.12. Experimental time-history of the net force acting in the streamwise direction on a square column due to bores generated by various upstream water depths h_u (adapted from Árnason [2005]).

3.4. Numerical Modeling of Bore-Structure Interaction using SPH

In regards to bore-structure interactions, the SPH method was first used by Gómez-Gesteira and Dalrymple [2004] to simulate the impact of bores with a tall free-standing structure. Using a weakly compressible SPH model (WCSPH), they reproduced the physical experiment performed by Yeh and Petroff [n.d.] where a dam-break wave impacted a square column with a side dimension of 0.12m inside a 1.60m long $\times$ 0.61m wide $\times$ 0.75m deep rectangular tank ($h_u = 0.30\text{m}$ and $h_d = 0.01\text{m}$). For their simulations, they used the dynamic boundary condition as described in Section 2.3.6, and it is not mentioned whether any correction scheme to prevent large fluctuations in the pressure field was used (see Section 2.3.5). Nevertheless, they obtained promising agreement between the experimental and the numerical time-history of the resulting force acting on the column (Figure 3.13). In addition, as the experiment of Yeh and Petroff was performed with a wet bed ($h_d = 0.01\text{m}$), they also performed a simulation with a dry bed and the resulting bore profiles are shown in Figure 3.14. It can be noticed that the bore propagating on the dry bed is faster as the presence of the water layer seems to be resisting the propagation of the bore over the wet bed (Figure 3.14-b). Lastly, it was also

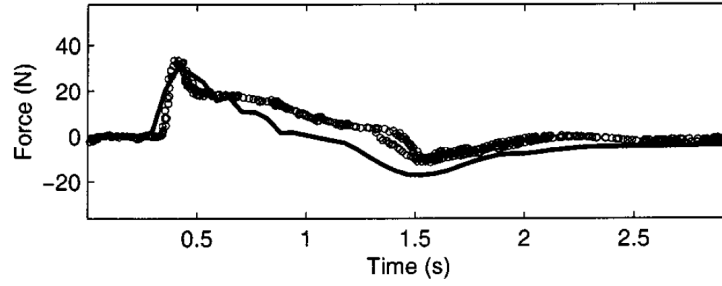


Figure 3.13. Comparison between experimental (circles) and numerical (solid line) time-history of the resulting force on a column (from Gómez-Gesteira and Dalrymple [2004]).

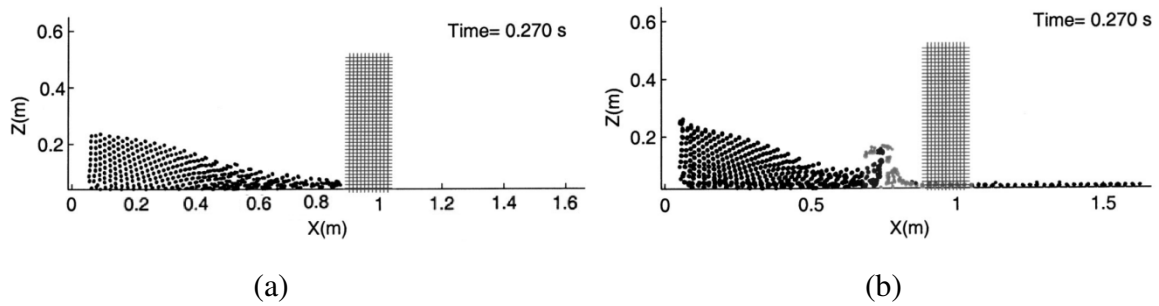


Figure 3.14. Comparison of bore profile for the (a) dry and (b) wet bed conditions moments before initial impact (from Gómez-Gesteira and Dalrymple [2004]).

noticed that the differences in the shape of the bore-front caused by the bed condition ultimately affected the distribution of the resulting force on the column (Figure 3.15).

The same set of experimental data was later reproduced by Silvester and Cleary [2006] in a study where a sensitivity analysis of various SPH simulation parameters was performed. Using a WCSPH model as well, they considered both a repulsive boundary condition and a dynamic boundary condition, and it is also not specified whether any correction scheme to prevent large fluctuations in the pressure field was used. It should be noted that the formulation of the repulsive boundary condition they considered is different than the one implemented in SPHysics as described in Section 2.3.6. Figure 3.16 displays, in sequence, the experiment of Yeh and Petroff simulated by Silvester and Cleary. As it was also the case in the physical experiments of Stansby *et al.* [1998] (Figure 3.8-a), the flow resistance by the downstream water layer also induces the tip of

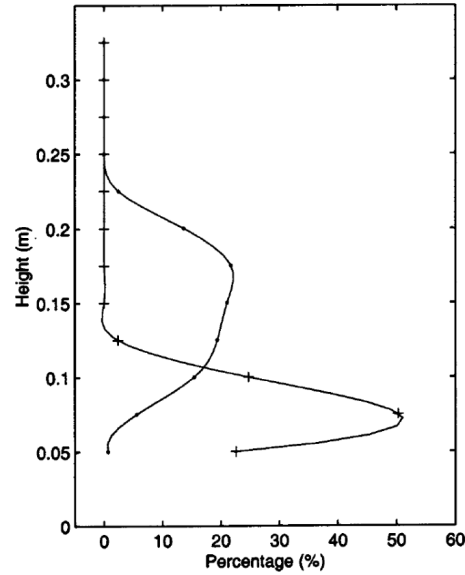


Figure 3.15. Vertical distribution of the maximum force acting on the upstream face of a column for dry (crosses) and wet (dots) bed conditions (from Gómez-Gesteira and Dalrymple [2004]).

the dam-break wave to break (Figure 3.16-b). The observations made by Silvester and Cleary are the following:

- As seen in Figure 3.17, the choice of the boundary condition among the two scenarios they considered did not significantly affect the resulting force on the column.
- In the case of a vertical gate, a slower opening of the latter will induce a higher impact force, as the mechanism by which the gravitational potential energy is converted to kinetic energy will be slightly different.
- Using their own formulation for an artificial viscosity [Cleary, 1998] (see Section 2.3.3), they found their numerical results to be insensitive to the viscosity of the fluid.
- An increase in the particle resolution will be accompanied with an increase in the accuracy of the numerical results.

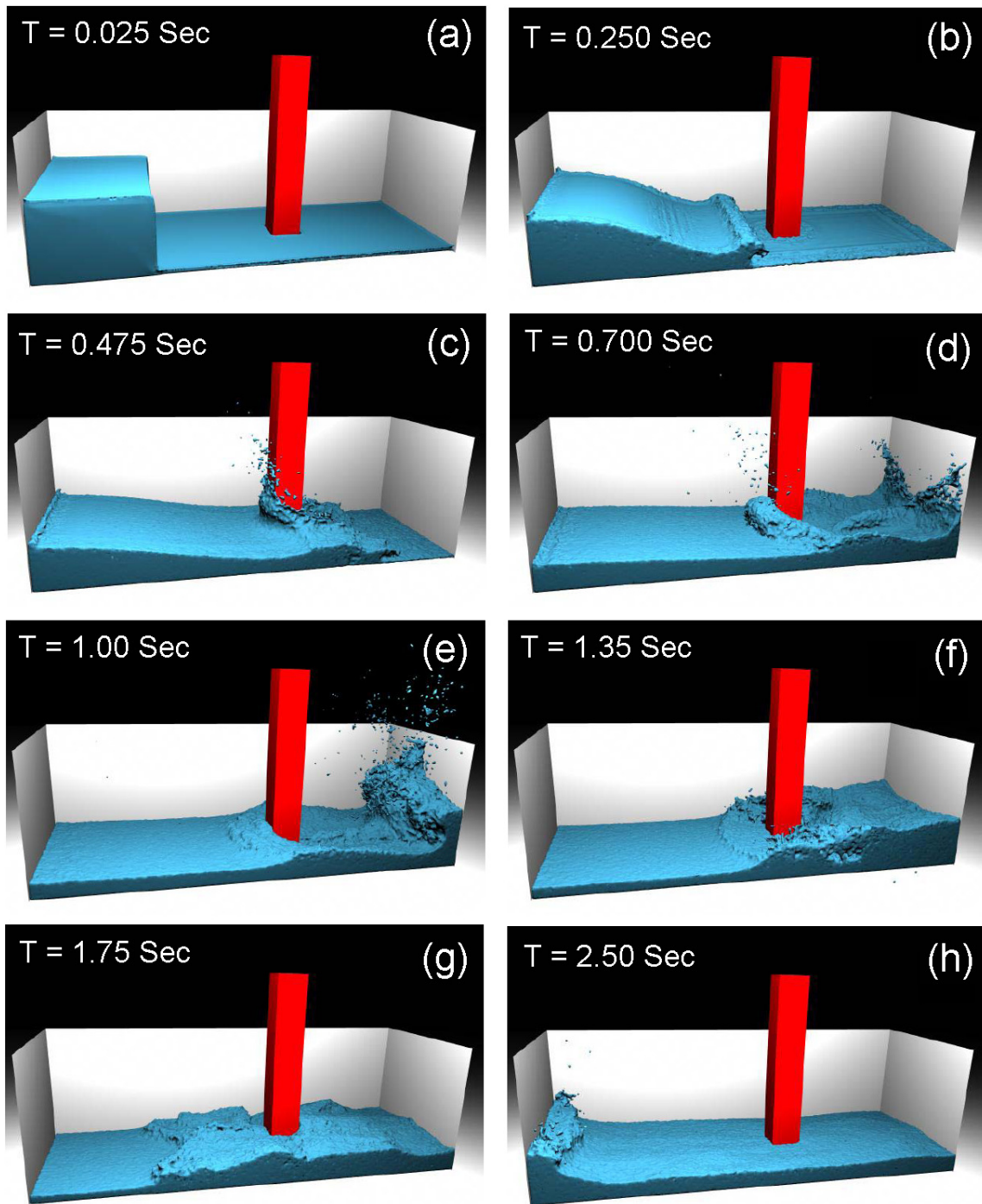


Figure 3.16. Numerical flow sequence of the experiment of Yeh and Petroff [n.d.] with $h_u = 0.30\text{m}$ and $h_d = 0.01\text{m}$ (adapted from Silvester and Cleary [2006]).

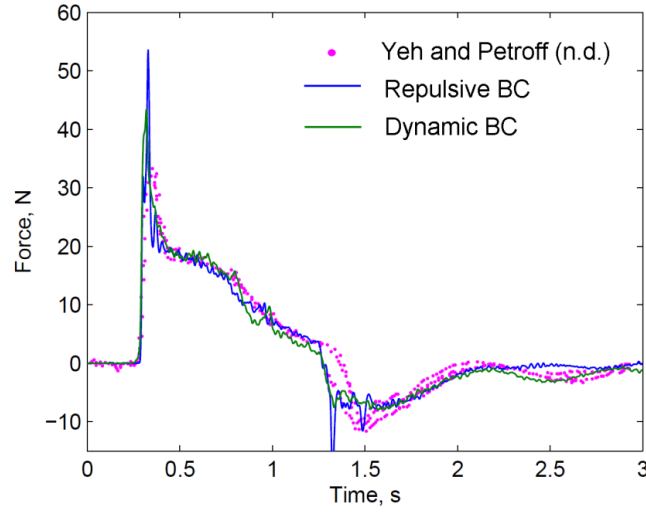


Figure 3.17. Comparison between experimental and numerical time-history of the resulting force on a column using both a repulsive and a dynamic boundary condition (from Silvester and Cleary [2006]).

By simulating the physical experiment of Kleefsman *et al.* [2005], Lee *et al.* [2010] compared a weakly compressible formulation with a truly incompressible formulation of the SPH method. Kleefsman *et al.* first performed this experiment for the validation of their grid-based FVM model that utilized the *Volume of Fluid* (VOF) technique to track the free-surface. Once again, this experiment was performed to investigate the impact of a dam-break wave with a 0.16m long $\times$ 0.40m wide $\times$ 0.16m high rectangular obstacle within a 3.22m long $\times$ 1.00m wide $\times$ 1.00m deep rectangular tank ($h_u = 0.55$ m). The results of Lee *et al.* showed that a WCSPH model would yield an inaccurate and highly-fluctuating pressure field compared to a truly incompressible SPH model (ISPH), see Figure 3.18. Although they noted the existence of density filters for improving the prediction of the pressure field (see Section 2.3.5), they did not explicitly mention whether or not they used one in their WCSPH simulation. In addition, they also observed that the splash at initial impact of the dam-break wave with the obstacle was much more realistic in their ISPH simulation as the intensity of this splash was significantly underestimated in their WCSPH simulation (Figure 3.19). Interestingly, by using a Shepard density filter to reduce fluctuations in the pressure field along with a

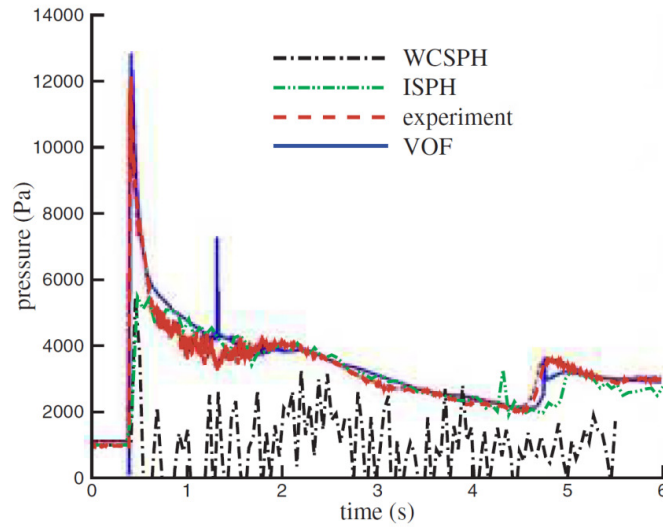


Figure 3.18. Experimental and numerical time-history of the pressure exerted on the upstream face of a small obstacle at an elevation of 0.021m above the experimental tank's bottom. The experimental and VOF results were obtained by Kleefsman *et al.* [2005] (from Lee *et al.* [2010]).

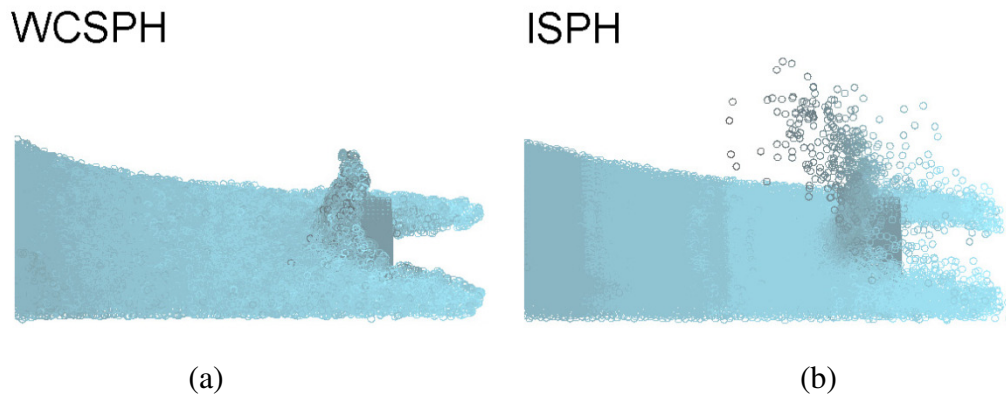


Figure 3.19. Comparison of the water splash 0.56 seconds after gate opening predicted by (a) WCSPH and (b) ISPH models (from Lee *et al.* [2010]).

relatively high particle resolution, Crespo *et al.* [2011] reproduced the same physical experiment using a WCSPH model. The latter, which executed on the Graphics Processing Unit (GPU), obtained both accurate pressure results and realistic splashing at initial impact (Figure 3.20 and Figure 3.21, respectively).

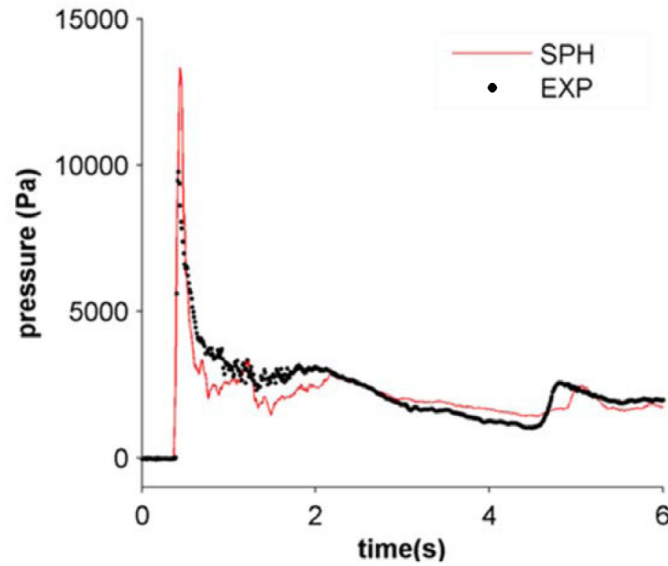


Figure 3.20. Experimental and numerical time-history of the pressure exerted on upstream face of a small obstacle at an elevation of 0.021m above the experimental tank's bottom. The experimental results were obtained by Kleefsman *et al.* [2005] (from Crespo *et al.* [2011]).

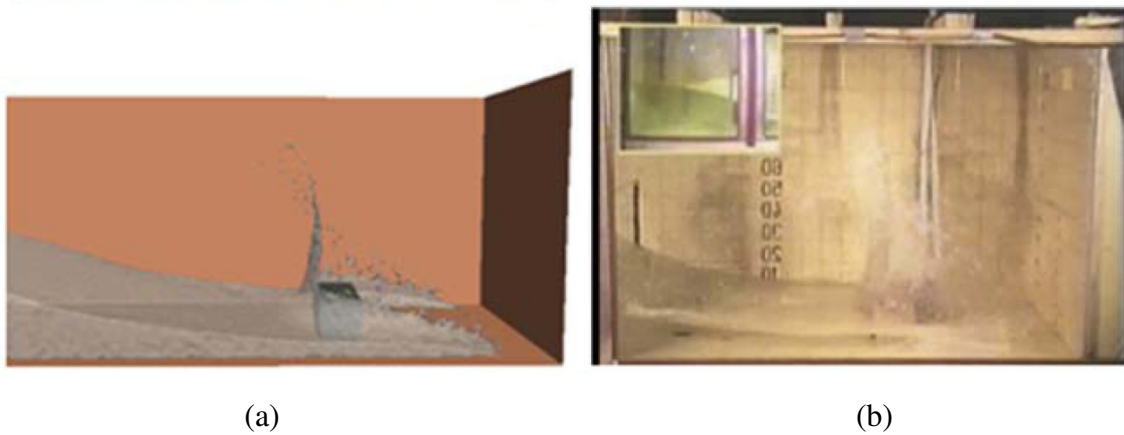


Figure 3.21. Comparison of the (a) numerical and (b) experimental water splash 0.56 seconds after the opening of the gate. The experimental image was taken by Kleefsman *et al.* [2005] (from Crespo *et al.* [2011]).

Other studies [e.g. Hughes and Graham, 2010; Shadloo *et al.*, 2011] have also compared WCSPH and ISPH models and these have shown that both approaches can produce similar results.

Finally, due to the relatively small size of the bores and the presence of a wall just downstream of the impacted structures that caused wave reflection (e.g. Figure 3.16), the experiments of Yeh and Petroff [n.d.] and Kleefsman *et al.* [2005] did not realistically reproduce the impact of tsunami bores on structures. Hence, realistic time-histories of hydrodynamic forces resulting from the extended action of tsunami bores were not obtained.

3.5. Conclusion

The influence of the bed condition (i.e. dry or wet) on the propagation of hydraulic bores is relatively well known in the literature. A downstream water depth (h_d) of even a few millimetres can reduce the velocity of the bore-fronts but more importantly increase their depth and steepness. These effects are attributed to the resistance that the downstream still water imposes on the propagation of bores [Jánosi *et al.*, 2004]. However, little is known about how these effects ultimately influence the hydrodynamic forces resulting from the interaction of hydraulic bores with structures. To date, studies considering not only both bed conditions but also two fundamentally different structure types, such as slender columns and infinitely wide walls, have not been performed either experimentally or numerically. Lastly, although numerous studies applied the SPH method to simulate bore-structure interactions, these only considered small-scale scenarios that could not provide realistic time-histories of hydrodynamic forces comparable to the ones of an actual tsunami-induced bore. This could perhaps be explained by the fact that before recent advances related to computing technology (e.g. GPU computing and parallel computing in general), it was difficult to model large computational domains accurately using the SPH method, due to the related high computational cost.

Chapter 4. Description of Physical and Numerical Experiments

This chapter describes in detail the several experimental setups considered for this study along with the computational domains that were conceived to reproduce them numerically. Although the main objective of this research is to reproduce the physical experiments performed within the related comprehensive study presented in Chapter 1, the smaller experiments of Yeh and Petroff [n.d.] and Kleefsman *et al.* [2005] are also reproduced. This was done to compare the results with previous SPH studies presented in Section 3.4 and also to test various numerical aspects before executing computationally expensive numerical runs. Representative results of a related sensitivity analysis are also included. The purpose of the sensitivity analysis was to guide in the calibration of the various SPH-related parameters covered in Chapter 2.

4.1. Physical Models

4.1.1 Yeh and Petroff [n.d.]

Performed at the University of Washington, Seattle, this experimental test was first referred to as a “bore in a box”. A dam-break hydraulic bore was generated in a 1.60m long $\times$ 0.6m wide $\times$ 0.75m deep rectangular tank, as shown in Figure 4.1. The volume of water initially contained behind a vertical gate was 0.40m long $\times$ 0.6m wide $\times$ 0.30m deep ($h_u = 0.30$ m). A square column with a side dimension of 0.12m and a height of 0.75m was located 0.56m downstream of the gate and at the centerline of the tank. The time-history of the net force acting on the column was measured using a load cell. No drainage of the tank was considered during the experiments, such that the hydraulic bore reflected against the downstream wall after the initial impact with the column. Due to

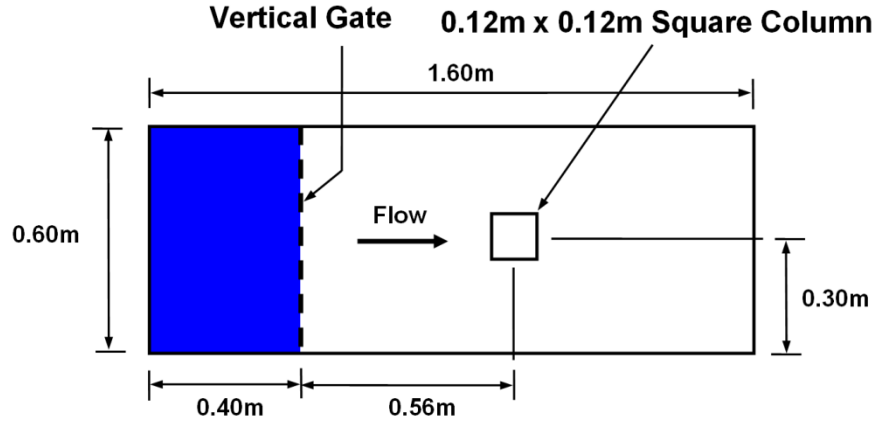


Figure 4.1. Plan view of the experimental setup of Yeh and Petroff [n.d.]

some difficulty in completely draining the tank prior to the beginning of the run, a water layer (h_d) of approximately 0.01m was present downstream of the gate.

4.1.2 Nouri *et al.* [2010] and Al-Faesly *et al.* [2012]

These physical experiments were performed within the aforementioned research program initiated at the University of Ottawa in collaboration with the Canadian Hydraulic Center of the National Research Council (NRC-CHC), in Ottawa, Canada. The experimental setup was first constructed for Nouri *et al.* [2010] and this facility was later reused by Al-Faesly *et al.* [2012] to extend the work.

Facility

The experiments were performed inside a 12.87m long $\times$ 2.7m wide $\times$ 1.4m deep High Discharge Flume (HDF) at NRC-CHC. This stainless steel flume was partitioned such that dam-break bores were generated in a 10.85m long $\times$ 1.3m wide $\times$ 1.4m deep channel section using water impounded and released by means of a swinging gate mechanism. The gate impounded a 5.6m long $\times$ 2.7m wide $\times$ 1.4m deep reservoir located immediately upstream of the former. A schematic of this arrangement is shown in Figure 4.2 and a photo of the arrangement is shown in Figure 4.3. As the possible length for a straight reservoir was physically limited, the flume was subdivided in this way in order to maximize the duration of the hydraulic bores (the effects of the resulting bend in the reservoir geometry will be assessed numerically in Chapter 7). Once the downstream

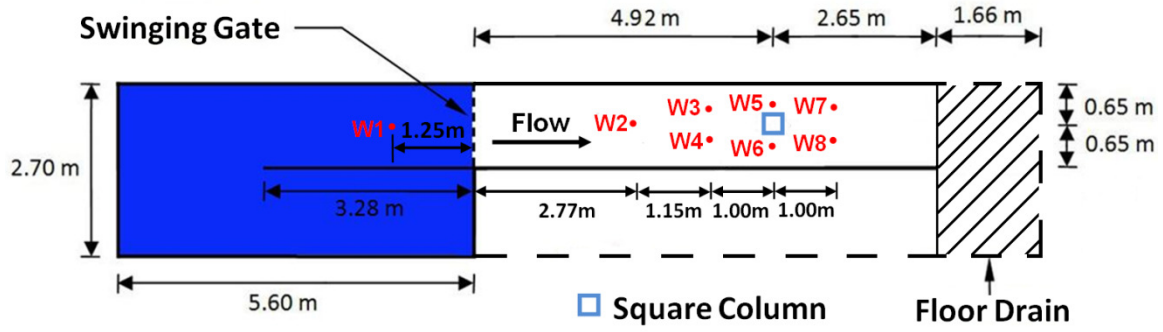


Figure 4.2. Plan view of the experimental setup of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] shown with a square column.



Figure 4.3. Picture of the experimental setup of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] with a full reservoir and looking downstream (courtesy of Taofiq Al-Faesly).

end of the flume was reached, water was evacuated through a vertical floor drain such that the flow upstream would not be affected. An electric winch and counter weight mechanism was used to achieve a quick opening of the gate (0.25 to 0.30s according to digital video recordings). According to Lauber and Hager [1998], ideal dam-break occurs if the dimensionless gate removal time $T_0 = t_0(g/h_u)^{1/2}$ is smaller than $2^{1/2}$, where t_0 is the removal time of the gate. For the selected impoundment depths (h_u) of 0.50, 0.75, 0.85, and 1.00m [Nouri *et al.*, 2010] and 0.55, 0.85, and 1.15m [Al-Faesly *et al.*, 2012], this criterion was satisfied. The experiments by Nouri *et al.* were only considered in this research for qualitative visual comparison with the numerical simulations.

Although rubber water-stops were used around the gate assembly, due to the large impoundment depths employed, minor leakages across the gate occurred. Due to the difficulty in completely drying the downstream section of the test channel between each run, most experiments were performed with a downstream water depth (h_d) of approximately 0.005m (see Figure 6.15-a). Runs performed with a “virtually” dry bed (see Figure 6.14-a) were only achieved by allowing sufficient time between runs for the flume’s bottom to dry. However, video recordings revealed that experimental runs were sometimes performed with a “partially” dry-bed condition (Figure 4.4). It should be noted that, for demonstrative purposes, Figure 6.14 and Figure 6.15 correspond to experimental runs performed with a mitigation wall and these were not simulated with the numerical model. As the influence of the bed condition seen in Figure 3.11 was only noticed in the later stages of the experimental program of Al-Faesly *et al.* [2012], the bed condition was only assessed for a very few of the experimental runs considered herein.

Finally, the details of the experimental runs available for quantitative comparison with numerical results are shown in Table 4.1.



Figure 4.4. Partially dry-bed condition observed in some of the experimental runs.

Table 4.1. Details of the experimental runs considered for quantitative comparison.

Experimental Run Label	h_u (m)	h_d (m)	Column	Corresponding Numerical Simulation	Device	Frequency (Hz)
E-55-DRY	0.55	~0	Square	D2-55	Dynamometer	1000
					Pressure transducers	200
E-55-WET	0.55	~0.005	Square	N/A	Dynamometer	1000
					Pressure transducers	200
					Wave gauges	500
E-85-N	0.85	N/A	None	D2-85-N	Wave gauges	250
E-85	0.85	N/A	Square	D2-85	Dynamometer	1000
					Pressure transducers	200
					Wave gauges	250
E-85-C	0.85	N/A	Circular	D2-85-C	Dynamometer	1000
					Pressure transducers	500
					Wave gauges	250
E-115	1.15	N/A	Square	D2-115	Dynamometer	1000
					Pressure transducers	200
					Wave gauges	200

Methodology

With the gate initially closed and locked, water was supplied to the upstream reservoir using a variable-pitch blade pump with a discharge varying from 0.1 to 1.0m³/s. As the water level approached the targeted impoundment depth (h_u), the supply rate was then adjusted to ensure that the water level remained essentially constant before opening the gate. Upon release of the gate, the water flowed rapidly downstream in the form of a hydraulic bore to impact the test structure.

Column Model

The structures impacted by the hydraulic bores were: (a) a 0.30m diameter circular column made of a polyvinyl chloride (PVC) pipe 9.0mm thick and 0.70m high, and (b) a sharp-edge square column with a side dimension of 0.30m and a height of 1.00m, constructed out of four 6.3mm thick Plexiglas sheets. In order to transfer the forces imposed by the hydraulic bores to the columns' bases effectively, the two models were mounted on a rigid steel frame, positioned 4.92m downstream from the gate.

Instrumentation

For recording the net base horizontal force acting on the columns, the lower part of their stiff frames was fixed to a six degree of freedom dynamometer (AMTI MC6 Series), which itself was rigidly bolted to the bottom of the steel flume. Also, the vertical pressure distribution exerted on the columns was recorded using ten differential pressure transducers (Honeywell 20PC Series) aligned vertically on its upstream face at elevations of 0.05, 0.10, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, and 0.55m above the flume's bottom. Figure 4.5 displays a schematic of the column model in which the pressure transducers are denoted by "P". Capacitance wave gauges (Akamina model WP-AO-V02-03) were employed to measure water surface elevations along the flume as well as the inundation depths around the test structures. These installations are denoted by "W" in Figure 4.2 and Figure 4.5, respectively. Velocities of the bore-fronts were estimated using a high-speed video camera (Redlake M-Series) that was referenced to a 0.3m × 0.3m orthogonal grid painted on the flume's bed. The sampling rate of the measuring devices varied from 200 to 1,000Hz as presented in Table 4.1.

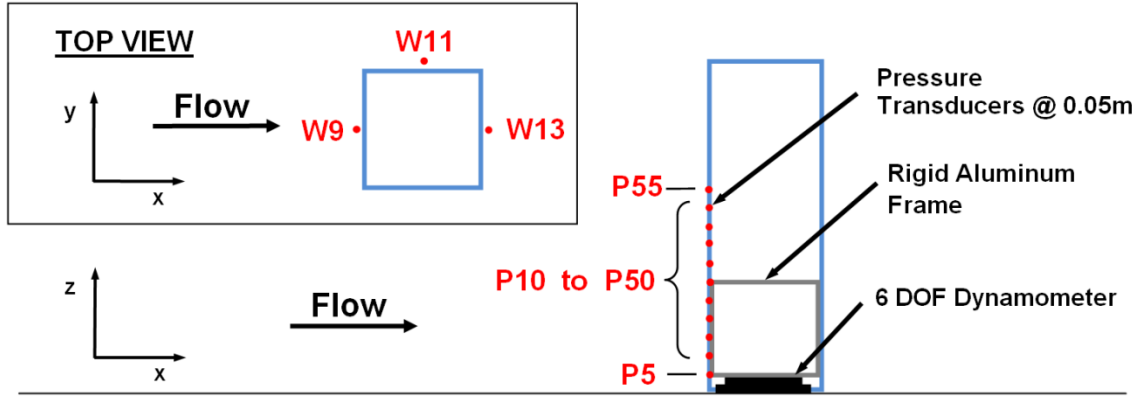


Figure 4.5. Schematic of experimental column model (square) and instrumentation used by Nouri *et al.* [2010] and Al-Faesly *et al.* [2012].

4.1.3 Kleefsman *et al.* [2005]

Performed at the Maritime Research Institute Netherlands (MARIN) and similar to the experiment of Yeh and Petroff [n.d.], this dam-break experiment was performed in a 3.22m long $\times$ 1.00m wide $\times$ 1.00m deep rectangular tank with $h_u = 0.55\text{m}$ and $h_d = 0$. However, rather than a slender column, the test structure was a short and relatively wide obstacle that measured 0.161m long $\times$ 0.403m wide $\times$ 0.161m high and which was located 1.248m downstream of a vertical gate (Figure 4.6). This gate was “instantaneously” raised by the release of weights. The time-history of the pressure exerted on both the upstream and top faces of the obstacle was recorded using eight pressure transducers (denoted by “P” in Figure 4.7). The time-history of the water surface elevation was recorded using four wave gages placed at various locations along the centerline of the experimental tank (denoted by “W” in Figure 4.6). The sampling rate of the measuring devices was 1000Hz.

4.2. Calibration of SPH-related Parameters

This section presents the most relevant results of the aforementioned related sensitivity analysis. In addition to assisting in calibrating the numerical model, a secondary objective of this exercise was to better understand the influence of various SPH-related parameters. The physical experiment considered herein was the one of Al-Faesly *et al.* [2012] with h_u

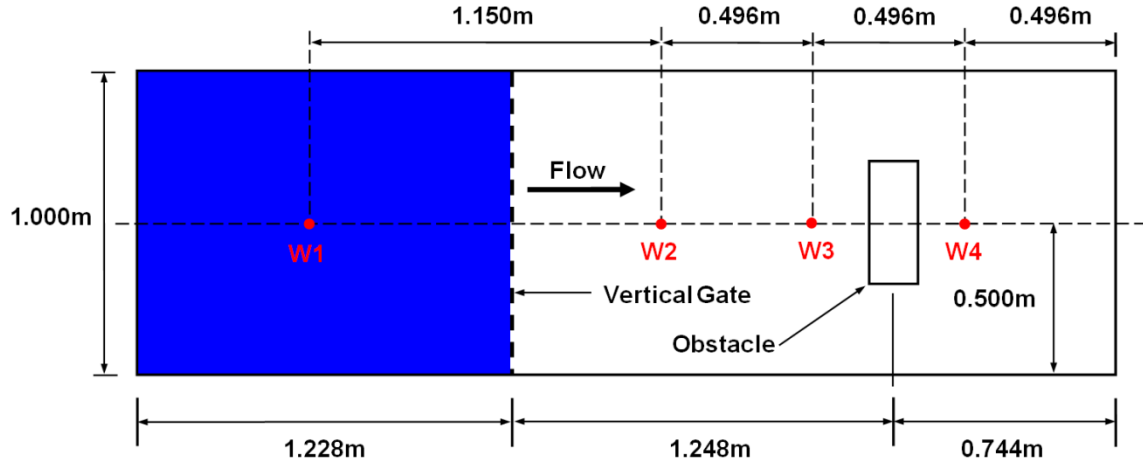


Figure 4.6. Plan view of the experimental setup of Kleefsman *et al.* [2005].

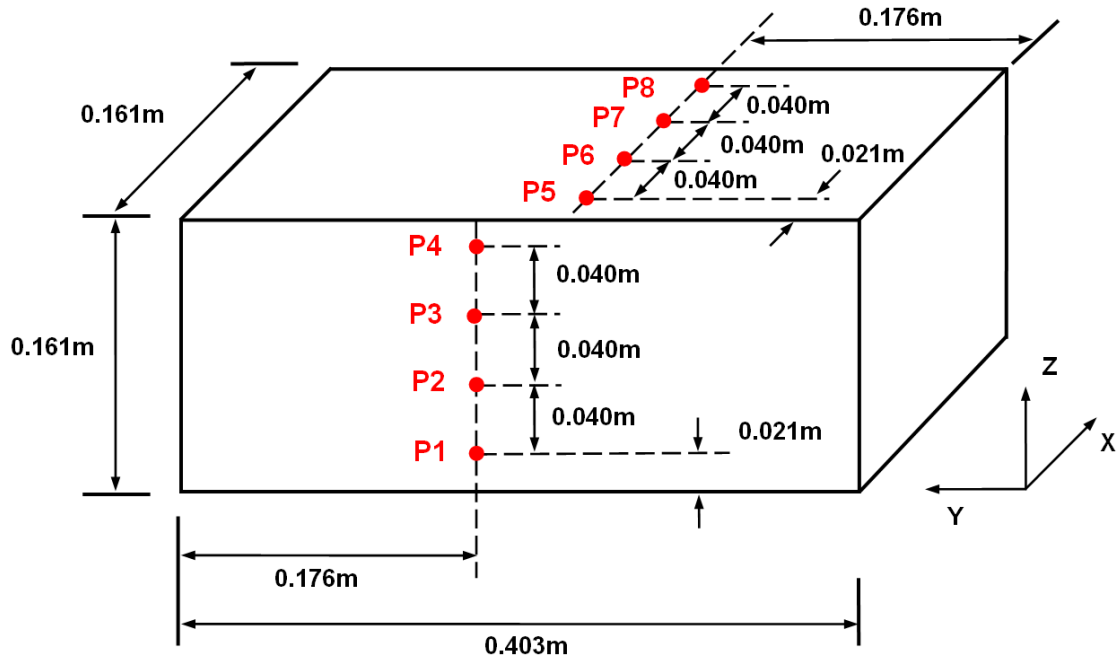


Figure 4.7. Test structure considered in the experiment of Kleefsman *et al.* [2005].

= 0.85m. This selection was based on the fact that the square column was, on one hand, easier to simulate, and also the fact that the middle value of $h_u = 0.85\text{m}$ was considered to be neither too small nor too large. It is important to point out that the physical experiments of Al-Faesly *et al.* [2012] were performed almost simultaneously with this sensitivity analysis. Because of some unexpected delays in the compilation of the physical results (and because the numerical runs had to be initiated in order to meet deadlines), the results of the sensitivity analysis were not compared with the physical experiments. Hence, the selection of the simulation parameters was based on judgement rather than by a more conventional calibration against physical results. The specially conceived procedure, with which numerical results are obtained from the raw numerical outputs, is described in Chapter 5.

At the time of model calibration, only the serial version of SPHysics was available. Hence, simulating the actual size of the physical experiment with reasonable resolution would have yielded excessive computational times (T_{comp}). According to Lauber and Hager [1998], the flow characteristics close to the front of a dam-break wave are not affected by the reservoir length (L) if $L/h_u > 2$. It was assumed that the initial impact force should be the same as in the physical experiment, even if the water reservoir was shorter. This will subsequently be verified numerically in Chapter 7. Thus, although the flow width ($W = 1.30\text{m}$) was kept the same as in the physical experiments, a smaller reservoir ($L = 2.5\text{m}$) was considered. The resulting geometry of the computational domain corresponds to the general one shown in Figure 4.14.

4.2.1 Preliminary selection of parameter and simulation options

Based on simulation examples [Gómez-Gesteira *et al.*, 2010a] and through a trial and error process, Table 4.2 contains the values of the parameters adopted for the base case of the sensitivity analysis. Varying one parameter at a time, the following subsections contain the numerical results of the net base horizontal force acting on the column for runs with a simulation time (T_{sim}) of 3 seconds. The influences of additional parameters for which the graphical results are shown in Appendix B are also discussed and the values selected for all the simulations that follow are listed.

Due to time limitations, some simulation aspects or options were selected based on the success of recent SPH studies and preliminary tests. Following the work of Silvester and Cleary [2006], Rogers *et al.*, [2010], and Omidvar *et al.*, [2012], in which the resulting force on objects was of primary importance, a repulsive boundary condition was considered [Rogers and Dalrymple, 2008]. Furthermore, based on the last two studies, a Riemann solver correction was considered against a density filter for preventing large fluctuations and inaccuracies in the pressure field (see Section 2.3.5). Lastly, in view of its simplicity, the adapted SPH formulation of Parshikov [1999] was selected.

Table 4.2. SPH-related parameters selected for the base case of the sensitivity analysis.

Parameter	Value
Initial inter-particle spacing (Δ)	0.0350m
Slope limiter (β)	1.3
Smoothing length (h)	0.0429m
Speed of sound (c_0)	46.2m/s
Time-stepping algorithm	Symplectic
Time-stepping coefficient (α_{time})	0.2
Smoothing kernel function	Quadratic
Kinematic viscosity (ν)	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Wall viscosity (ν_{wall})	$8 \times 10^{-5} \text{ m}^2/\text{s}$
XSPH Coefficient (ϵ)	0.5

4.2.2 Particle Spacing

The inter-particle spacing (Δ) is the distance that initially separates all particles in a SPH simulation. In the case of boundary particles, which are generally fixed in space, this spacing is constant throughout the simulation. In the case of water particles, this initial inter-particle spacing will dictate how many particles will be located within a given volume of water, and hence, this will influence the mass associated to each water particle.

Although the actual distance between water particles may change during the simulation (i.e. water particles with high pressure will be closer to each other while water particles with low pressure will be more distant), this initial inter-particle spacing is a good indicator of how close the water particles will be from one another during the simulation.

Figure 4.8 shows that the inter-particle spacing has significant influence on the net force acting on the column. When the spacing is too large, it seems that the magnitude of the force is reduced significantly. It is also clear that, as the latter is reduced (i.e. the particle resolution is increased), the solution not only converges but the initial peak at impact is more defined. Based on the results obtained by Silvester and Cleary [2006] (see literature review), this type of convergence was expected. A similar observation can also be made for the velocity of the bore-front, as the time of impact occurs slightly earlier as Δ decreases.

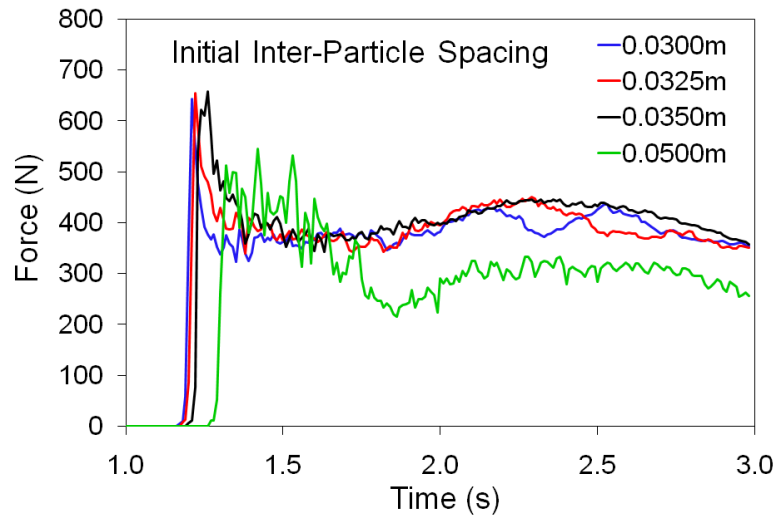


Figure 4.8. Time-history of the net base horizontal force acting on the column for various initial inter-particle spacing (Δ).

However, it is clear from Table 4.3 that, as the inter-particle spacing decreases, the time required for the simulations (i.e. the computational time) increases significantly. This was also expected, as the computational complexity of the serial version of the SPHysics code is $O(n \log_2 n)$ [Gómez-Gesteira *et al.*, 2010a], where n is the total number of particles in the simulation (i.e. including both fluid and boundary particles). In other

words, compared to a linear increase that is associated with a computational complexity of $O(n)$, the computational time increases more dramatically with n .

Table 4.3. Total number of particles and computational times for various inter-particle spacings (Δ).

Δ (m)	Number of Particles	Computational Time (days)
0.0300	275,841	8.41
0.0325	225,173	6.21
0.0350	182,104	4.49
0.0500	71,959	0.49

4.2.3 Slope Limiter

This parameter is related to the Riemann solver correction mentioned in Section 2.3.5 and described in Appendix A.

Figure 4.9 shows that the choice of the slope limiter (β) has a clear influence on the time-history of the resulting force acting on the column. As β increases, the magnitude of the force tends to increase. However, it seems that when a certain threshold value is reached, there is no further influence on the force. Moreover, it seems that the initial impact occurs earlier as β increases. The latter implies that as β decreases, the velocity of the bore-front decreases. As mentioned in Section 2.3.5, this supports the possible excessive diffusion or “numerical velocity” related to the use of Riemann solvers.

Also, while Table 4.4 shows that β has little influence on the computational time required to perform the simulation, it seems that the larger it is the slightly higher the computational time will be.

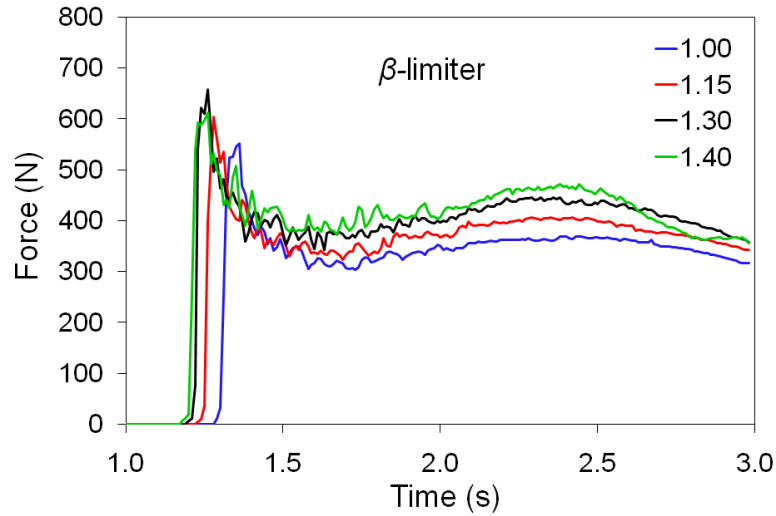


Figure 4.9. Time-history of the net base horizontal force acting on the column for various slope limiters (β).

Table 4.4. Computational times for various slope limiters (β).

β	Computational Time (days)
1.00	4.31
1.15	4.40
1.30	4.49
1.40	4.55

4.2.4 Smoothing Length

As depicted in Figure 2.2, the smoothing length (h) controls the size of the influence domain of the smoothing kernel at a given particle. This means that more neighbouring particles will contribute to the particle approximation as h increases (see Section 2.2.2). Hence, it is logical that, as the particle resolution increases, the number of neighbouring particles in the influence domain also increases and ultimately a smaller h is required. In the SPHysics implementation, the smoothing length depends on the selected particle resolution as follows:

$$[4.1] \quad h = \alpha_{smooth} \sqrt{\Delta_x^2 + \Delta_y^2 + \Delta_z^2}$$

where Δ_x , Δ_y , and Δ_z are the initial inter-particle spacing in the X, Y and Z directions. For all simulations in this thesis, the initial inter-particle spacing is considered the same in all directions (i.e. $\Delta_x = \Delta_y = \Delta_z$).

Figure 4.10 shows that, as the smoothing length increases, the initial peak in the force at impact also increases. However, when the smoothing length becomes too large, this peak eventually decreases. This can be explained by the fact that, as the smoothing length increases, more water particles are included in the particle approximation, and furthermore the fluid properties are being smoothed. In other words, a smoothing length too large will tend to average the fluid properties excessively and include particles that are remote from the region of interest. On the other hand, since the particle approximation is performed with an insufficient number of water particles, a smoothing length too small might not provide accurate results. As for the magnitude of the force after impact, the choice of the smoothing length seems to have a lesser influence, although the aforementioned averaging effect can still be observed. This could perhaps be explained by the lower pressure gradients following initial impact.

As for the computational time, Table 4.5 shows that the smoothing length has some influence on the time required for the simulation. As the smoothing length increases, the computational time increases as well. This can be explained by the greater number of neighbouring water particles involved per particle approximation and the accompanying greater amount of calculations.

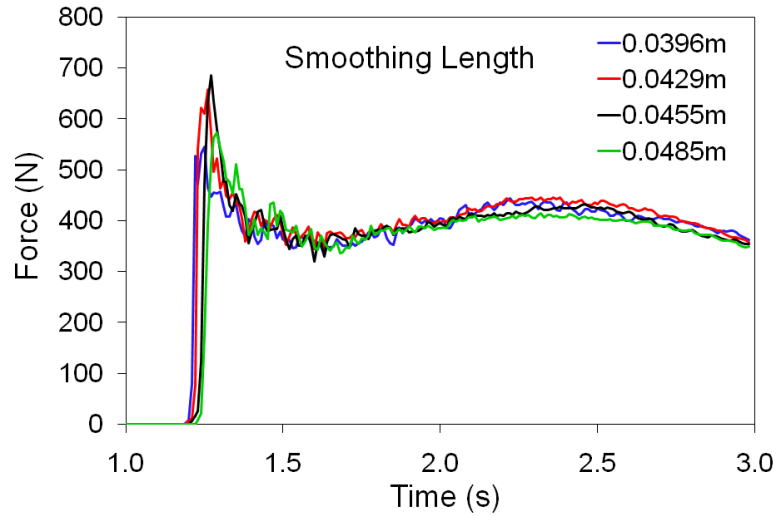


Figure 4.10. Time-history of the net base horizontal force acting on the column for various smoothing lengths (h).

Table 4.5. Computational time for various smoothing lengths (h).

α_{smooth}	h (m)	Computational Time (days)
0.65	0.0396	4.07
0.71	0.0429	4.49
0.75	0.0455	4.82
0.80	0.0485	5.29

4.2.5 Speed of Sound

The speed of sound in water at the reference density (c_0) is involved in many aspects of the WCSPH simulations. These aspects are: the calculation of the pressure field through an equation of state (Eq. 2.66), the determination of the time step based on the Courant condition (Eq. 2.79 indirectly through Eq. 2.69), and the determination of the repulsive boundary forces (Eq. 2.111 indirectly through Eq. 2.69). In order to ensure that the speed of sound is at least ten times greater than the maximum flow speed, Monaghan [1994] suggested selecting the speed of sound as follows:

$$[4.2] \quad c_0 = \alpha_{sound} \sqrt{gh_u}$$

where $\sqrt{gh_u}$ is the approximate speed of a dam-break wave with water height h_u , and α_{sound} would normally be taken as 10 in order to fulfil the aforementioned condition.

Surprisingly, c_0 does not significantly affect the magnitude of the force at initial impact (Figure 4.11). By inspection Eq. 2.66, one would expect that, as c_0 increases, the computed pressure would also increase. Therefore, the resulting force on the column would be larger as the former is obtained by integration of the numerical pressure (see Chapter 5 for the details on the post-processing of numerical results). In fact, Figure 4.11 shows that after the initial impact, the resulting force on the column is generally higher for smaller values of the speed of sound.

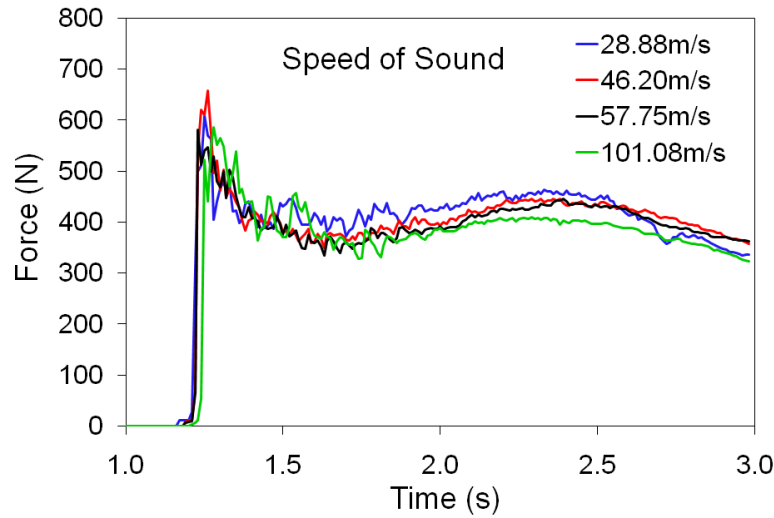


Figure 4.11. Time-history of the net base horizontal force acting on the column for various speed of sound at the reference water density (c_0).

Table 4.6 shows that higher values of c_0 will significantly increase the computational time. This can be explained by smaller time steps required to fulfill the Courant condition (Eq. 2.79).

Table 4.6. Computational time for various speeds of sound at the reference water density (c_0).

α_{sound}	c_0 (m/s)	Computational Time (days)
10	28.88	2.89
16	46.20	4.49
20	57.75	5.41
35	101.08	9.23

4.2.6 Additional Parameters

Further graphical results of the sensitivity analysis, which show the influence of other parameters, are provided in Appendix B. In summary, these results show that:

- Compared to the predictor-corrector algorithm (see Section 2.3.4), the symplectic time-stepping algorithm generates higher forces (and pressures) at initial impact.
- Within the range tested ($\alpha_{time} = 0.2$ to 0.4), the time step coefficient (see Section 2.3.4) has no effect on the simulation results, although its influence on the computational time is directly proportional. In other words, it would be possible to reduce the computational time by half only by multiplying this coefficient by a factor of two.
- The choice between the quadratic and cubic spline smoothing kernels has negligible influence on the results of the simulation (see Section 2.3.1).
- When using the laminar viscosity with Sub-Particle Scale (SPS) turbulence to model the diffusion terms of the momentum equation (see Section 2.3.3), realistic values of the kinematic viscosity (ν) have negligible influence on the results of the simulation. The values tested corresponded to water temperatures varying

from 0 to 30°C. To obtain results with more variation would require the selection of unrealistic or extreme viscosity values.

- The lower the wall viscosity (ν_{wall}), the greater the resulting forces (and pressures) acting on the column at and beyond initial impact. Deduced by earlier impact times, this can be explained by the reduced friction which ultimately increases the velocity of the bore (see Section 2.3.6).

4.2.7 Selected Values

Except for the particle resolution, the parameters considered for the above sensitivity analysis did not have an extreme influence on the results of the simulations. Hence, their values were selected such that (i) the resulting force was maximized at both impact and beyond, (ii) the diffusion effects would be minimized, and (iii) the computational time (T_{comp}) would be short. In the case of the particle resolution, it is clear that the higher the resolution (i.e. the smaller the initial inter-particle spacing), the more accurate the simulation results will be. However, as the computational time increases drastically with the resolution, a compromise was required in order to execute the simulations within a reasonable amount of time. The values of the SPH-related parameters, which were employed in the simulations discussed in the following sections, are presented in Table 4.7.

4.3. Computational Domains

For all five computational domains detailed below, the walls and bottom of the flume (or tank) along with the test structures were modeled using perfectly-fixed repulsive boundary particles. Assuming that the opening of the gate was sufficiently fast in all the previously described physical experiments (Section 4.1), the dam-break scenarios were considered to be ‘ideal’ according to Lauber and Hager [1998]. Hence, for the sake of simplicity, water in the numerical simulations was assumed to be instantaneously released and no gates were modeled.

Table 4.7. Final SPH-related parameters selected for numerical simulations.

Parameter	Value
Initial inter-particle spacing (Δ)	Variable
Slope limiter (β)	1.25
Smoothing length coefficient (α_{smooth})	0.725
Speed of sound coefficient (α_{sound})	16
Time stepping algorithm	Symplectic
Time stepping coefficient (α_{time})	0.4
Smoothing kernel function	Quadratic
Kinematic viscosity (ν)	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Wall viscosity (ν_{wall})	$8 \times 10^{-5} \text{ m}^2/\text{s}$
XSPH coefficient (ε)	0.5

4.3.1 Yeh and Petroff [n.d.]

The computational domain reproduced exactly the dimensions of the experimental setup described in Section 4.1.1. Being at a relatively small scale, the purpose of this simulation was to test in a short amount of time the procedure conceived for the post-processing of raw numerical outputs (Chapter 5). The initial configuration of the computational domain is shown in Figure 4.12 just moments after release of water in the reservoir, and details of the related simulations are presented in Table 4.8 (T_{sim} and T_{comp} are the physical time and computational time of the simulation, respectively). As explained in the following section, the experimental downstream water depth (h_d) of 0.01m could not be accurately modelled and a dry-bed condition was assumed instead. This simulation was performed using the serial version of SPHysics and the numerical outputs were obtained at a frequency of 100Hz.

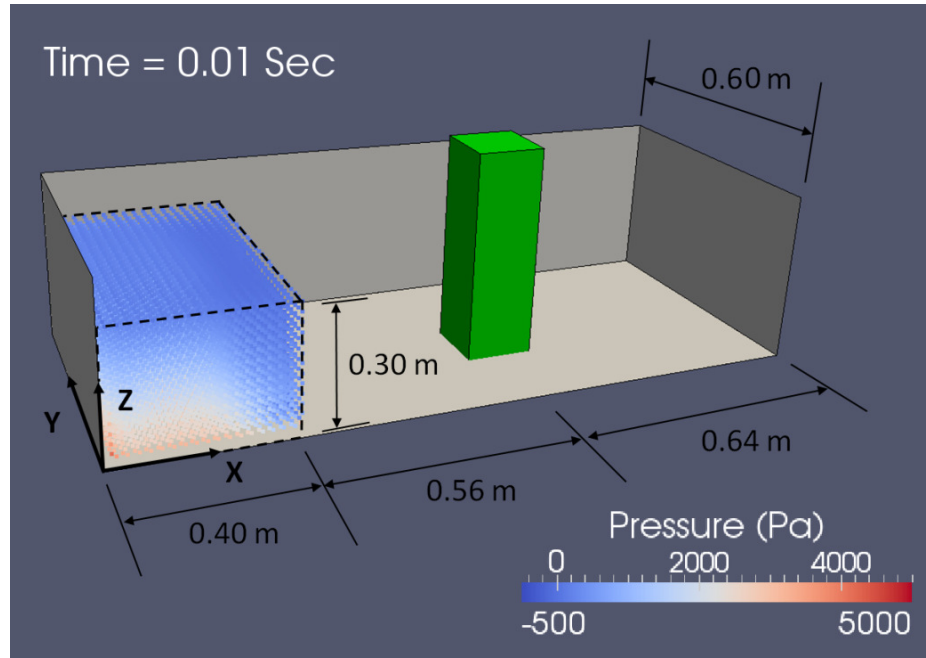


Figure 4.12. Initial configuration of computational domain for simulating the experiment of Yeh and Petroff [n.d.].

Table 4.8. Simulation details for the experiment of Yeh and Petroff [n.d.].

Simulation Label	h_u (m)	h_d (m)	Δ (m)	Number of Particles	CPUs	T_{sim} (sec)	T_{comp} (days)
D1	0.30	0.00	0.0225	21,232	1	2.4	0.10
D2	0.30	0.00	0.0125	105,037	1	2.4	2.08

4.3.2 Nouri et al. [2010] and Al-Faesly et al. [2012]

These experiments were simulated in two ways: (a) using a computational domain that reproduced exactly their actual geometry, and (b) using computational domains with an adapted geometry. The purpose of (a) was to compare quantitatively numerical results with experimental ones in order to validate the numerical model (Chapter 6). In the case of (b), the goal was to investigate, without introducing excessive computational costs, various aspects that will be discussed later.

Due to the repulsive boundary condition considered, it was not possible to reproduce the exact thickness of the thin water layer initially present in the physical experiment downstream of the gate prior to its opening. This is because repulsive forces exerted on the fluid particles by the boundary particles are too strong, magnitude-wise, when compared to the hydrostatic force of the still fluid acting on the bottom of the channel. This results in fluid particles being repulsed to an elevation considerably higher than the experimental $h_d \approx 0.005\text{m}$. Consequently, unless mentioned otherwise, a dry-bed condition was assumed here for the simulations and therefore, some discrepancy at initial impact was to be expected (see Figure 3.11).

Actual Geometry

This computational domain reproduced exactly the dimensions of the experimental setup described in Section 4.1.2. No downstream wall was considered in order to allow fluid particles to leave the computational domain once they reached the end of the flume. This way, the effect of the floor drain present in the physical experiment was accurately reproduced. Figure 4.13 shows the initial configuration of the computational domain just moments after the release of the water in the reservoir. Details of the related simulations are presented in Table 4.9. Also, due to the large size of the computational domain with respect to the selected resolution, the related numerical simulations were performed with the parallel version of SPHysics (i.e. *ParallelSPHysics*), as indicated by the number of CPUs greater than one (Table 4.9). Lastly, due to the limited hard disk memory, numerical outputs were usually obtained at a frequency of 100Hz. However, to ensure proper capture of the sudden rise in pressure and force, for simulations D2-55 and D2-85, a high sampling frequency of 1000Hz was used over short intervals covering initial impact.

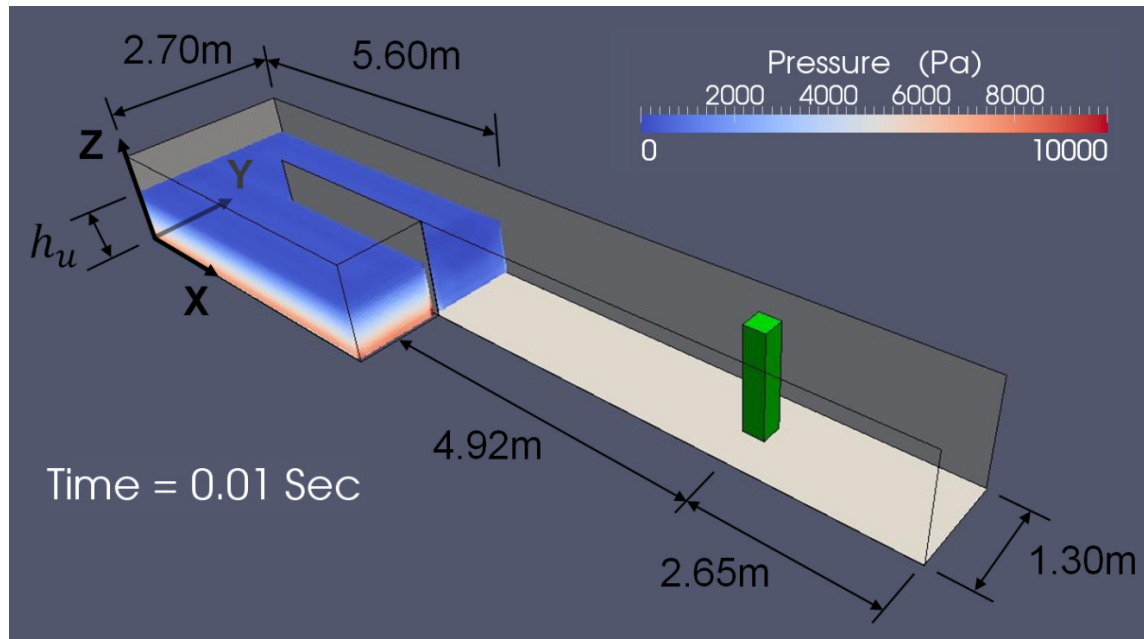


Figure 4.13. . Initial configuration of computational domain for simulating the experiments of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] shown with a square column.

Table 4.9. Simulation details for the experiment of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012].

Simulation Label	Column	h_u (m)	h_d (m)	Δ (m)	Number of Particles	CPUs	T_{sim} (sec)	T_{comp} (days)
D2-55	Square	0.55	0.00	0.035	459,260	32	10.0	4.86
D2-85	Square	0.85	0.00	0.035	654,648	32	10.0	6.94
D2-85-C	Circular	0.85	0.00	0.035	653,106	32	10.0	6.93
D2-85-N	None	0.85	0.00	0.035	648,060	32	10.0	6.88
D2-115	Square	1.15	0.00	0.035	849,580	32	10.0	10.42

Adapted Geometries

Two adapted computational domains based on the experiments of Al-Faesly *et al.* [2012] were employed to investigate the influence of reservoir geometry (bend vs. straight), reservoir length (L), flow width (W), particle resolution, and the influence of the downstream bed condition (i.e. dry vs. wet). The bed condition was investigated with respect to not only the propagation of bores but more importantly to the resulting hydrodynamic forces acting on a free-standing square column (Figure 4.14) and a vertical wall (Figure 4.15). In the runs where the influence of a wet bed is investigated, h_d corresponds to a value greater than the one used in the physical experiment (0.080m vs. 0.005m). Hence, only a qualitative comparison between numerical results was performed in this case (Section 7.4). In fact, as a result of the high strength of the repulsive forces exerted by boundary particles on the fluid particles, this h_d was the smallest one possible. Figure 4.14 shows the initial configuration of the general computational domain before the release of the water in the reservoir, and details of the related simulations are presented in Table 4.10.

In the case of the second adapted computational domain, the location of the wall with respect to the downstream end of the reservoir corresponds to the same location as the upstream face of the column considered in the experiments of Al-Faesly *et al.* [2012] (i.e. Figure 4.13 and Figure 4.14). As this wall covers the entire width of the flume, it prevents any fluid particles from leaving the computational domain (i.e. no floor drain). Figure 4.15 shows the initial configuration of this second adapted computational domain prior to release of water in the reservoir and details of the related simulations are presented in Table 4.11.

In general, numerical outputs were obtained at a frequency of 100Hz for the simulations performed for the adapted geometries described above. However, for runs D3-L1 and D3-L3 a high sampling frequency of 1000Hz was used over short intervals covering initial wave impact, while for runs D3-DRY, D3-WET, D4-DRY, and D4-WET, a sampling frequency of 200Hz was used throughout their entire durations.

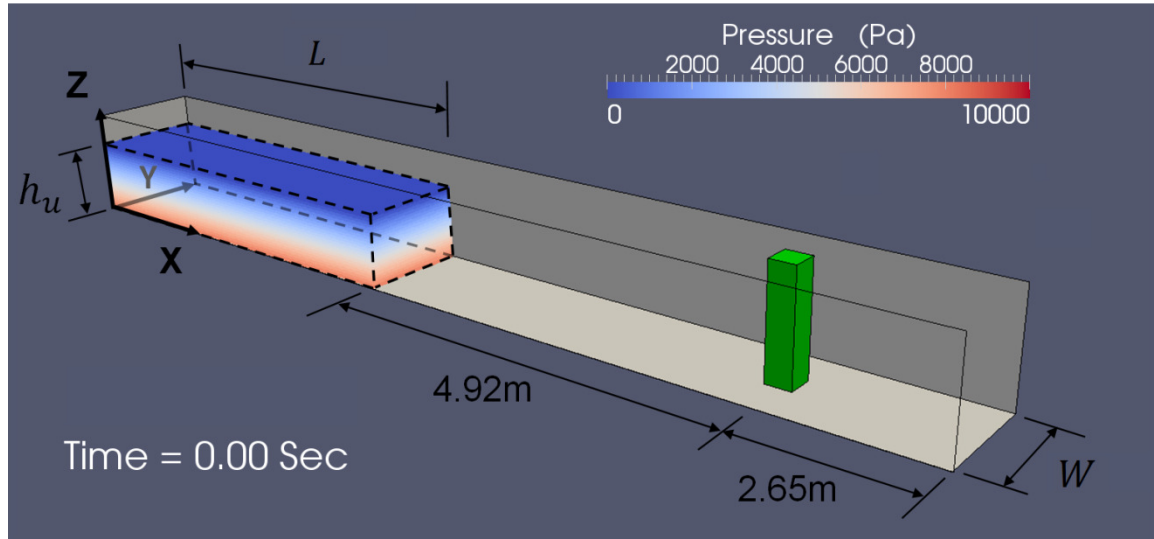


Figure 4.14. Initial configuration of computational domain for simulating the general experiment based on Al-Faesly *et al.* [2012] shown with a dry bed.

Table 4.10. Simulation details for general experiments based on Al-Faesly *et al.* [2012].

Simulation Label	h_u (m)	h_d (m)	L (m)	W (m)	Δ (m)	Number of Particles	CPU s	T_{sim} (sec)	T_{comp} (days)
D3-HIGH	0.85	0	6.00	1.30	0.025	933,093	32	10.0	12.51
D3-WIDE	0.85	0	6.00	2.60	0.035	666,513	32	10.0	7.07
D3-L1	0.85	0	6.00	1.30	0.035	341,514	32	10.0	3.47
D3-L2	0.85	0	11.20	1.30	0.035	611,794	32	10.0	6.48
D3-L3	0.85	0	15.00	1.30	0.035	817,876	32	11.1	10.03
D3-DRY	0.85	0	5.00	1.30	0.035	291,472	1	5.0	6.41
D3-WET	0.85	0.08	5.00	1.30	0.035	298,852	1	5.0	6.53

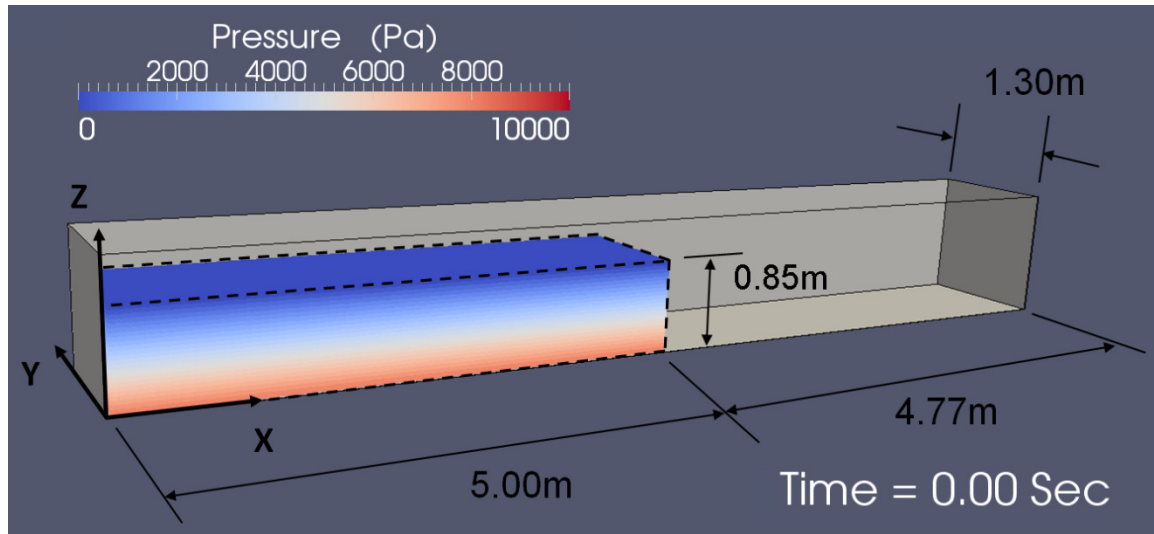


Figure 4.15. Initial configuration of computational domain for the experiments with wall test structure based on Al-Faesly *et al.* [2012] shown with a dry bed.

Table 4.11. Simulation details for the experiment with wall test structure based on Al-Faesly *et al.* [2012].

Simulation Label	h_u (m)	h_d (m)	Δ (m)	Number of Particles	CPU's	T_{sim} (sec)	T_{comp} (days)
D4-DRY	0.85	0.00	0.035	281,456	1	5.0	6.14
D4-WET	0.85	0.08	0.035	286,316	1	5.0	6.32

4.3.3 Kleefsman et al. [2005]

This computational domain reproduced exactly the dimensions of the experimental setup described in Section 4.1.3. The purpose of the related simulation was to extend the validation of the numerical by comparing the numerical results to previous studies (Chapter 6), and to investigate the influence of the particle resolution on the “numerical” viscosity associated to Riemann solvers. Figure 4.16 shows the initial configuration just moments after release of water in the reservoir, and simulation details are presented in Table 4.12. The simulations were performed using the serial version of SPHysics and

numerical outputs were obtained at a frequency of 100Hz for simulations D5-LOW and D5-MED, and 200Hz for simulation D5-HIGH.

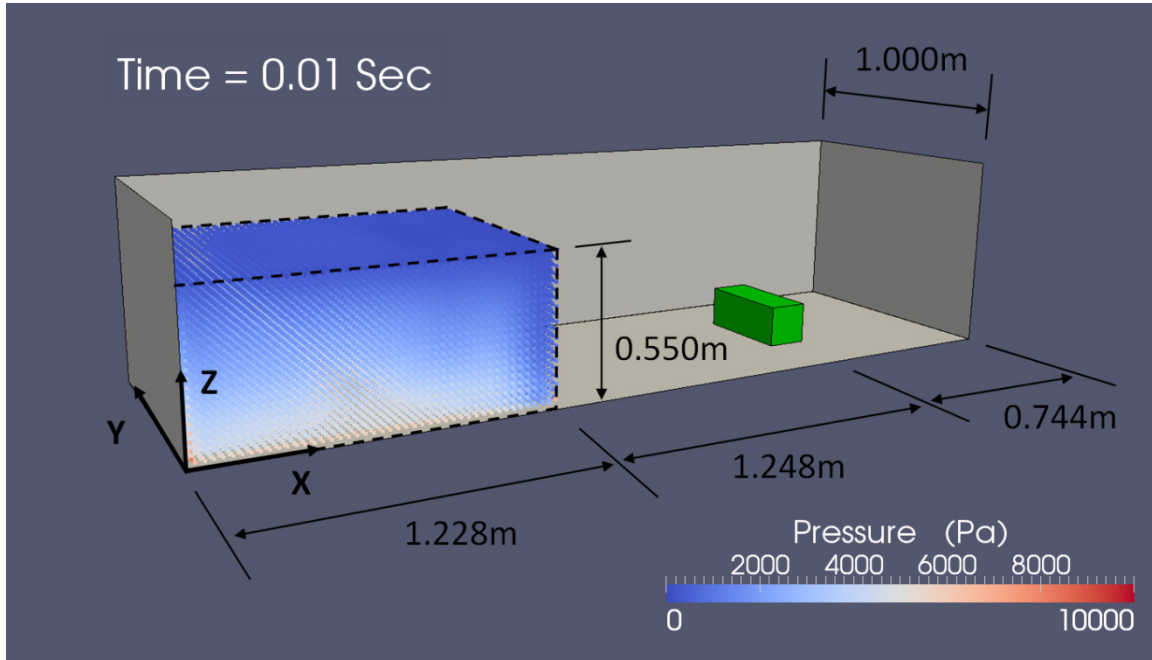


Figure 4.16. Initial configuration of computational domain for simulating the experiment of Kleefsman *et al.* [2005].

Table 4.12. Simulation details for the experiment of Kleefsman *et al.* [2005].

Run Label	h_u (m)	h_d (m)	Δ (m)	Number of Particles	CPUs	T_{sim} (sec)	T_{comp} (days)
D5-LOW	0.55	0.00	0.0350	77,690	1	6.0	2.92
D5-MED	0.55	0.00	0.0200	192,060	1	6.0	9.24
D5-HIGH	0.55	0.00	0.0155	397,412	1	6.0	29.25

Chapter 5. Post-Processing of Numerical Outputs

Since the SPHysics implementation of the SPH method can be used to simulate various types of flow problem with very different configurations, no means is provided within source code for obtaining specific results from the simulations. Therefore, it was necessary to implement ways to do so and two approaches which required additional coding are described and evaluated in this chapter. The first one takes advantage of the repulsive forces of boundary particles acting on fluid particles already being calculated in the simulation (Eq. 2.113 and Eq. 2.118), making its implementation within the SPHysics source code relatively straightforward. On the other hand, requiring the full implementation of external programs, the second approach consists in “sampling” particles located within a certain region of interest by searching the raw output files created by the numerical model.

5.1. Repulsive Force of Boundary Particles

This approach is for the determination of the resulting hydrodynamic force acting on an arbitrary object. Based on the concept of the repulsive boundary treatment discussed Section 2.3.6 and following the work of Rogers *et al.* [2010], the resultant hydrodynamic force acting on a solid object (fixed or moving) can be obtained by summing the repulsive forces exerted by the surrounding fluid particles on all boundary particles that compose this object. By the principle of equal and opposite action, the force per unit mass $\mathbf{f}_{ki}$ exerted on a boundary particle k due to a fluid particle i is given by:

$$[5.1] \quad m_k \mathbf{f}_{ki} = -m_i \mathbf{f}_{ik}$$

where m_i and m_k are the masses of fluid particle i and boundary particle k , respectively. Subsequently, the total force per unit mass $\mathbf{f}_k$ acting on a boundary particle k due to all its N neighbouring fluid particles is given by:

$$[5.2] \quad \mathbf{f}_k = \sum_{i=1}^N \mathbf{f}_{ki}$$

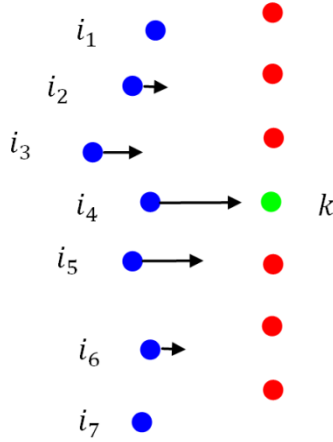


Figure 5.1. Total force per unit mass $\mathbf{f}_k$ on a boundary particle k due to neighbouring fluid particles (normal components shown only).

Schematically, $\mathbf{f}_k$ is represented in Figure 5.1 above. Rearranging Eq. 5.1 and Eq. 5.2 yields:

$$[5.3] \quad \mathbf{f}_k = - \sum_{i=1}^N \frac{m_i \mathbf{f}_{ik}}{m_k}$$

Note that, the force per unit mass $\mathbf{f}_{ik}$ on particle i due to a boundary k in the above equations is given by the summation of the normal component (Eq. 2.113) and the radial component due to wall friction (Eq. 2.118). In order to obtain the resultant force vector $\mathbf{F}_{object}$ on the object, the summation of the resultant force per unit mass on each of its composing boundary particles multiplied by their associated weights is performed as follows:

$$[5.4] \quad \mathbf{F}_{object} = \sum_{k=1}^{B_{object}} m_k \mathbf{f}_k$$

where B_{object} is the total number of boundary particles forming the solid object. Rearranging Eq. 5.3 and Eq. 5.4 yields:

$$[5.5] \quad \mathbf{F}_{object} = - \sum_{k=1}^{B_{object}} m_k \sum_{i=1}^N \frac{m_i \mathbf{f}_{ik}}{m_k}$$

Finally, since the mass of the boundary particles is constant over a simulation,

$$[5.6] \quad \mathbf{F}_{object} = - \sum_{k=1}^{B_{object}} \sum_{i=1}^N m_i \mathbf{f}_{ik}$$

One advantage of this approach is that the repulsive boundary forces $\mathbf{f}_{ik}$ exerted on fluid particles by boundary particles are already being calculated within the SPHysics source code. Hence, it is only required to add the necessary code instructions in order to have the resulting force acting on an arbitrary object given by Eq. 5.6 written to an additional ASCII file. However, this approach does not provide any measurements of pressure, velocity, or water surface elevation.

5.2. Particle Sampling

Since the SPH method is based on a Lagrangian description, the fluid particles carry along their physical properties as they travel across the computational domain. This can become a disadvantage when it is required to determine the value of a flow property at a specific point in the fluid that does not necessarily correspond to the location of a water particle. Inspired by Gesteira and Dalrymple [2004] the approach, referred to hereafter as *particle sampling*, consists mainly in first searching for the fluid particles that are within

a certain sub-region of the computational domain, or *sampling volume*, and then in approximating the value of a particular flow property, or field variable, based on the properties associated to these “sampled” fluid particles. This approximated flow property would then be assumed to be measured at the center of the sampling volume.

Generated at a specified frequency, the raw outputs of the SPHysics source code used here consist of ASCII files in which each line contains the properties of a single particle in the simulation (e.g. position, velocity, density, mass, pressure, etc.). In other words, each of these files could be considered a “snapshot” of the computational domain at a given instant during the simulation [note that, for all of these output files, the line index of a particle is constant over the simulation and independent of the actual position of this particle within the computational domain]. In order to obtain results from these files using particle sampling as explained above, a series of object-oriented post-processing programs were implemented in C++. The entire source code of the latter along with user guidance can be found in Appendix C.

In this study, the flow property within a single sampling volume is calculated via a kernel approximation (discretized) as stated by Eq. 2.22. The shape of this sampling volume (a sphere of radius $2h$) was selected in analogy to the influence domain of the smoothing kernel as depicted in Figure 2.2. As an example, taking the velocity as the flow property to be evaluated:

$$[5.7] \quad \bar{\mathbf{u}} = \sum_{i=1}^{N_{volume}} \frac{m_i}{\rho_i} \mathbf{u}_i W(\mathbf{x}_i - \mathbf{x}_v, h)$$

where N_{volume} is the total number of fluid particles within the sampling volume v , and $\mathbf{x}_v$ is the position vector of the center of this sampling volume. The selection of the radius of the sampling volume is an important consideration and will be discussed in Section 5.3.2. Firstly, to ensure that the contributions of fluid particles, which are remote from the region of interest, are insignificant, the selected radius should not be too large. On the other hand, if it is too small, it is possible that too few water particles will be within the sampling volume. This behaviour will be discussed further in Section 5.3.2 in regards to

selection of the post-processing smoothing length. When calculating the density as the flow property, Eq. 5.7 leads to:

$$[5.8] \quad \bar{\rho} = \sum_{i=1}^{N_{volume}} \frac{m_i}{\rho_i} \rho_i W(\mathbf{x}_i - \mathbf{x}_v, h) = \sum_{i=1}^{N_{volume}} m_i W(\mathbf{x}_i - \mathbf{x}_v, h)$$

As discussed in Section 2.3.5 when dealing with truncated influence domains, for which the normalization condition is not satisfied (Eq. 2.4), the correction suggested by Bonet and Lok [1999] can be applied to both Eq. 5.7 and Eq. 5.8. For instance, Eq. 5.8 becomes:

$$[5.9] \quad \bar{\rho} = \sum_{i=1}^{N_{volume}} m_i W(\mathbf{x}_i - \mathbf{x}_v, h) \bigg/ \sum_{i=1}^{N_{volume}} \frac{m_i}{\rho_i} W(\mathbf{x}_i - \mathbf{x}_v, h)$$

As for the pressure, it can be calculated using the equation of state (Eq. 2.66) based on the density given by either Eq. 5.8 or Eq. 5.9. While the speed of sound at the reference water density (c_0) in Eq. 2.66 applied in the post-processing should be the same as the one considered in the simulation, the smoothing length (h) in Eq. 5.7, Eq. 5.8, and Eq. 5.9 can be different.

5.2.1 Resultant Force on Column

Of most interest here, the net horizontal force acting streamwise (i.e. X direction) on a column can be determined by integrating the pressure exerted over its surface area. In the case of a circular column, the surface area is first divided into tributary areas as depicted in Figure 5.2. The sampling volumes are located adjacent to and at the center of their corresponding tributary areas. Hence, the centers of the sampling volumes are equidistant from one another. In order to determine whether a water particle is within a particular sampling volume or not, its distance from the center of the sampling volume is compared to the radius of that volume. Once all the water particles within each of the sampling volumes have been listed, the approximated pressure $\bar{P}$ within the sampling volume is

obtained by first calculating the average density ($\bar{\rho}$) according to Eq. 5.9, and then by applying the equation of state (Eq. 2.66). The streamwise resultant force F_X acting on the column is then calculated as follows:

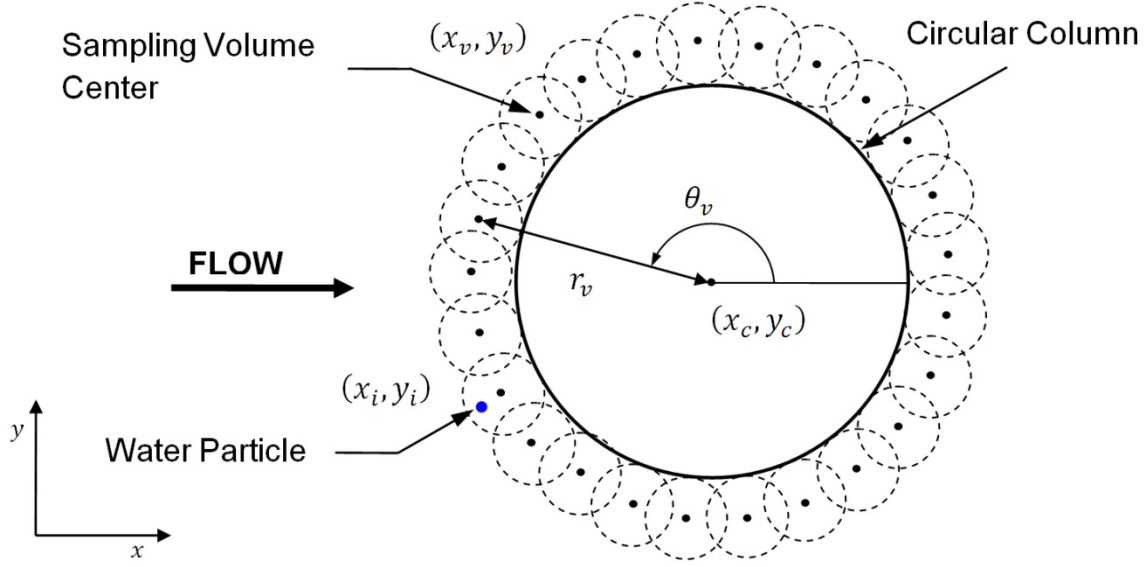


Figure 5.2. Plan view of circular column surrounded with spherical sampling volumes of radius $2h$.

$$[5.10] \quad F_X = \int_A \bar{P} \cos \theta_F dA$$

$$[5.11] \quad dA = \Delta\theta_v r_c \Delta z_v$$

and

$$[5.12] \quad \theta_F = \theta_v - \pi$$

where r_c is the radius of the circular column, Δz_v is the difference in elevation between the centers of overlaid sampling volumes (i.e. the height of the tributary areas), and $\Delta\theta_v$ is the difference in orientation between two adjacent sampling volumes (i.e. the angle corresponding to the arc of the tributary areas). The force contribution of a single

sampling volume is shown schematically in Figure 5.3. Lastly, this integration procedure was carefully developed by taking into consideration the possibility of having both positive and negative pressure acting at any location on the column.

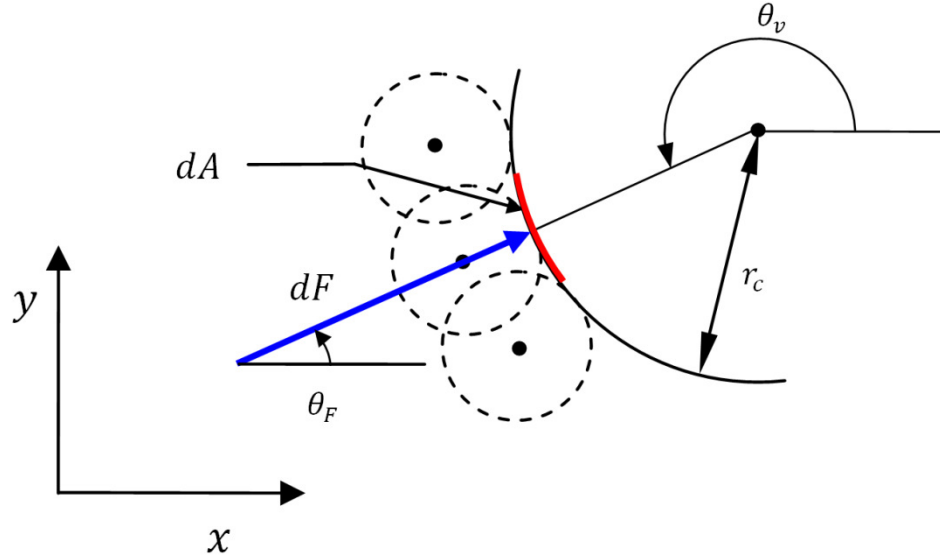


Figure 5.3. Force contribution dF of a single sampling volume acting on tributary area dA .

5.2.2 Water Surface Elevation

The surface elevation of the fluid can be obtained by placing *virtual wave gages* within the computational domain. These virtual wave gages are represented by a single sampling volume elongated along its height (Figure 5.4). The surface elevation at the center of the sampling volume is assumed to be equal to the elevation of the highest water particle located within it. Judgement is required when evaluating the water surface using this approach when splashing or wave breaking occurs. In analogy to the size of the spherical sampling volumes discussed in the previous section, the narrower the length and width of this elongated sampling volume is, the more precise the determination of the free-surface will be. Nevertheless, if these dimensions are insufficient with respect to the particle resolution, it is possible that the highest water particle within the sampling volume (if any) will be situated at an elevation below the local free-surface.

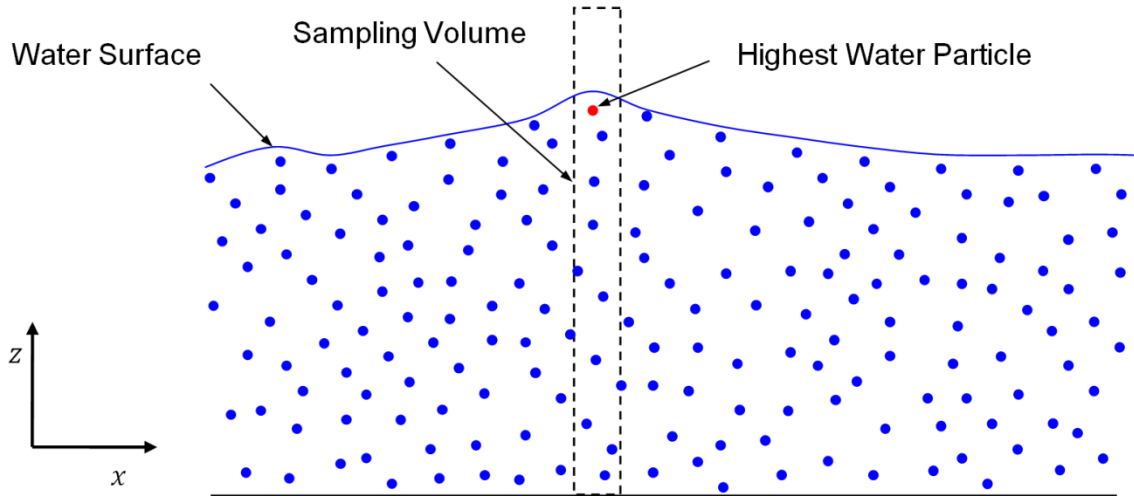


Figure 5.4. Schematic of virtual wave gage for the determination of the water surface elevation.

5.3. Performance and Adjustments

In the following sub-sections, the approaches for obtaining specific results from the simulations described previously are evaluated in order to identify the preferred ones which would be used throughout this study. Lastly, some adjustments were made in an attempt to improve their effectiveness.

5.3.1 Repulsive Force of Boundary Particles

The summation of the repulsive forces by the boundary particles (Eq. 5.6) is used to obtain the net force acting on the column for simulations D1 and D2, which correspond to the experiment of Yeh and Petroff [n.d.]. With marginal agreement, Figure 5.5-a shows a comparison of numerical and experimental time-history of the net force acting on the column for simulations with Δ of 0.0225 (D1) and 0.0125m (D2). It should be noted that the second peak in the force (negative) is explained by the reflection of the hydraulic bore on the downstream wall present in the experiment (see Figure 3.16 and Figure 4.12). The discrepancy of approximately 0.05 second in the occurrence time of initial impact, which translates into a slightly slower experimental bore-front velocity, can most likely be attributed to the 1.0 cm layer of water (h_d) present over the bottom the experimental box

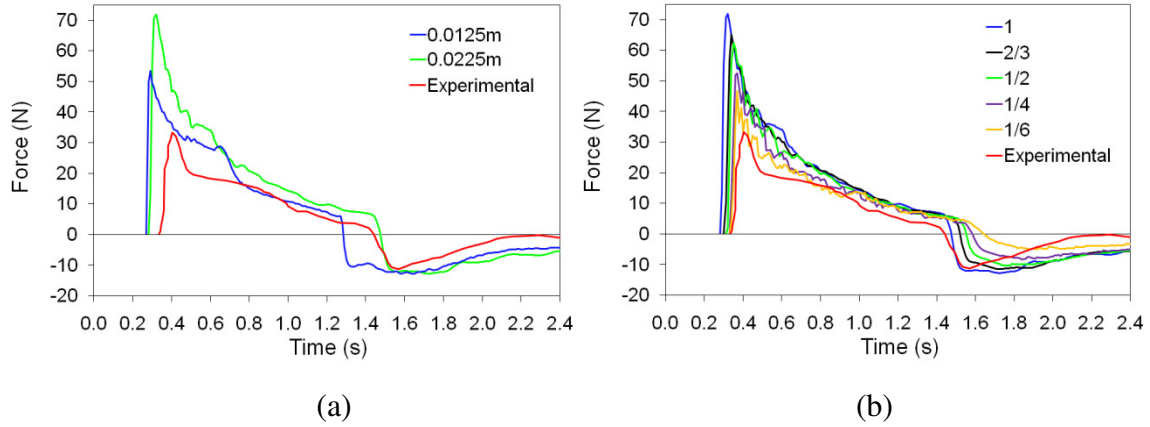
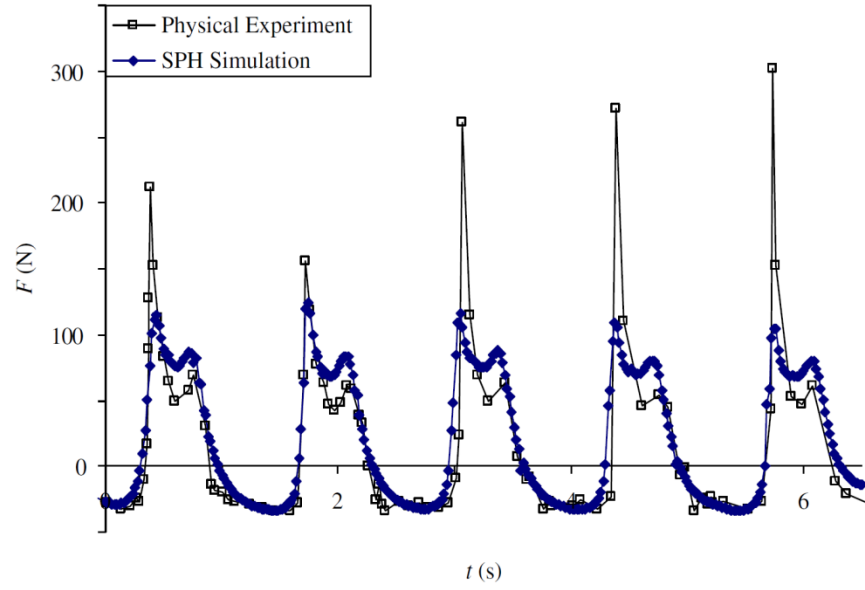
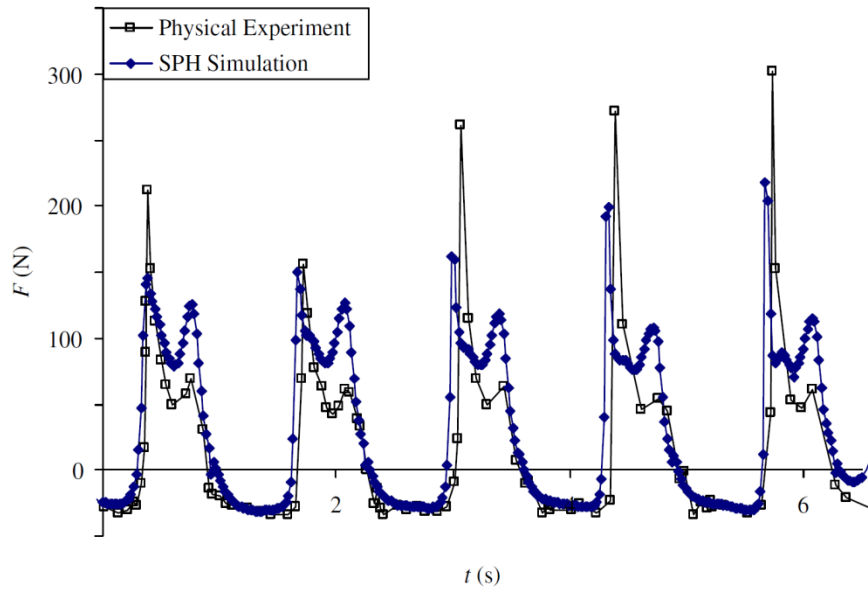


Figure 5.5. Comparison of numerical and experimental time-history of the net force acting on the column obtained through the summation of boundary repulsive forces: (a) with different initial inter-particle spacings (simulations D1 and D2); (b) with various speed of sound reduction factor (κ) for simulation D1.

not being simulated. As discussed in Section 4.3.2, this omission is due to the difficulty in modeling shallow water depths when using the repulsive boundary condition implemented in SPHysics. In any case, it is clear that the discrepancy in the magnitude of the initial impact force is considerable, even with an increased particle resolution (Figure 5.5-a). However, as noticed by Rogers *et al.* [2010], an increase in the resolution generally increases the overall accuracy (Figure 5.6). They also used SPHysics along with the same approach to determine the resulting force acting on a breakwater caisson due the impact of incoming waves. With similar degrees of discrepancy, their results also show some difficulty in predicting the peaks in the resulting force, although the experimental results are both over and under predicted. However, in this study, this force is only being over predicted when considering the experiment of Yeh and Petroff [n.d.] (Figure 5.5-a). Interestingly, with respect to the impulsive peaks, the agreement observed by Rogers *et al.* is improved with the increase in resolution, although the agreement of the subsequent peaks is not as good (Figure 5.6-a and Figure 5.6-b). Furthermore, it can be observed in the results of Rogers *et al.* [2010] that the occurrence time of the impulsive peaks occurs generally sooner with a high resolution (Figure 5.6-b). On the other hand, this occurrence time is in better agreement with the physical results in the case of the lower resolution (Figure 5.6-a). In the current study, as the particle resolution



(a)



(b)

Figure 5.6. Numerical and experimental time-history of the wave-induced horizontal force acting on a caisson breakwater obtained through the summation of boundary repulsive forces by Rogers *et al.* [2010] with an initial particle spacing (Δ) of (a) 0.010m, and (b) 0.005m.

is varied, the occurrence times of the peaks also vary (Figure 5.5-a). The time difference between the positive peak (initial impact) and negative peak (reflected wave) decreases from 1.24 to 1.05 seconds from the lower resolution to the higher resolution. This time difference is 1.16 seconds in the case of the physical experiment.

As the resulting force obtained by the summation of the repulsive forces exerted by the boundary particles generally overestimates the experimental value (Figure 5.5-a), it was concluded that, given their formulation described in Section 2.3.6, these repulsive forces could perhaps be too strong. In a relatively quick attempt to address this issue, the strength of these repulsive forces was reduced by applying a reduction factor κ to the particle speed of sound (c_i) in Eq. 2.111 as follows:

$$[5.1] \quad A = \frac{1}{h} \{0.01(\kappa c_i)^2 + \xi(\kappa c_i)[(\mathbf{u}_i - \mathbf{u}_k) \cdot \mathbf{n}]\}$$

As c_i depends on the speed of sound in water at the reference density c_0 (according to Eq. 2.69), in order to avoid altering the compressibility of the water, the latter was not reduced directly. Figure 5.5-b shows that better agreement is obtained as the value of κ decreases, although the agreement in the negative portion of the time-history of the force deteriorates. Another disadvantage of this modification is that, if the strength of the repulsive forces is reduced further (i.e. $\kappa < 1/6$), water particles will start flowing out of the computational domain. This is because the repulsive forces associated with the boundary particles become less important compared to the hydrostatic forces at the bottom of the water reservoir.

Finally, as mentioned in the literature review (Section 3.4), Silvester and Cleary [2006] also calculated the resulting force acting on the column for the physical experiment of Yeh and Petroff [n.d.] by summing of the repulsive forces exerted by the boundary particles (Figure 3.17). Although their numerical results slightly overestimate the peaks in the time-history of the force, relatively good agreement is obtained. It should be noted that the formulation of the repulsive boundary condition they considered is different than the one implemented in SPHysics.

5.3.2 Particle Sampling

Determination of Pressure

For the determination of the pressure exerted on the upstream face of the column, sampling volumes were initially placed such that their outer surface would be adjacent to the boundary particles forming the structure (Figure 5.2). With this arrangement, unrealistic pressure time-histories were obtained. During visualization, it was noticed that, along the entire submerged height of the column, a vertical layer of water particles with low pressures would form between the incoming flow and the column's upstream face, and that more realistic pressures would only develop further upstream. Based on the experimental observations of Nouri *et al.* [2010], it is possible for negative pressures to develop, but only at higher elevations on the column (e.g. higher than 30cm), where a flow jet is reflected upstream. Furthermore, after initial bore impact and as a result of a decrease in the flow velocity, a void of particles would gradually form between approaching water particles and the boundary particles forming the column. One could view this void as the result of the repulsive forces by the boundary particles reaching equilibrium with the incoming flow momentum. In a plane close to the flume's bottom, Figure 5.7 and Figure 5.8 demonstrate the development of unrealistically low pressures and a void of particles at the front of the square and circular column in simulations D2-85 and D2-85-C, respectively (for all figures in which an X axis is displayed, the origin considered corresponds to the downstream end of the reservoir). Also, for all figures showing still images from the numerical simulations, the black lines represent solid boundaries.

In order to verify if this is unrealistic, or if such unphysical layer can be avoided by increasing the particle resolution, Figure 5.9 shows a time sequence of the bore-column impact but for a simulation with an initial inter-particle spacing of 0.025m (simulation D3-HIGH) rather than 0.035m (simulations D2-85 and D2-85-C). Unfortunately, this unrealistic layer still forms and its effects do not seem to be attenuated by using a higher resolution. Note that, with respect to the time of initial impact, the frames shown in Figure 5.9 correspond to the same ones as in Figure 5.7 and

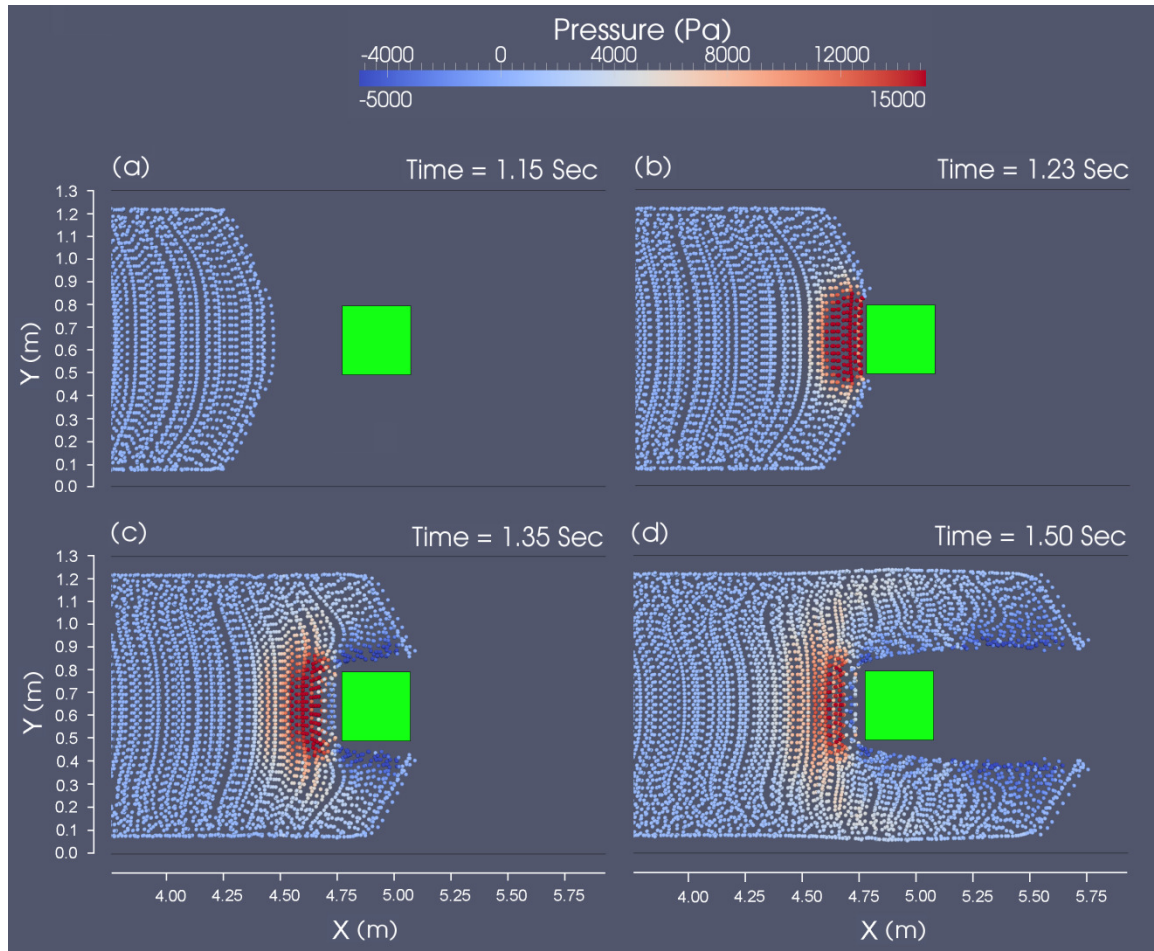


Figure 5.7. Region of unrealistic pressures and void developing beyond initial impact upstream of the square column in a plane close the flume’s bottom (simulation D2-85).

Figure 5.8. This difference in time can be explained by the fact that the convergence associated with an increase in the resolution tends to produce slightly faster bore-front velocities (see Figure 4.8 of the sensitivity analysis included in Chapter 4). In a mathematical and numerical study on errors related to the SPH method, Vaughan *et al.*, [2007] mentioned the occurrence of a “numerical boundary layer” when considering a repulsive type boundary condition. According to their observations, a layer of closely packed fluid particles can form near a boundary, very similar to what can be seen in Figures 5.7, 5.8, and 5.9 near the lateral boundaries shown by black lines. Their explanation for this phenomenon is as follows: In order to oppose the lateral hydrostatic force, a pressure gradient (i.e. a density gradient) must develop. In order for the density to increase at a particular point, particles must become more packed. Where a particle has many

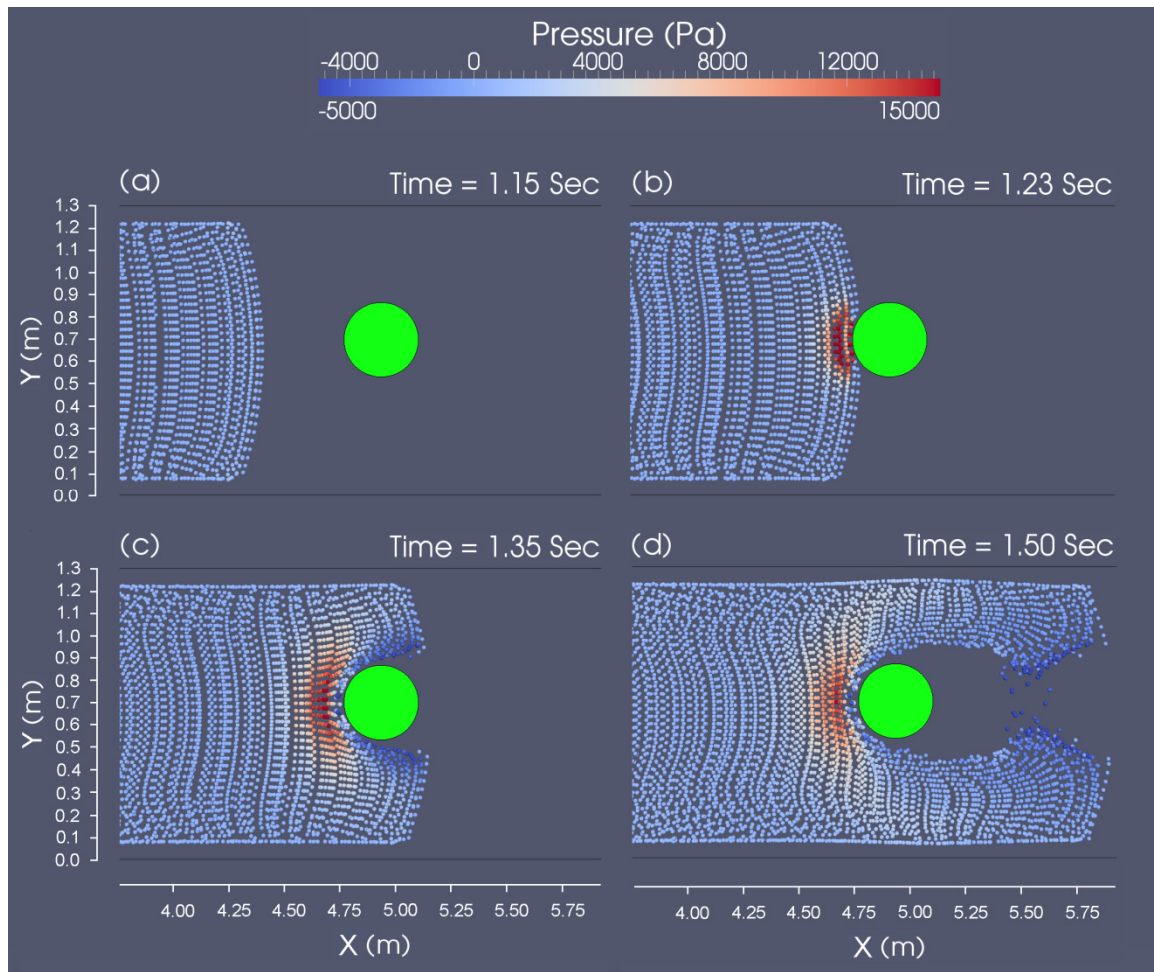


Figure 5.8. Region of unrealistic pressures and void developing beyond initial impact upstream of the circular column in a plane close the flume's bottom (simulation D2-85-C).

neighbours, each needs only to move a relatively small distance to cause the density to increase. Conversely, where a particle has few neighbours (as in the case close to a boundary due to its influence domain being truncated (Figure 2.5)), this particle and its neighbours must move much more. Although this explains the clustering of fluid particles near boundaries, it does not explain why unrealistically low pressures develop directly upstream of the column (e.g. Figure 5.7-b). It should be noted that, for any other solid boundary in the computational domain subjected to hydrostatic pressure, realistic pressure gradients are observed. For instance, Figure 5.10 shows (simulation D2-85) the

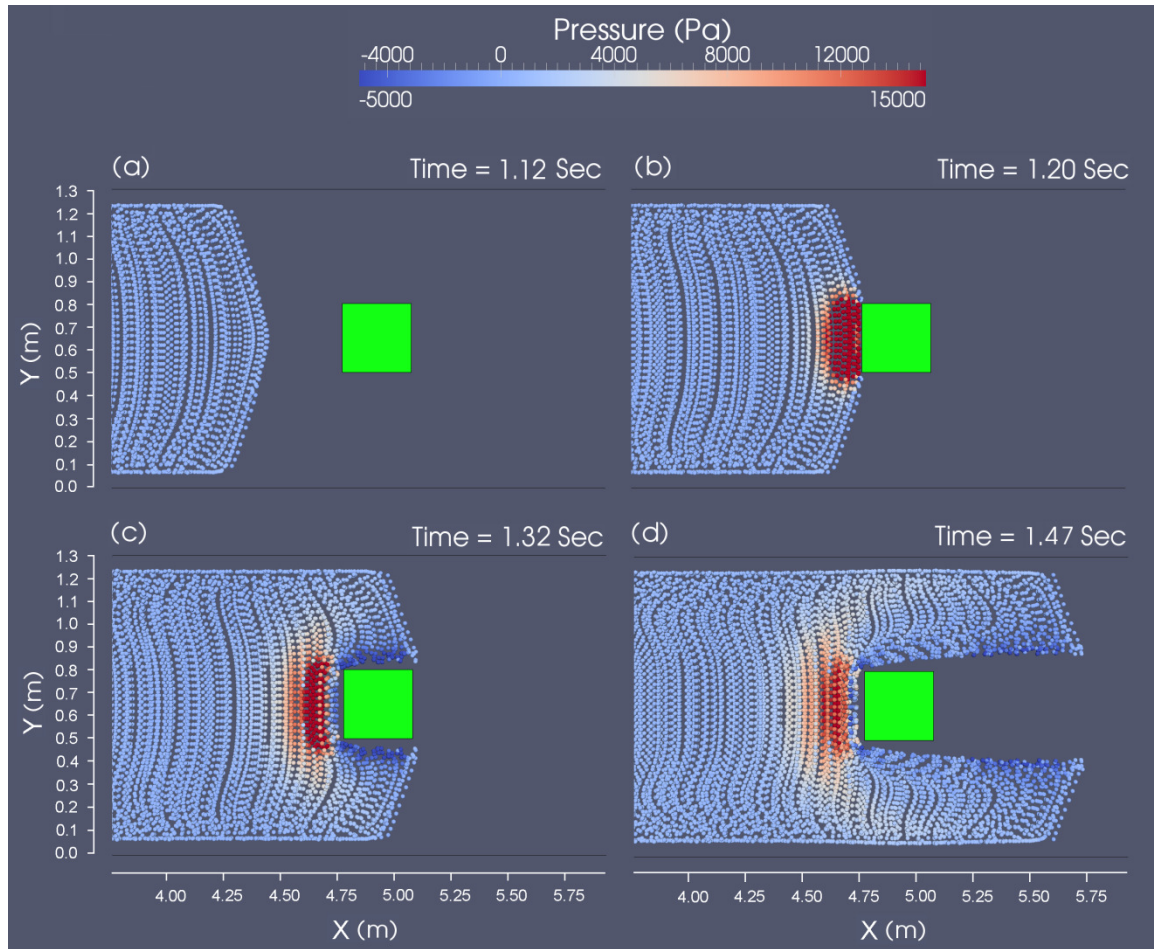


Figure 5.9. Region of unrealistic pressures and void developing beyond initial impact upstream of the square column in a plane close the flume's bottom for a simulation with a higher resolution (simulation D3-HIGH).

accurate hydrostatic pressure distribution acting on the upstream wall of the reservoir before the arrival of the negative dam-break wave.

Since the inaccuracy discussed above would have required further investigation to have it resolved, it was decided to work around it instead. In order to sample particles where the pressure field seemed more realistic, some distance, or *spacing*, was introduced between the outer surface of the sampling volumes and the structure's edge. This can be seen in Figure 5.11-b in comparison to the initial arrangement show in Figure 5.11-a. Since Figure 5.7, Figure 5.8, or Figure 5.9, show that the location of more realistic pressures changes with time – from closer to the column's boundary at impact and to further upstream after – it was required to develop a procedure based on varying the

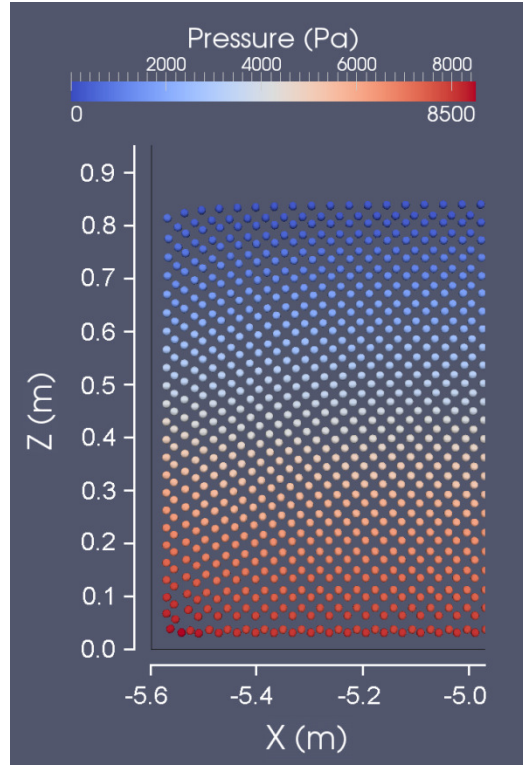


Figure 5.10. Hydrostatic pressure distribution of fluid particles acting on the upstream wall of the water reservoir before the arrival of the negative dam-break wave for simulation D2-85.

spacing between the sampling volumes and the column's boundary. Taking, for example, the determination of the pressure at each transducer used in the physical experiment of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012], the procedure is as follows: (1) Sampling volumes are first aligned vertically with their outer surface adjacent to the column's surface (i.e. no spacing), and with their center elevations adjusted such that they correspond to the elevations of the pressures transducers used in the physical experiments (see Figure 4.5 and Figure 5.11-a). (2) With the sampling volumes initially positioned this way, the pressure time-history at the corresponding transducers is first obtained. (3) Afterwards, spacing is introduced between the sampling volume and the column, and a second time-history of pressures is obtained. (4) The last step is then repeated by gradually increasing the spacing until it becomes considerably large. For example, some of the pressure time-histories obtained with various spacings are shown in Figure 5.12 for simulation D2-85-C. Compared to time-histories obtained using greater spacings, it can

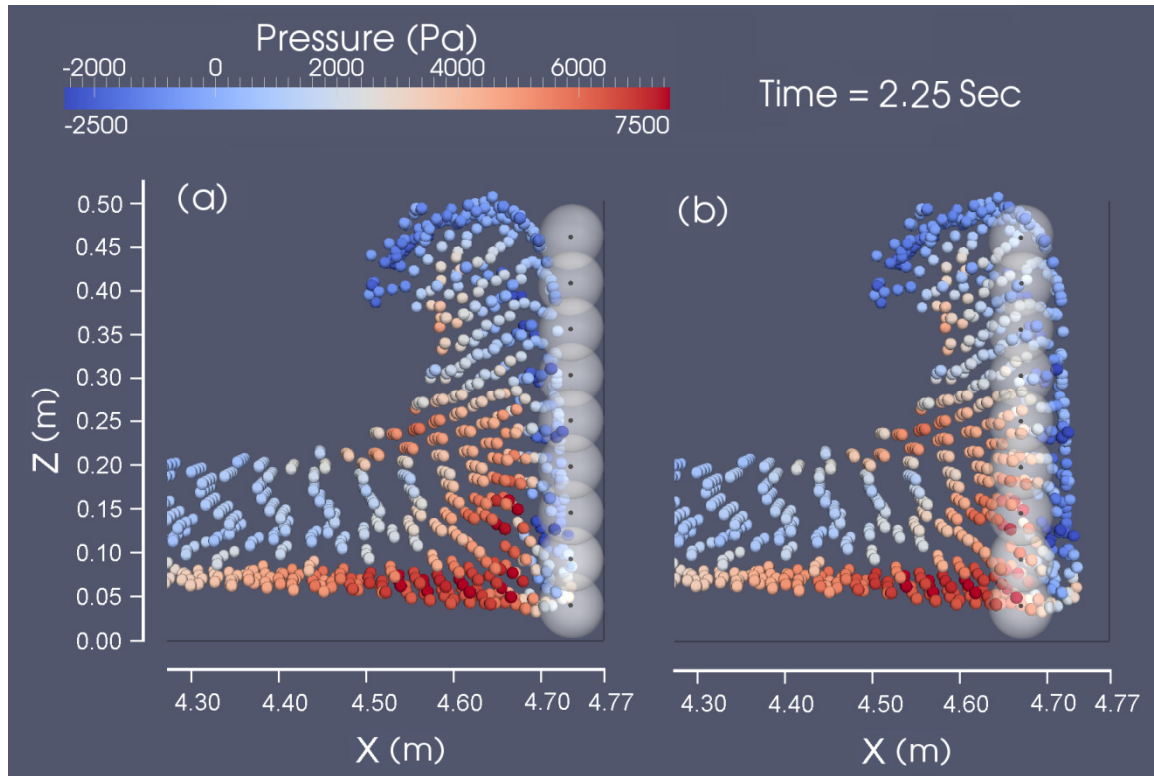


Figure 5.11. Cross-section along the X-Z plane passing through the column's center showing the arrangement of sampling volumes for the determination of the pressure time-history: (a) with no spacing; (b) with a spacing of 0.06m from the column's edge (black lines represent solid boundaries).

be observed that the closer the sampling volumes are to the column's edge, the higher the pressures at initial impact are. However, at this location directly next to the column, the pressures beyond initial impact become unrealistic as they remain very low (Figure 5.12-a). As the spacing increases and as particles are being sampled further upstream of the column, the measured pressures at initial impact become lower while the pressures just after impact begin to develop (Figure 5.12-b). Once a given spacing is reached, the pressures after impact become maximum (Figure 5.12-c), and if the spacing is increased further, all pressures reduce as they are being measured too far away from the column, where the pressure field tends to be more hydrostatic (Figure 5.12-d). Lastly, in order to obtain the overall pressure time-history, as shown in Figure 5.13, the maximum value (either positive or negative) from each of the time-histories obtained with different

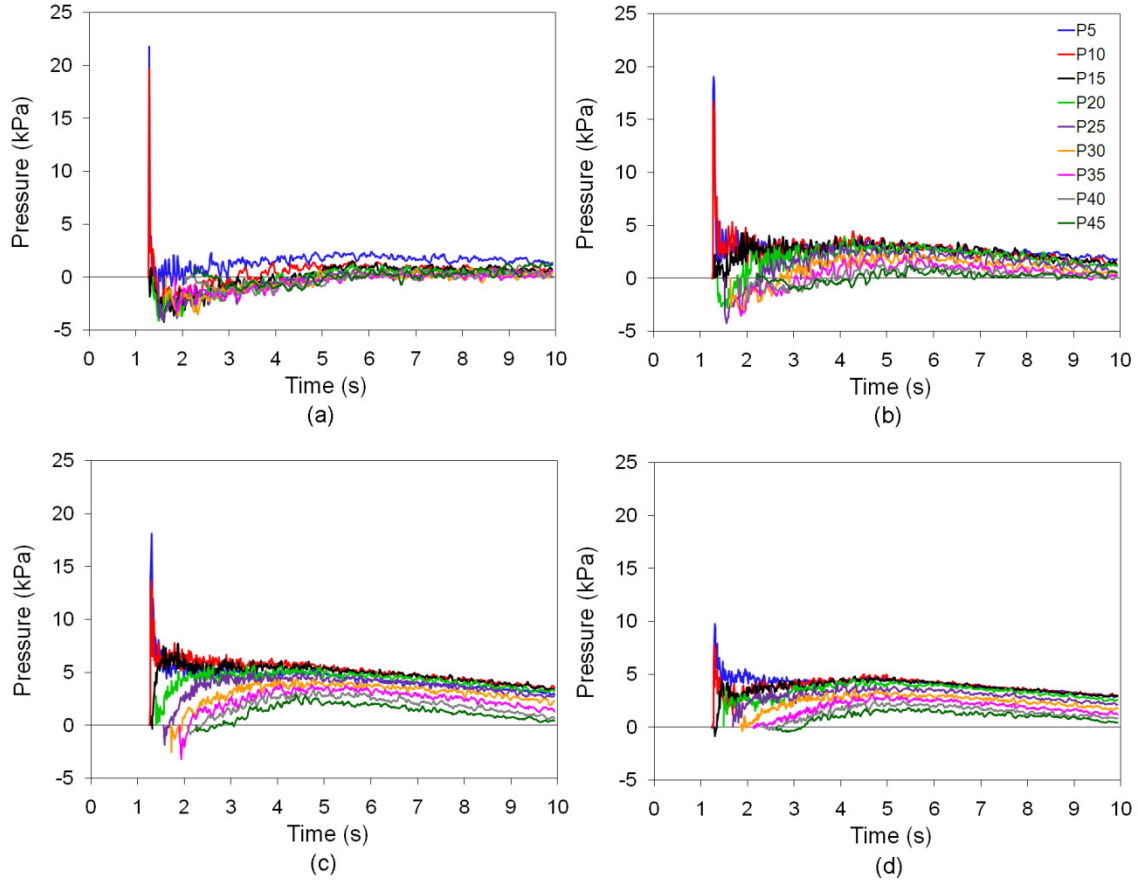


Figure 5.12. Numerical pressure time-histories at transducers P5 to P45 obtained with no spacing (a), and spacings of (b) 0.03m, (c) 0.06m, and (d) 0.12m for simulation D2-85-C.

spacings is selected for all time instants. Note that, only four spacings are shown in Figure 5.12, although the spacing for this example was in fact varied from none to 0.20m, with 0.01m increments. This spacing increment was selected by trial and error based on the initial inter-particle spacing of the simulation.

The results presented in Figure 5.12 and Figure 5.13 were obtained via the kernel approximation (discretized) of the density within spherical sampling volumes of radius $2h$ using Eq. 5.9 followed by the equation of state (Eq. 2.66). As mentioned previously, the speed of sound in water at the reference density (c_0) considered in the post-processing is the same as in the numerical simulation. However, as discussed below, it was noticed that using the same smoothing length (h) did not necessarily produce the best results. Considering a sampling volume whose center is located 0.05m above the flume's bottom

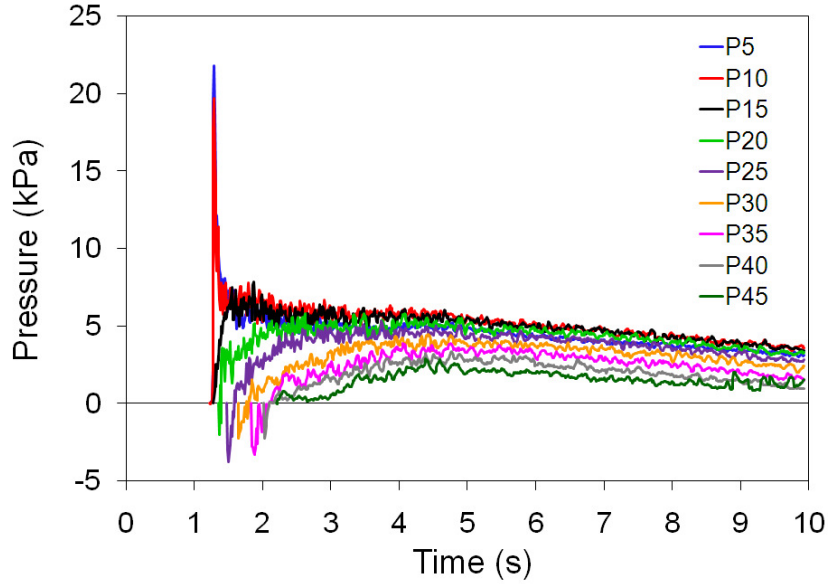


Figure 5.13. Numerical time-history of the pressure at corresponding transducers derived from the absolute maximums of the time-histories obtained with various spacings for simulation D2-85-C.

and 0.06m upstream of the column for simulation D2-85-C (i.e. P5 in Figure 5.12-c), Figure 5.14 shows the time-history of the pressure calculated using various smoothing lengths. It can be observed that when h is too small (i.e. 0.010m), the results are noisier, and it is possible that no water particles are within the sampling volume at a given time instant. This results in a calculated pressure of zero. As h increases, the calculated pressure becomes generally more precise and the spike at initial impact is better captured. Conversely, for h values greater than 0.020m, although the pressure time-history is smoother as more particles are taking part in the kernel approximation, the magnitude of this measurement slightly decreases as the sampling volume expands away from the region of interest (i.e. away from the column's front where the pressures are higher). Note here that the actual smoothing length considered for simulation D2-85-C was 0.044m. Based on the results shown in Figure 5.14, the selected smoothing length in the post-processing of raw numerical outputs was approximately 0.5 of the one utilized in the actual simulations.

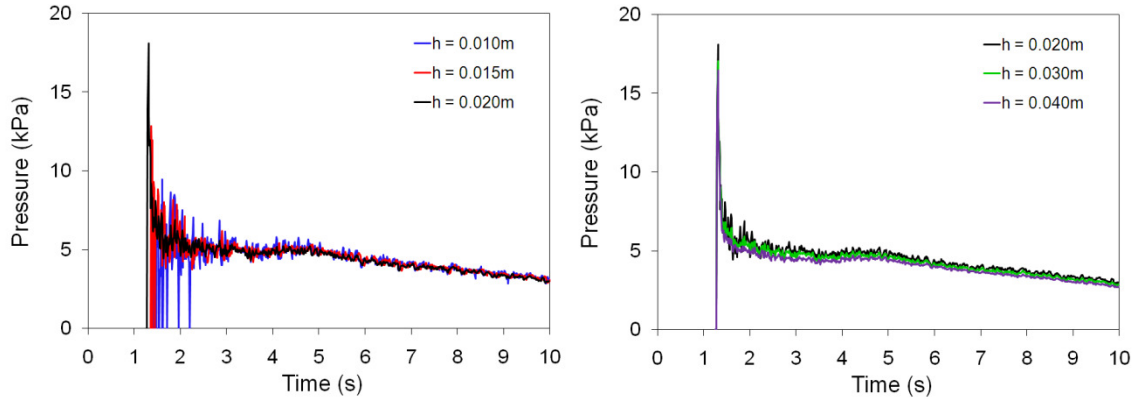


Figure 5.14. Pressure time-histories at transducer P5 for simulation D2-85-C obtain using various smoothing lengths (h).

Determination of Resultant Force

As discussed in Section 5.2.1, the resulting force acting streamwise on the column can be obtained by integrating the pressure exerted on its surface area. As for the determination of the pressure acting on the upstream face of the column discussed in the previous section, it was also required to introduce a spacing between the sampling volumes and the surface of the column structure. As shown in Figure 5.15 in the case of simulation D1, the resulting force on the column will depend on the value of this spacing. It can be seen that the time-history of the force follows the same behaviour as was observed for the time-history of the pressure (Figure 5.12). For a small spacing, the force at impact is the highest while the force after impact is relatively low. This can be explained by the fact that, at impact, the pressures are high at locations closer to the column; however, as the simulation advances, low pressures and a void of particles develop at the column's edge (e.g. Figure 5.7). Alternatively, for a larger spacing, the peak in the force at impact is not detected. This is because, as particles are sampled away from the high impact pressures, the force after impact tends to be higher, since the region of higher more realistic pressures migrates upstream. By taking the maximum at each time instant of the time-histories of the force obtained with gradually increasing spacings, the overall time-history of the net force acting on the column is determined. The numerical and experimental [Yeh and Petroff, n.d.] time-histories of the net force acting on the column for simulation D1 are presented in Figure 5.16. By comparing numerical results with the experimental

results of Yeh and Petroff [n.d.], good agreement can be observed in the magnitude of the force and in the time separating the positive peak (initial impact) and the negative peak (reflected wave). It is important to note that, for representative purposes, the experimental time-history of the force presented in Figure 5.16 was shifted by 0.05 seconds earlier in time. This discrepancy in the time of impact can most likely be attributed to the fact that the 1.0cm layer of water that was present over the bottom of the experimental box (wet-bed condition) was not simulated.

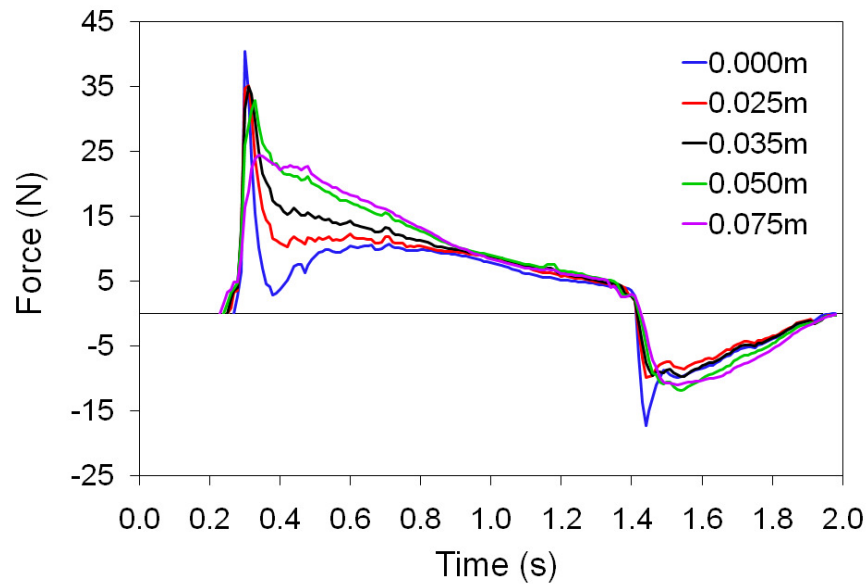


Figure 5.15. Numerical time-history of the resulting force acting on the column for simulation D1 obtained by pressure integration for various distances separating the sampling volumes from the column's surface.

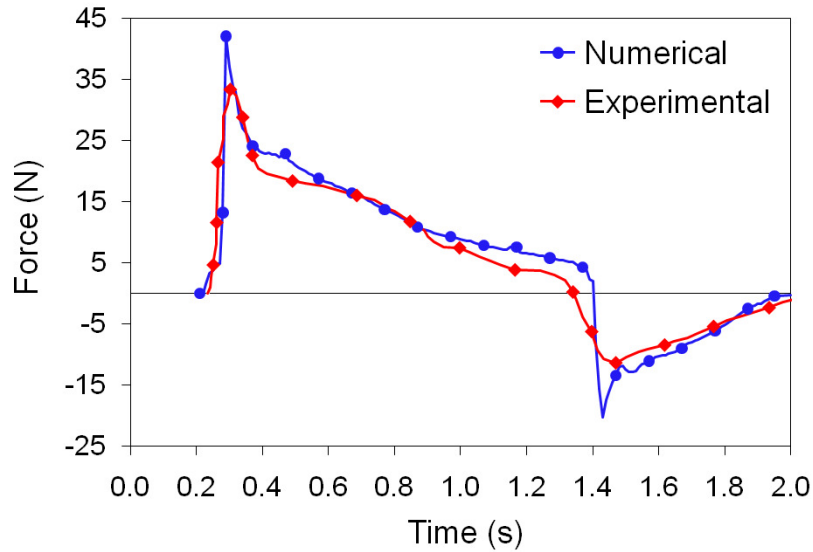


Figure 5.16. Numerical and experimental time-history of the net force acting on a square column for simulation D1.

5.4. Discussion

In this chapter, two different approaches for obtaining specific results from the numerical simulations were presented and evaluated. The first one consisted in the determination of the resulting force acting on an object through the summation of the repulsive forces exerted by the boundary particles that form this object. Based on preliminary attempts, marginal agreement with the experimental data of Yeh and Petroff [n.d.] was obtained. Moreover, it seemed that improving this agreement would have required a high particle resolution, and possibly additional modifications to the formulation of the repulsive boundary condition. On the other hand, determination of the resulting force by pressure integration through the particle sampling approach provided satisfactory results even with a lower particle resolution. Hence, the latter will be utilized throughout the remainder of this thesis in the determination of the resulting force acting on solid objects. Also, this second approach also allowed for determination of the pressure, velocity, and water surface elevation in a relatively straightforward manner.

Lastly, it should be mentioned that, if determination of the force resulting from the impact on columns of floating debris was of interest, then the summation of the repulsive forces exerted by the boundary particles would have been a suitable approach

for estimating the latter. As floating objects are also formed of boundary particles, it would be required to consider the repulsive forces exerted by the boundary particles forming the impacted column on the ones forming the incoming floating object. For more information on the modeling of floating objects within the SPH method in general, the reader is referred to Monaghan [2005] and Gómez-Gesteira *et al.* [2010a].

Chapter 6. Comparison of Numerical with Physical Results

As the core of this thesis, a detailed quantitative comparison between numerical and experimental results is presented in this chapter. Physical experiments were described in Section 4.1. First, the experiments of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] are considered. Subsequently, the single experiment of Kleefsman *et al.* [2005] is also considered not only to further validate the SPH numerical model but also further investigate some discrepancies observed in the first comparison of this chapter.

6.1. Experiments of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012]

This section is sub-divided according to the measurements compared, namely (i) the water surface elevation (h), (ii) the bore-front velocity (U), (iii) the pressure distribution on the upstream face of the column, and (iv) the net base horizontal force acting (streamwise) on the column (F_X). As mentioned in Section 4.1.2, initial leakage through the gate occurred in the physical experiments. Hence, most experimental runs were performed with a downstream water depth $h_d \approx 0.005\text{m}$. However, by allowing sufficient time between runs for the flume's bottom to dry, it was possible to perform runs with a “virtually” dry bed (see Figure 6.14-a). It should also be noted that runs were sometimes performed with a “partially” dry-bed condition (Figure 4.4), and that the bed condition was only assessed in the later stages of the experimental program of Al-Faesly *et al.* [2012]. Hence, throughout this section, if the bed condition (i.e. dry or wet) is known for a particular run, it will be mentioned explicitly. As mentioned in Section 4.3.2, due to some difficulties in reproducing exactly the actual h_d ($\sim 0.005\text{m}$), all SPH results presented in this section correspond to a “dry-bed” condition. Also, as most runs were

performed using the square column model, it will be mentioned explicitly when the circular one is considered.

To investigate further some discrepancies observed between SPH simulations and physical measurements, also included are those numerical results obtained using grid-based FVM along with VOF. These particular results were obtained by Zhihui Ma, graduate student at the University of Ottawa, who has been working in parallel on the same research project using the *Flow-3D* software package.

Due to unknown reasons and confirmed by video analysis, regardless of the measurement (i.e. water surface elevation, pressure, and force), the actual time reference of all experimental time-histories presented herein is unknown. This leads one to surmise that the measuring devices were not properly synchronized with the opening of the gate in the physical experiments. Unfortunately, this required the shifting of all experimental data from an arbitrary amount of time (sometimes up to 3 seconds), making comparison of the impact occurrence time futile. Depending on the measurement, the actual approach used to perform this time shift is explained in the respective section.

Lastly, with the exception of run E-115, only one set of experimental data per run was provided for all the measuring devices (i.e. no duplicates). In the case of run E-115, a second set of experimental data was provided. [The corresponding comparison with numerical results is provided in Appendix D].

6.1.1 Water Surface Elevation

The locations of wave gauges used in the physical experiments are shown in Figure 4.2 and Figure 4.4. The following approach was used to adjust the time reference of the water surface elevation data: For a particular run, the time-history of all wave gauges corresponded to one single time series. The time shift was determined by having the initial segment of the experimental time-history of the water surface elevation at wave gauge W1 (in the reservoir) match the numerical one (see Figure 6.1-a). Afterwards, this time shift was systematically applied to all other wave gauges.

Figure 6.1 shows the numerical and experimental time-history of the water surface elevation at wave gauges W1 to W8 for a run performed without a column structure and with an impoundment depth $h_u = 0.85\text{m}$ (experimental run E-85-N).

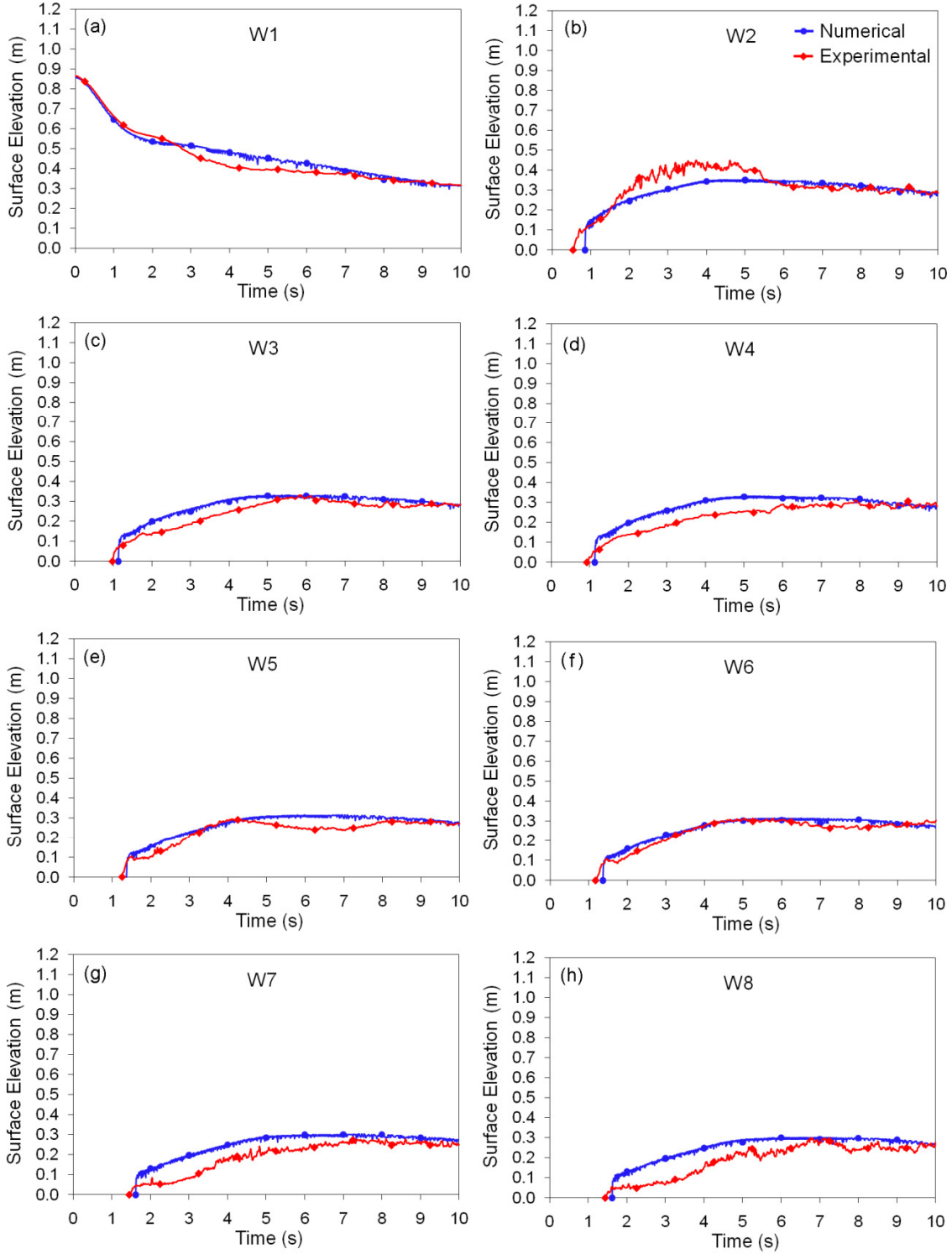


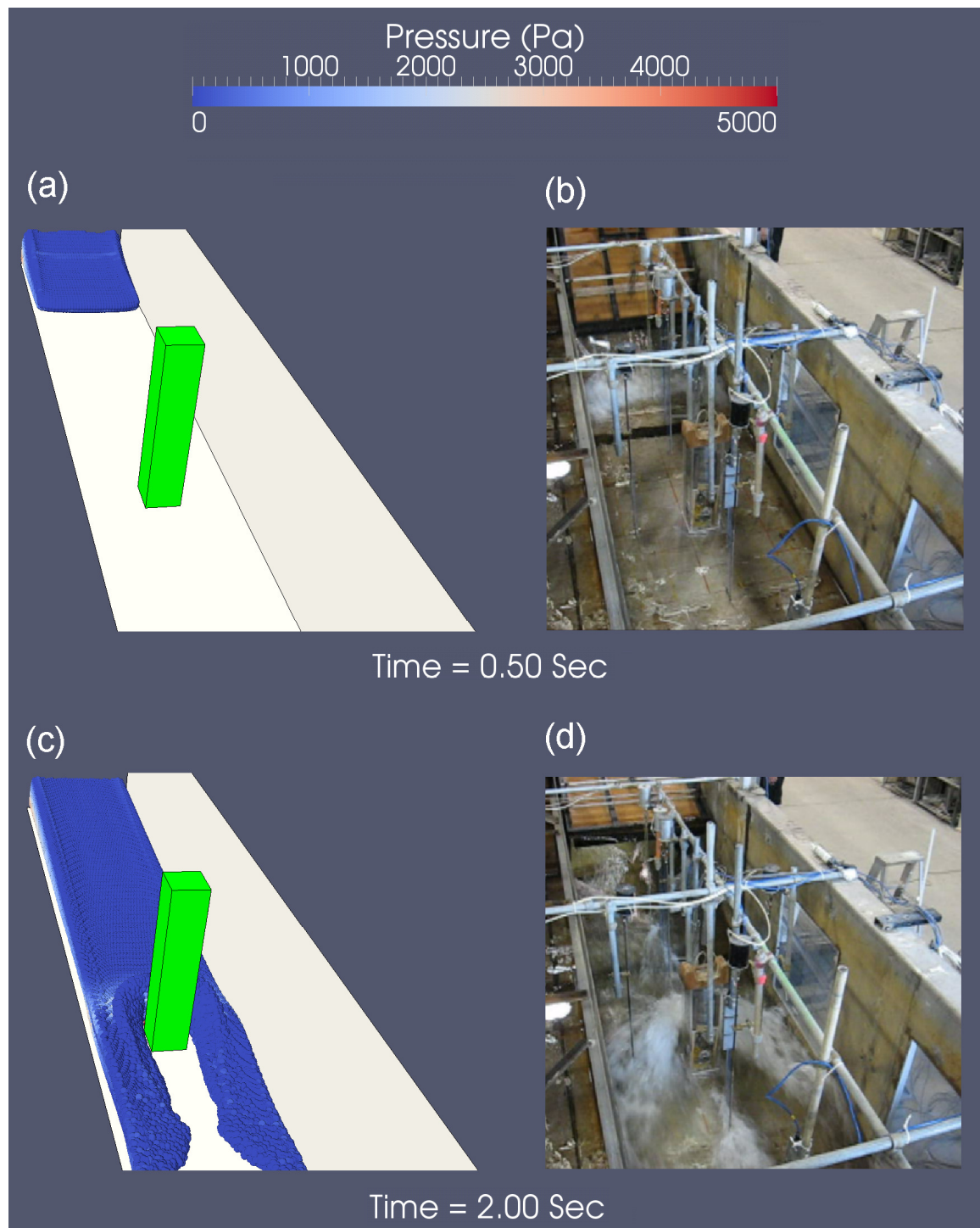
Figure 6.1. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, (d) W4, (e) W5, (f) W6, (g) W7, and (h) W8 for $h_u = 0.85\text{m}$ and without a column (experimental run E-85-N and simulation D2-85-N).

Although generally good agreement can be observed, Figure 6.1-a shows that the experimental time-history at W1 dips momentarily below the numerical one. This discrepancy is also observed for all other comparisons with different h_u . On the other hand, in the case of W2, the experimental time-history momentarily surpasses the one predicted by the numerical model. From video visualization (Figure 6.2), it was noticed that, for all runs, diagonal cross-waves would meet near the location of W2, thereby explaining the latter discrepancy. The formation of these cross-waves observed in the experimental results but not in the numerical ones will be examined in Section 7.1.



Figure 6.2. Cross-waves occurring upstream of the column in the physical experiments (experiment E-85-C).

In the experimental runs that employed the column structure, it should be noted that, with the exception of the run with a circular column (experiment E-85-C), wave gauges W2, W3, and W4 were not utilized as they were believed to cause unwanted flow disturbances. Moreover, as the assessment of the inundation depths next to the column (W9, W11, and W13) required more wave gauges, W5 and W6 were sacrificed in all runs. Before proceeding ahead with a quantitative comparison of the time- history of the water surface elevation, the main flow features of the overall experiment are (visually)



For caption see following page.

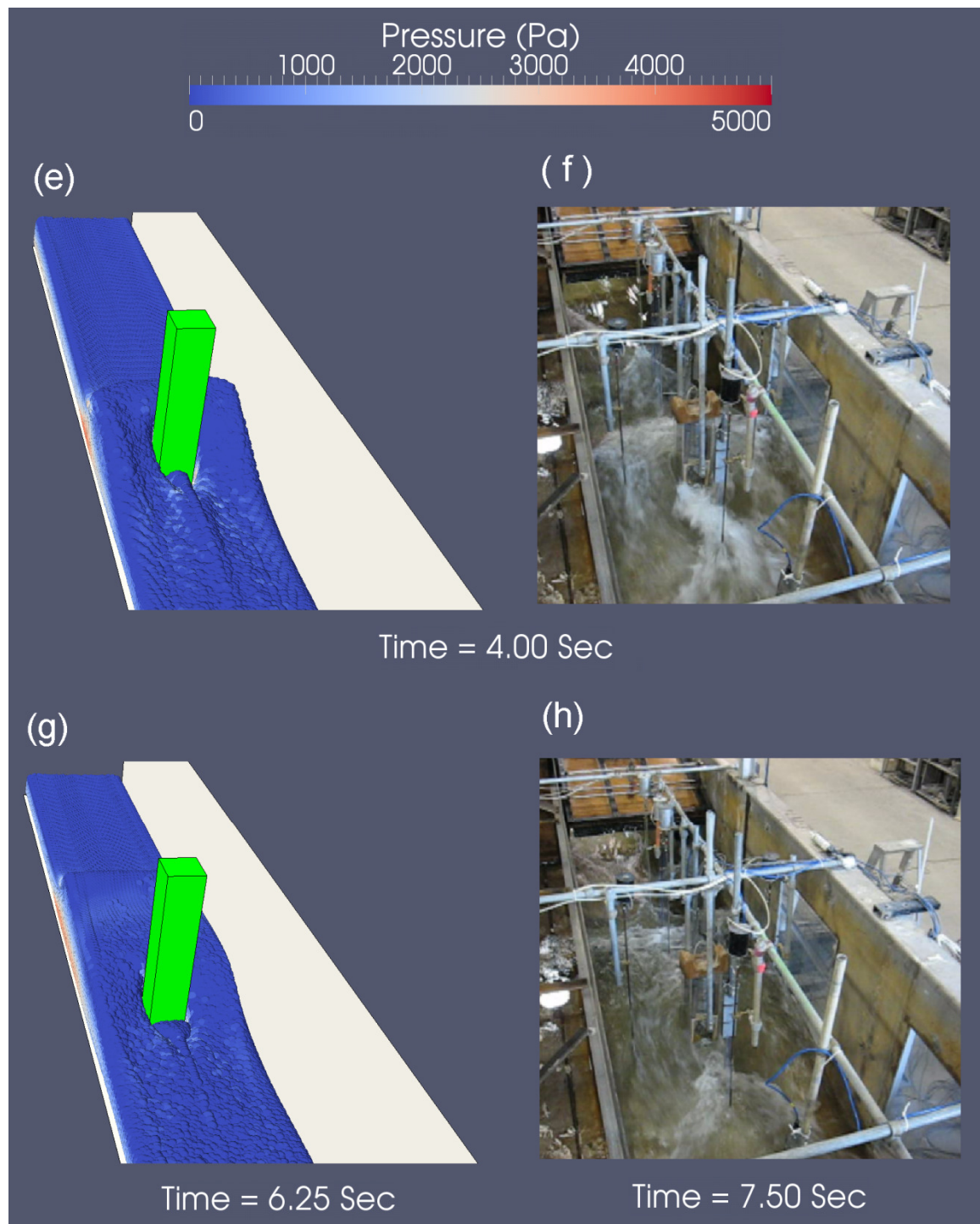


Figure 6.3. Qualitative comparison of the numerical (left, simulation D2-85) and experimental (right, Nouri *et al.* [2010]) water surface (a, b) 0.50, (c, d) 2.00, and (e, f) 4.00 seconds after gate opening ($h_u = 0.85\text{m}$). (g) and (h) shows the out-of-phase upstream propagation of an obstruction-induced wave at 6.25 and 7.50 seconds, respectively.

presented. Figure 6.3 shows experimental still frames along with corresponding ones from numerical simulation D2-85. [It should be noted that the ‘roughened’ water surface (seen in the still frames of the numerical simulation in Figure 6.3) is produced by the post-visualization process. As reconstructing a smooth water surface between adjacent particles would have been cumbersome, in order to make the main flow features more visible, each particle is represented visually by a relatively large sphere]. Moments after impact (Figure 6.3-c and Figure 6.3-d), it can be observed that, in the case of the physical experiment, the flow around the square column rapidly develop into a downstream wake. As the influence of the bed condition is investigated numerically (Section 7.4), it will be demonstrated that the latter plays a role in the formation of this wake. Furthermore, the cross-waves observed in the physical experiment (Figure 6.2) are not present in the numerical simulation (Figure 6.3-c). As time advances, the water level at the front of the column rises and the wake behind the column strengthens, both experimentally and numerically. The fully developed wake in the numerical simulation (Figure 6.3-e) is clearly visible and remains relatively straight. On the other hand, as expected with the vortex shedding resulting from highly turbulent flows around blunt obstacles, the experimental one is highly unstable and translates side to side. Eventually, due to the flow obstruction, or “bottleneck”, caused by the column, a wave propagates back upstream (Figure 6.3-g and Figure 6.3-h). From an upstream perspective (at a location just above the swinging gate looking downstream), the still frame in Figure 6.4 clearly shows this wave as it reaches wave gauge W2 in experimental run E-85-C. As experimental velocity measurements, other than the bore-front velocity, were not performed, video analysis was employed to study the nature of this wave. The video record led one to believe that the wave seen in Figure 6.4 could perhaps be a hydraulic jump. From the observed relatively slower movement of the entrained air bubbles downstream of the wave, the flow velocity at this location seemed subcritical, compared to what appeared to be a supercritical regime upstream. Furthermore, the presence of entrained air bubbles downstream of the wave is characteristic of the high turbulence and mixing induced by a hydraulic jump. The obstruction caused by the column model (and the resulting wave propagating upstream) are further investigated in Section 7.2.

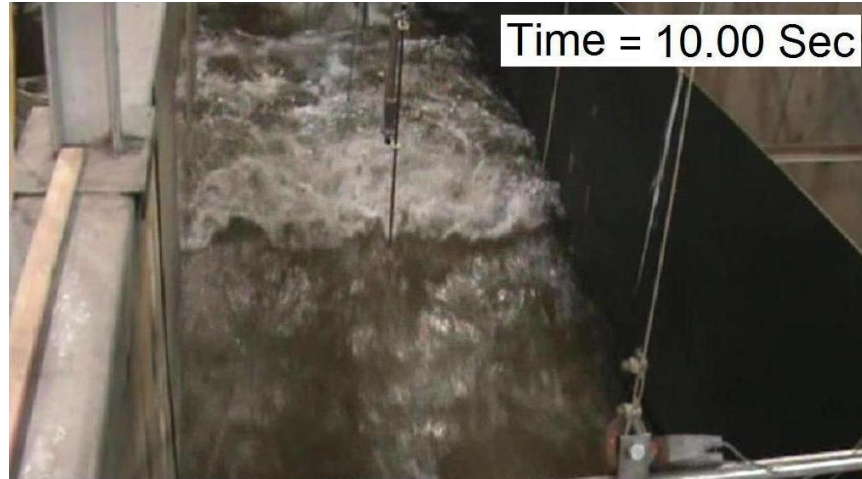


Figure 6.4. Obstruction-induced wave propagating upstream in the physical experiments ($h_u = 0.85\text{m}$, experimental run E-85-C).

Figures 6.5, 6.6, 6.7, and 6.8 show numerical and experimental time-histories of the water surface elevation related to experimental runs E-55-WET, E-85, E-85-C, and E-115, respectively. In all of these figures, the agreement for wave gauges W1, W7, and W8 is shown to be fairly good. However, in the case of gauges W9, W11, and W13, which are located directly at the front, side, and back of the column, respectively (see Figure 4.5), the agreement is marginal.

In the case of gauge W9, although the initial rise in the water surface predicted by the numerical model is much more gradual compared to the one recorded experimentally, some agreement is obtained at later times (e.g. Figure 6.6-d). An explanation for this discrepancy will be provided later in this section, as the bore interaction with the column is visualized frame-by-frame. Also, for the run with $h_u = 1.15\text{m}$ (E-115), the intensity of the splashing at initial impact, along with the high momentum of the flow against the wave gauge, makes the reading of the water surface elevation difficult (Figure 6.8-c). From here on, the water depth at W9 will be referred to as the “runup” of the water on the upstream face of the column.

With the exception of run E-85-C, for which the circular column was considered, the agreement at wave gauge W11 is relatively poor. Based on the aerodynamics of the square column, this disagreement is believed to be due to the significant flow separation from the column’s lateral sides, which makes the wave gauge measurement difficult.

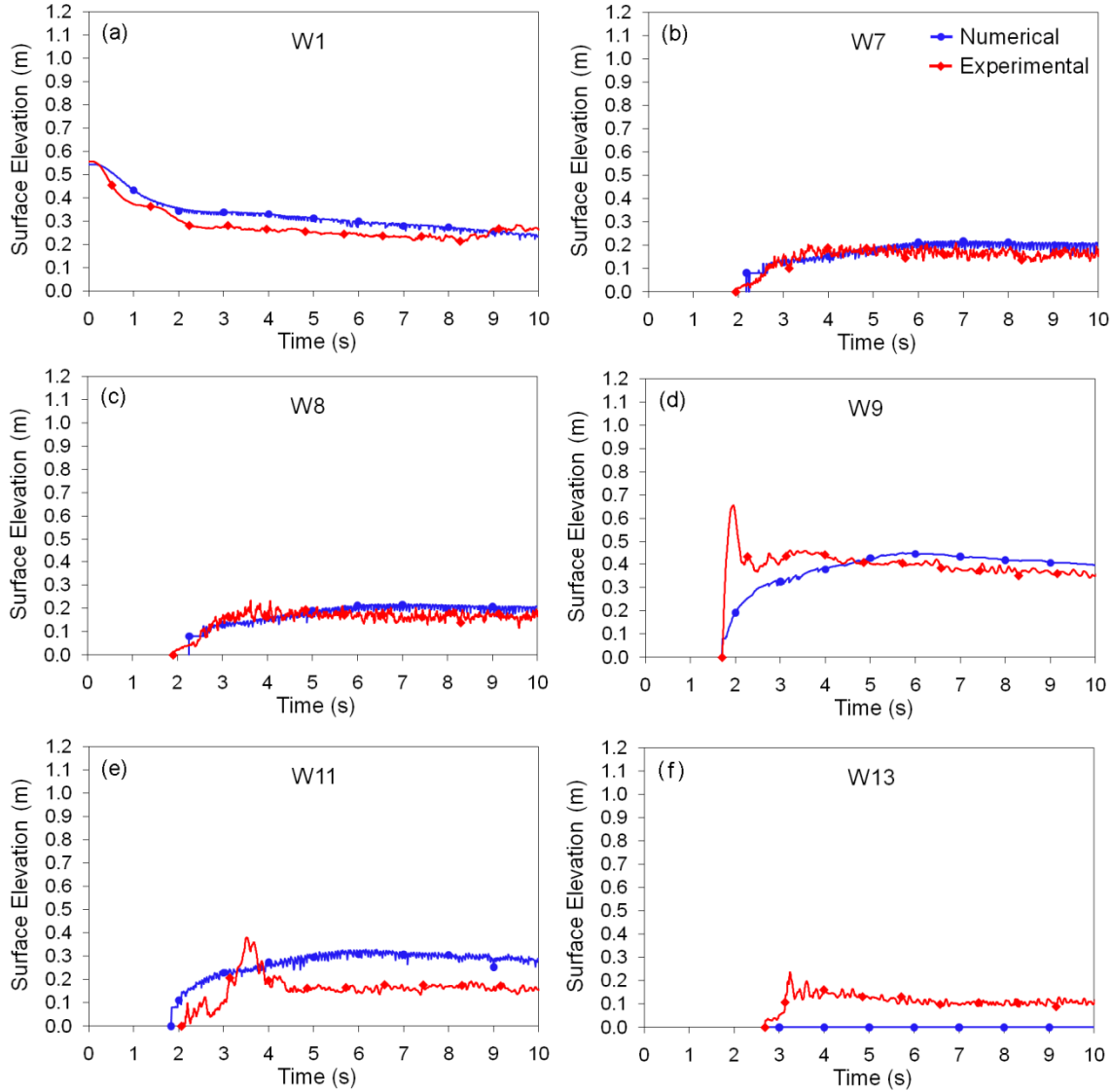


Figure 6.5. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W7, (c) W8, (d) W9, (e) W11, and (f) W13 for $h_u = 0.55\text{m}$ (experimental run E-55-WET and simulation D2-55).

To support this hypothesis and taking for example experimental run E-115 (Figure 6.8-d), it is difficult to believe that, 4 seconds after gate opening, the (middle) flow depth on the side of the column would be 0.04m while it is 0.89m at the front of the column (Figure 6.8-c). Furthermore, considering run E-85-C, with the circular column being more aerodynamic, the reading at W11 is more stable and seems more realistic (Figure 6.7-g). Lastly, the relatively coarse resolution used in the numerical simulation (an initial inter-

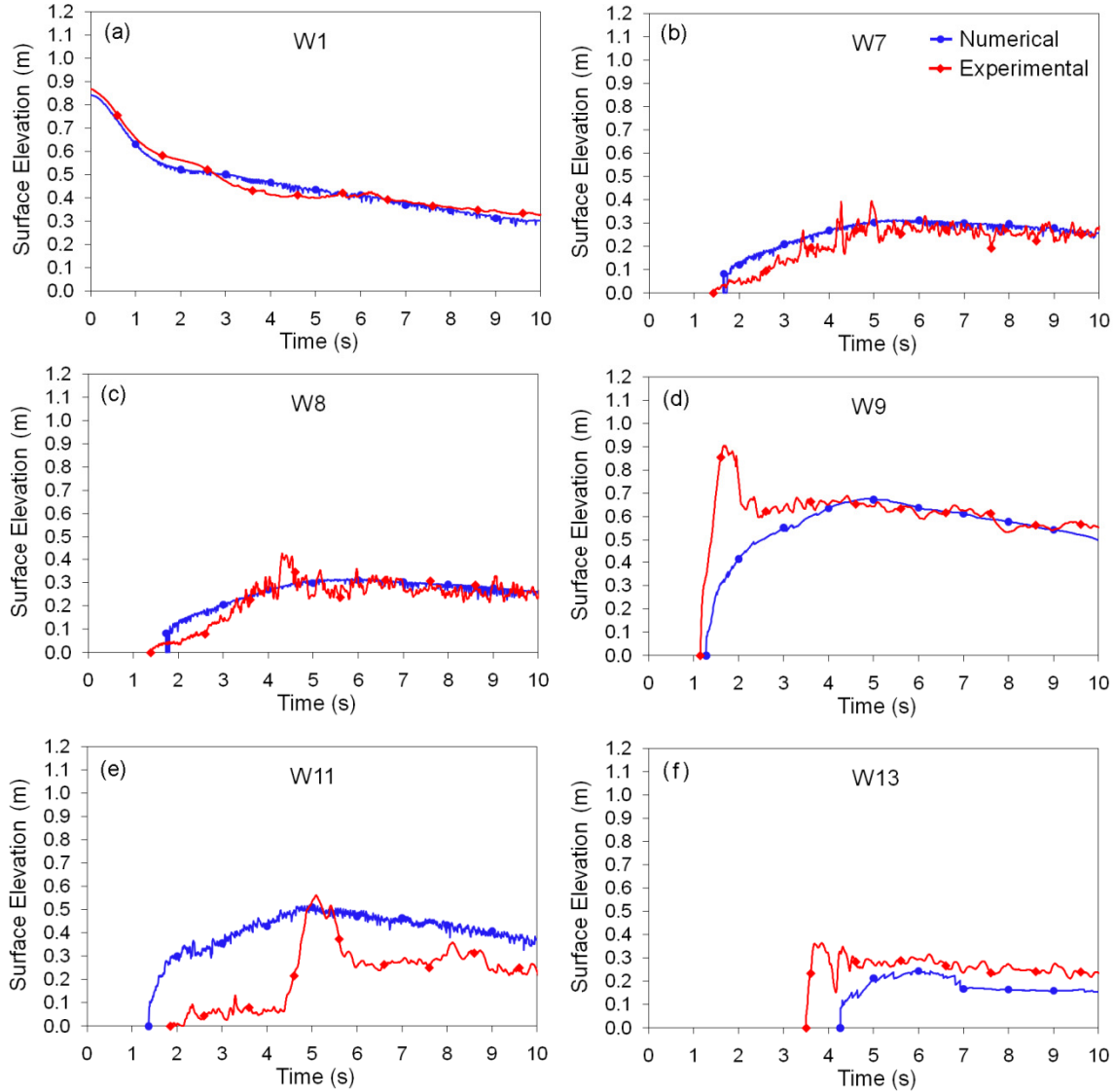


Figure 6.6. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W7, (c) W8, (d) W9, (e) W11, and (f) W13 for $h_u = 0.85\text{m}$ (experimental run E-85 and simulation D2-85).

particle spacing of 0.035m compared to a column side dimension of 0.30m), and the observed inconveniences of the repulsive boundary condition discussed in Section 5.3.2 (i.e. void at solid boundaries), could render the numerical reproduction of the highly turbulent and unstable flow at this precise location (W11) particularly challenging.

In the case of gauge W13, some discrepancy is also observed although it seems to be diminishing as the impoundment depth increases. In the case of numerical simulation

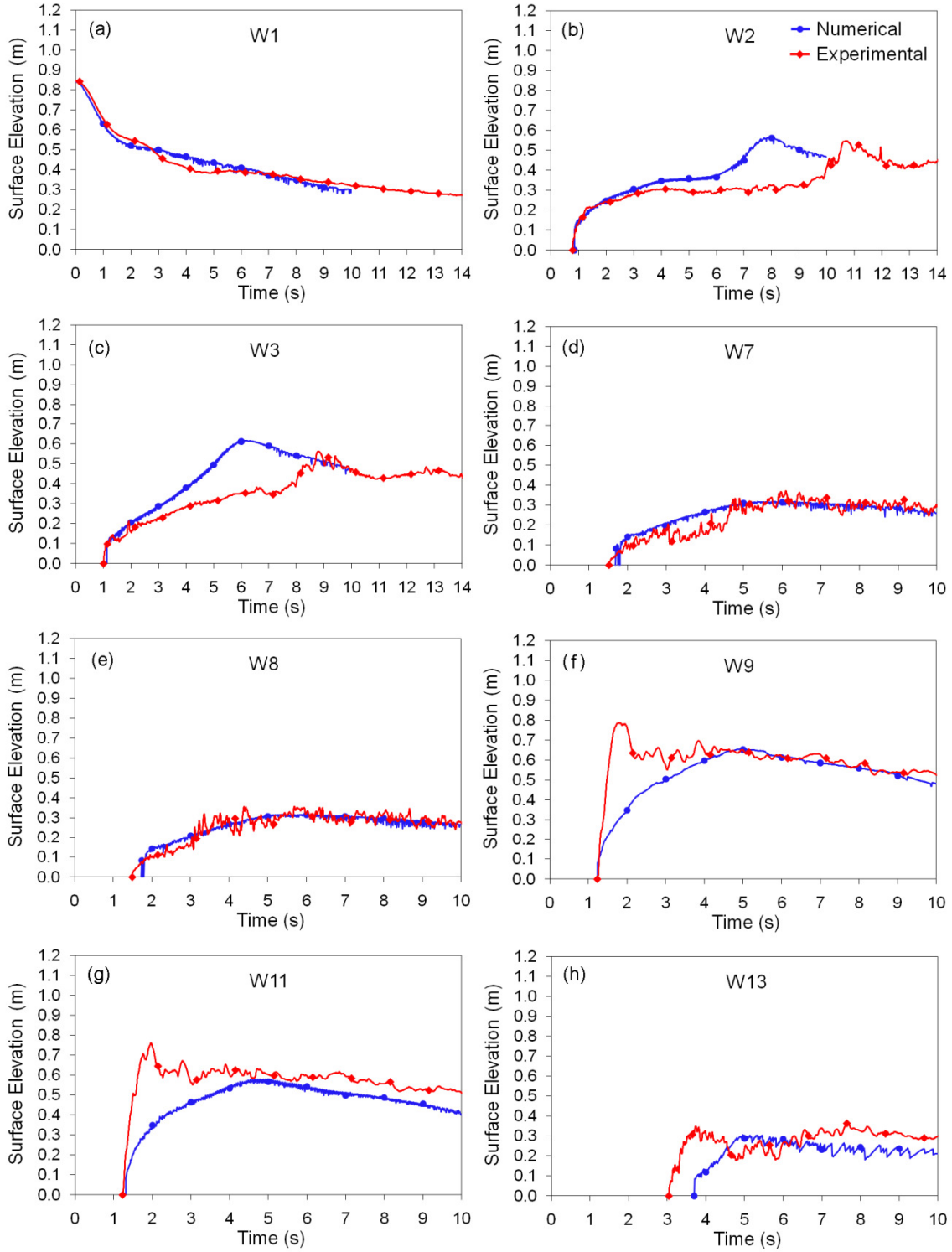


Figure 6.7. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, (d) W7, (e) W8, (f) W9, (g) W11, and (h) W13 for $h_u = 0.85\text{m}$ and with the circular column (experimental run E-85-C and simulation D2-85-C).

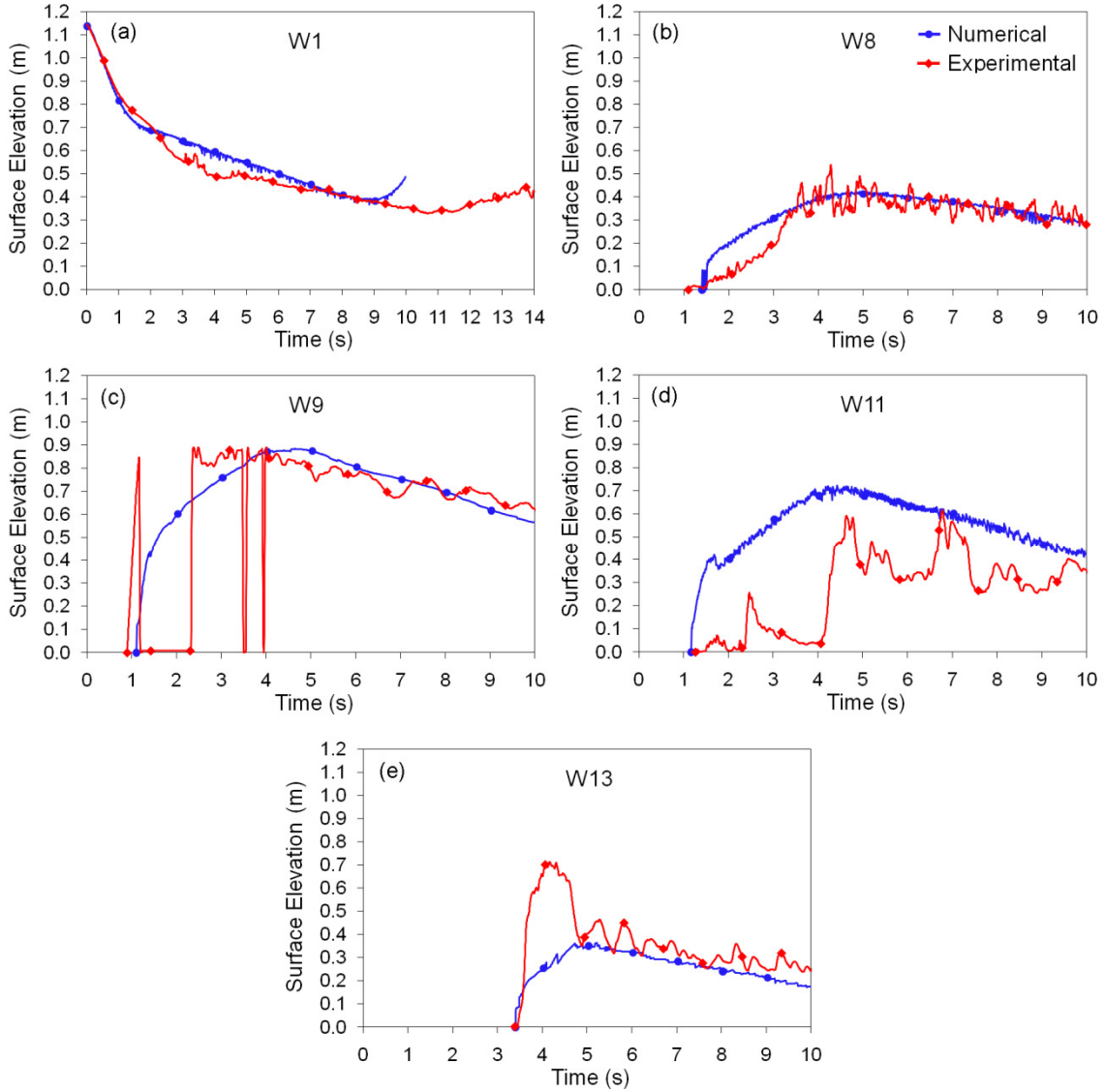


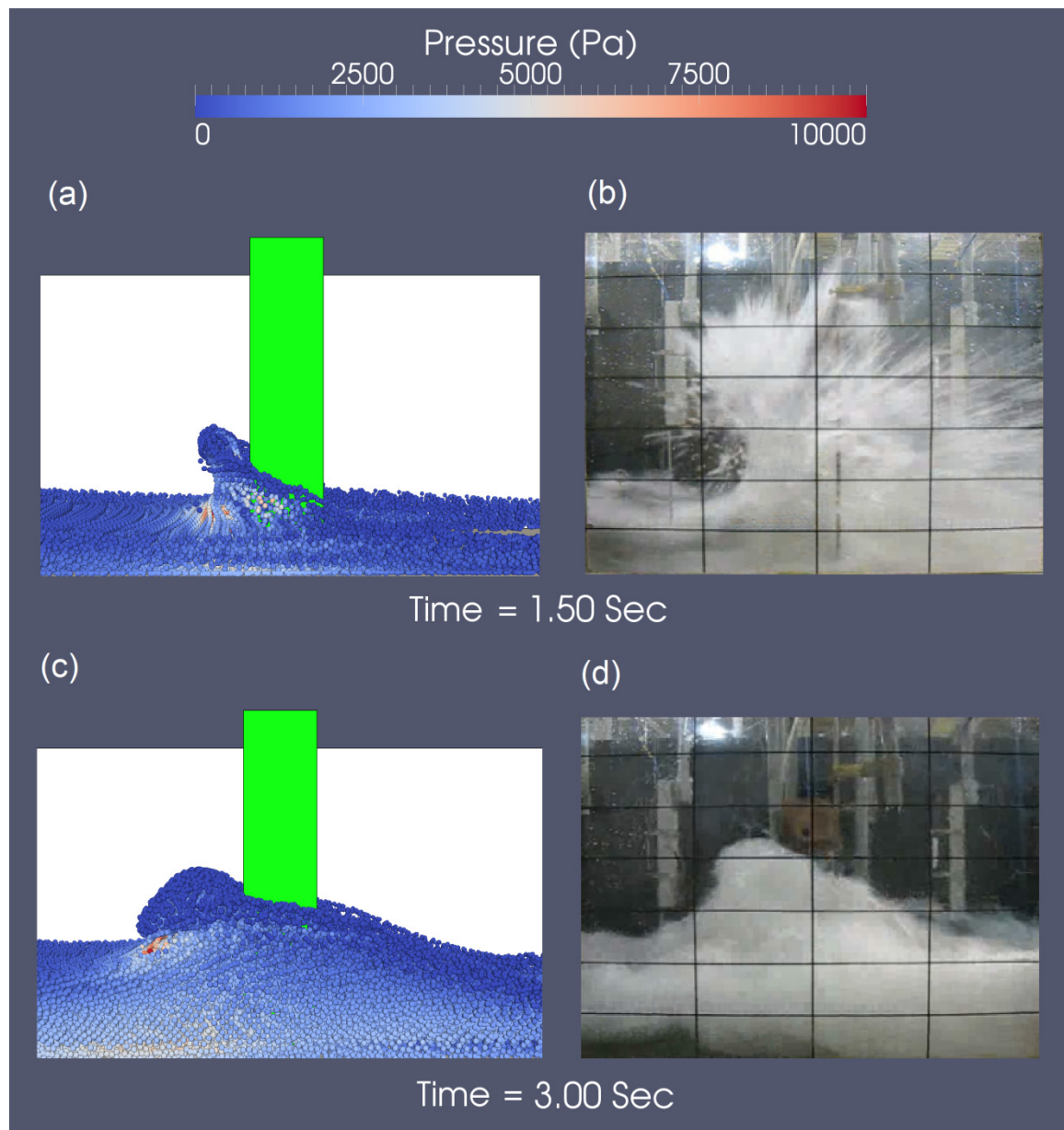
Figure 6.8. Comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W8, (c) W9, (d) W11, and (e) W13 for $h_u = 1.15\text{m}$ (experimental run E-115 and simulation D2-115).

D2-55, the wake never reaches the downstream edge of the column (Figure 6.5-f), while in the case of simulation D2-115, the agreement is relatively good with the exception of the sudden initial rise recorded in the physical experiment not being captured by the numerical model (Figure 6.8-e). Most likely, due to the better aerodynamics of the circular column (experimental run E-85-C), the magnitude-wise agreement between the numerical and experimental results at W13 is relatively good (Figure 6.7-h). However, as

was mentioned previously with $h_u = 0.85\text{m}$ (Figure 6.3-c and Figure 6.3-d), the wake in the numerical simulations generally develops more slowly than was observed experimentally (e.g. Figure 6.6-f).

Considering the circular column run (E-85-C), in which gauges W2 and W3 were used, some time lag is clearly visible in the propagation of the obstruction-induced wave discussed previously. This discrepancy is partially observed for experiment E-115 at gauge W1 (Figure 6.8-a). Considering the peak of the wave (run E-85-C), the actual time lag between the physical experiment and the numerical simulation is 2.81 and 2.71 seconds at gauges W2 and W3, respectively. Furthermore, the average celerity for this wave to travel from W3 to W2 is 1.65 and 1.56m/s, with a decrease in height of 0.015 and 0.055m, in the physical experiment and in the numerical simulation, respectively. Hence, with the wave celerities being approximately the same, this shows that, compared to the numerical wave, the experimental wave takes longer to initiate its propagation. The latter was noted visually when selecting the still frames shown in Figure 6.3-g and Figure 6.3-h. Lastly, the greater decrease in wave height in the numerical simulation could be an indication of excessive diffusion.

On the matter of the discrepancy observed in the runup (i.e. gauge W9) in the earlier stages of the bore-structure interaction (e.g. Figure 6.6-d), a qualitative comparison of still frames is provided in Figure 6.9 (which views the column structure from the side). The run shown was performed with $h_u = 1.00\text{m}$ [Nouri *et al.*, 2010], while the numerical one was performed with $h_u = 1.15\text{m}$ (simulation D2-115). The main discrepancy lies in the fact that the numerical model does not reproduce accurately the significant splash of water that occurs at initial impact (see Figure 6.9-a and Figure 6.9-b). However, later in the simulation (beyond ~3 seconds), the agreement at W9 and in the overall profile of the water surface within the vicinity of the column is fairly good. The lack of splashing at initial impact in the numerical simulation is believed to be due to the relatively coarse particle resolution employed in conjunction with the use of a Riemann solver for removing inaccuracies in the pressure field (see Section 2.3.5). The main outcome of this combination is excessive “numerical viscosity”, which prevents the water particles from reaching higher elevations and separating from one another at the moment of initial impact. Initially, this discrepancy was very confusing, as SPH is known to



For caption see following page.

accurately reproduce significant water surface deformations (e.g. Figure 3.16). It was only after a personal communication with Dr. Benedict Rogers, one of the developers of SPHysics, that the latter explanation was found. Unfortunately, due to circumstances at the time, it was not possible to fix this discrepancy with the experiments of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012], as access to a cluster computer was no longer available for the execution of the parallel version of SPHysics. Hence, as the experiment of Kleefsman *et al.* [2005] was much smaller (and could be simulated using the serial

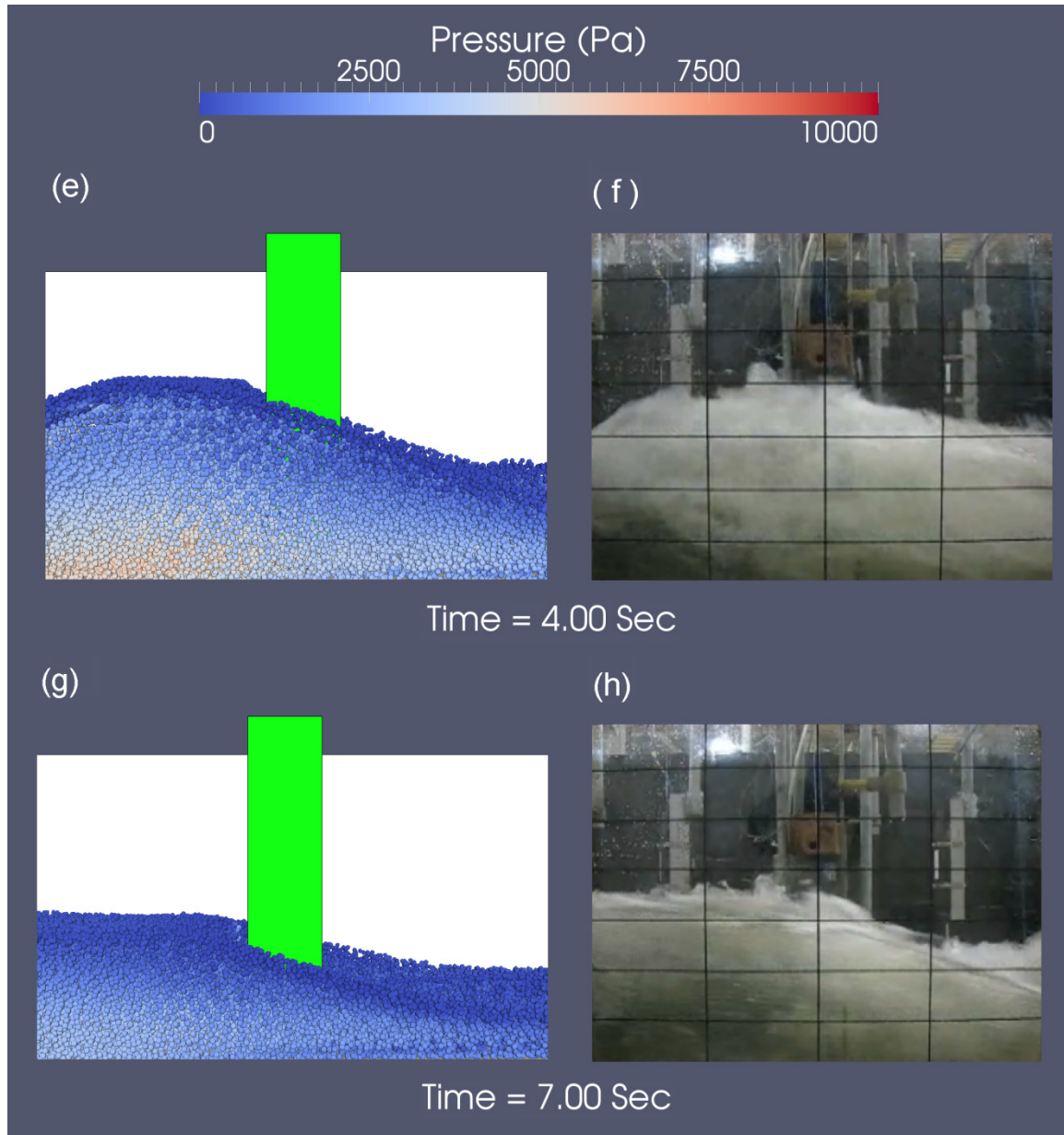
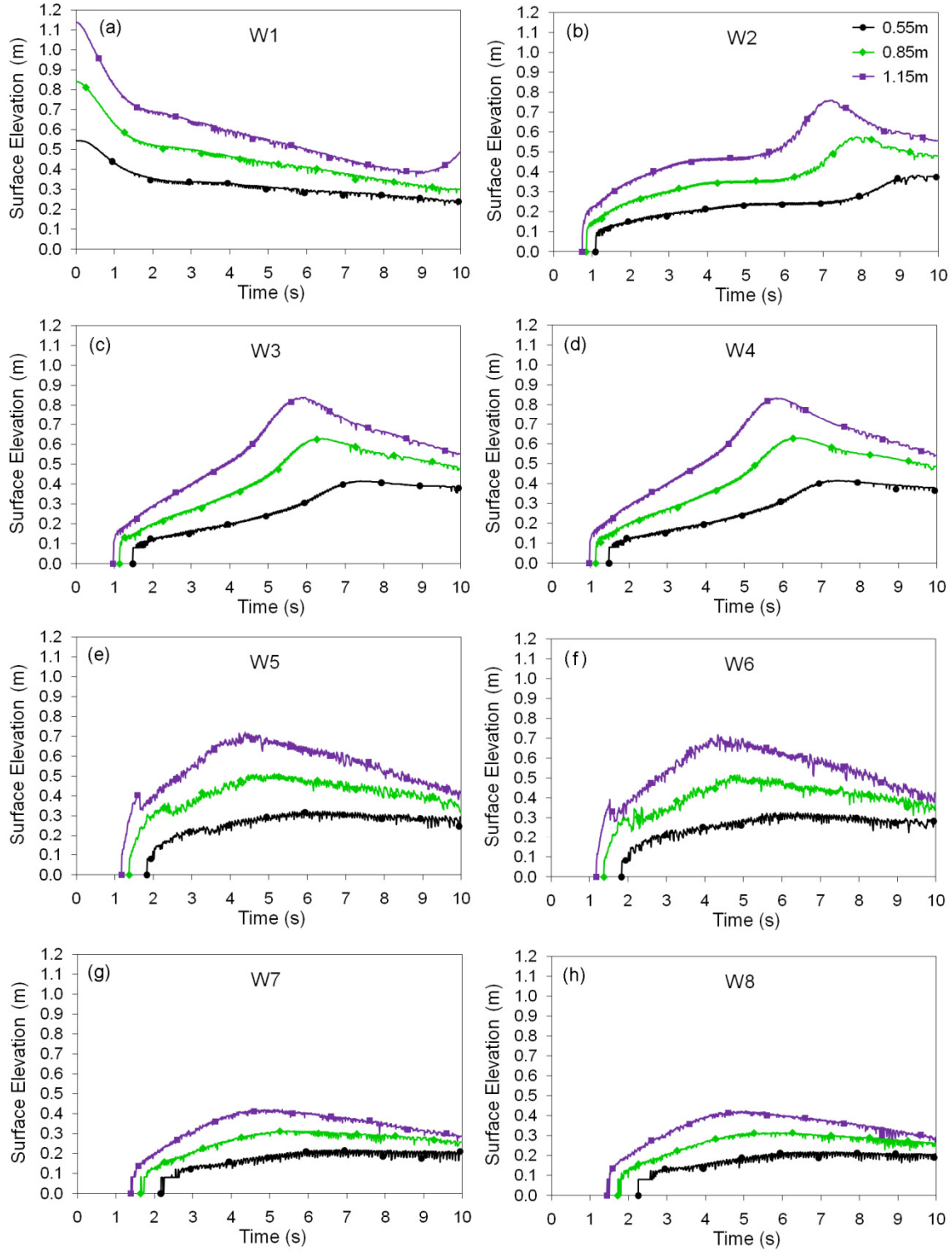


Figure 6.9. Qualitative comparison of numerical (left, simulation D2-115) and experimental (right, Nouri *et al.* [2010]) water surface profiles (a, b) 1.50, (c, d) 3.00, (e, f) 4.00, and (g, h) 7.00 seconds after gate opening. Experimental $h_u = 1.00\text{m}$ and numerical $h_u = 1.15\text{m}$.

version of the numerical model), that experiment was used to investigate the influence of particle resolution on the splashing phenomenon at initial impact when using a Riemann solver. The results of this investigation are presented in the second quantitative comparison of this chapter (Section 6.2).

Figure 6.10 shows the numerical time-history of the water surface elevation at all wave gauges and for all impoundment depths. The numerical results shown correspond to simulations D2-55, D2-85, and D2-115. As expected, the greater the impoundment depth, the stronger the generated hydraulic bore will be (and the deeper the flow will be at all wave gauges). Considering gauge W2, Figure 6.10-b shows that, before the arrival of the aforementioned obstruction-induced wave, the rate at which the water surface elevation increases beyond the passing of the bore-front increases as the impoundment depth increases. Furthermore, the obstruction caused by the column becomes more significant as the impoundment depth increases. As a result, the resulting wave occurs sooner and its height becomes more significant (e.g. Figure 6.10-c).



For caption see following page.

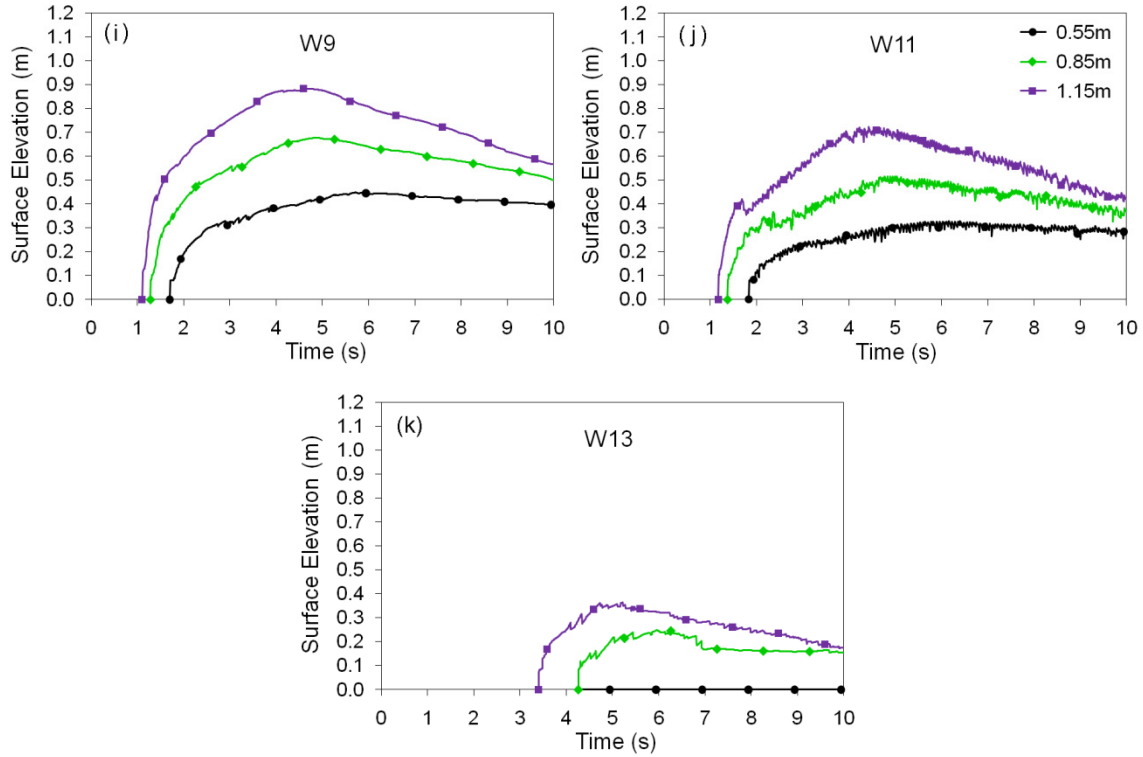


Figure 6.10. Numerical time-history of the water surface elevation at all wave gauges for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).

6.1.2 Bore-Front Velocity

The experimental bore-front velocities considered for comparison were obtained using a high-speed video camera, whose field of view was referenced to a 0.30m square grid marked on the flume's bottom. For experimental runs in which the column model was absent, the bore-front velocity was determined where this one is normally located. Except for E-85-N, these particular experimental runs are not listed in Table 4.1. In the case of the numerical simulations, the bore-front velocity was determined just before impact with the column (slightly upstream). This is because, other than the simulation D2-85-N, associated numerical runs (D2-55 and D2-115) had the column model present. Table 6.1 provides the numerical and experimental bore-front velocities along with the corresponding true relative error, where the experimental value is considered to be the true value in the calculation of this error. With the absolute value of the largest true

Table 6.1. Comparison of numerical and experimental bore-front velocity.

Simulation	h_u (m)	U		ε_t (%)
		Numerical (m/s)	Experimental (m/s)	
D2-55	0.55	3.13	2.99	-4.68
D2-85	0.85	4.45	4.04	-10.15
D2-115	1.15	5.43	5.64	3.72

relative error around 10%, the agreement is arguable. There are many factors that could potentially explain discrepancies.

First, as the video analysis was performed visually by the experimentalist, it is plausible that measuring errors occurred. For instance, Figure 6.11 shows still frames (plan view) of the bore-front propagating over the 0.30m square grid marked on the flume's bottom ($h_u = 0.85\text{m}$). As the shape of the bore-front is irregular, the exact arrival of the bore-front at each of the white transversal grid lines is open to question.

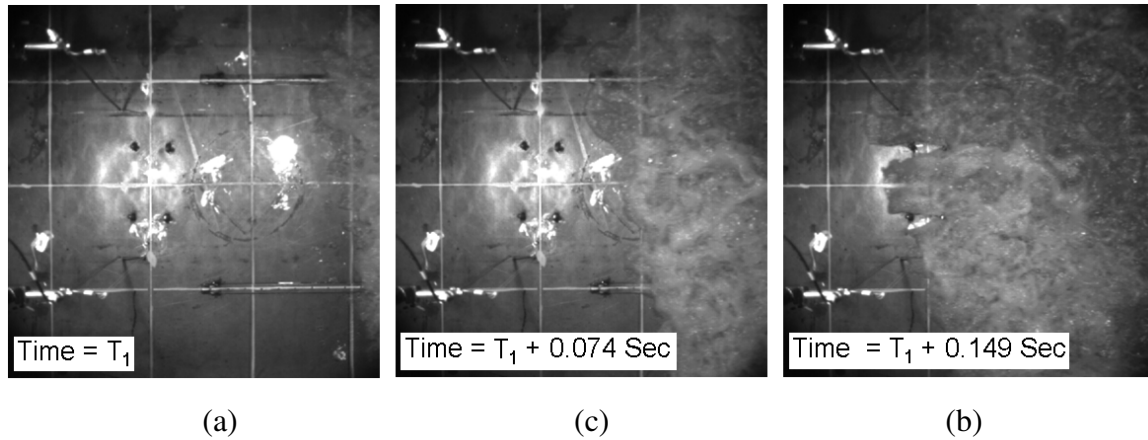


Figure 6.11. Experimental still frames (plan view) of the bore-front at the location of the column ($h_u = 0.85\text{m}$ and $h_d \approx 0.005\text{m}$, time T_1 is unknown).

Secondly, it is unknown how many experimental runs with the same impoundment depth (i.e. duplicates) were considered in the determination of the corresponding bore-front velocity. As the experiments were performed at a relatively

large scale, variations in the actual bore-front velocity, even for identical experimental runs, are likely to occur.

Lastly, the bed condition (i.e. wet, “partially” dry, or “virtually” dry) was not assessed for the experimental values presented in Table 6.1. As discussed in the literature review (e.g. Figure 3.4), the bed condition can significantly affect the bore-front velocity. To support this, using still images taken by the high-speed video camera once again, Figure 6.14 shows the propagation of a bore ($h_d = 0.55\text{m}$) over a “virtually” dry bed before collision with a mitigation wall (topic not covered in this thesis). Even over a short distance, it is clear that the bore-front segment propagating over the dry portion of the flume’s bottom advances more rapidly. Therefore, as the numerical simulations considered in this chapter were all preformed with a perfectly dry bed, this last factor could explain why the numerical model generally over predicts the bore-front velocity.

On the other hand, as selection of the SPH-related parameters was not based on a thorough comparison with experimental results (see Section 4.2), it is possible that better agreement would have been obtained if a more conventional calibration had been performed.

6.1.3 Pressure

Comparisons of the numerical and experimental time-history of the vertical pressure distribution exerted on the upstream face of the column ($h_u = 0.55\text{m}$ and for both dry- and wet-bed conditions), are presented in Figure 6.12 and Figure 6.13, respectively. With the exception of the spikes at initial impact, the agreement is quite good. However, at elevations higher than 0.35m (e.g. Figure 6.13-g), the numerical model overestimates pressures. This is a consequence of the runup (i.e. water surface elevation at gauge W9) being slightly greater in the numerical simulation beyond ~5 seconds (see Figure 6.5-d).

Shock pressures, characterized by high magnitude and short duration, are clearly visible (Figure 6.12) at initial impact of the hydraulic bore with the column. This is for both the numerical and experimental results, as both run E-55-DRY and simulation D2-55 correspond to a dry-bed condition. Discrepancies reside in the magnitude of the pressure spikes being overestimated by the numerical model at pressure transducers P5 and P10, and the experimental spikes at P15 and P20 not being captured by the latter. On

Chapter 6 Comparison of Numerical with Physical Results - Experiments of Nouri et al. [2010] and Al-Faesly et al. [2012]

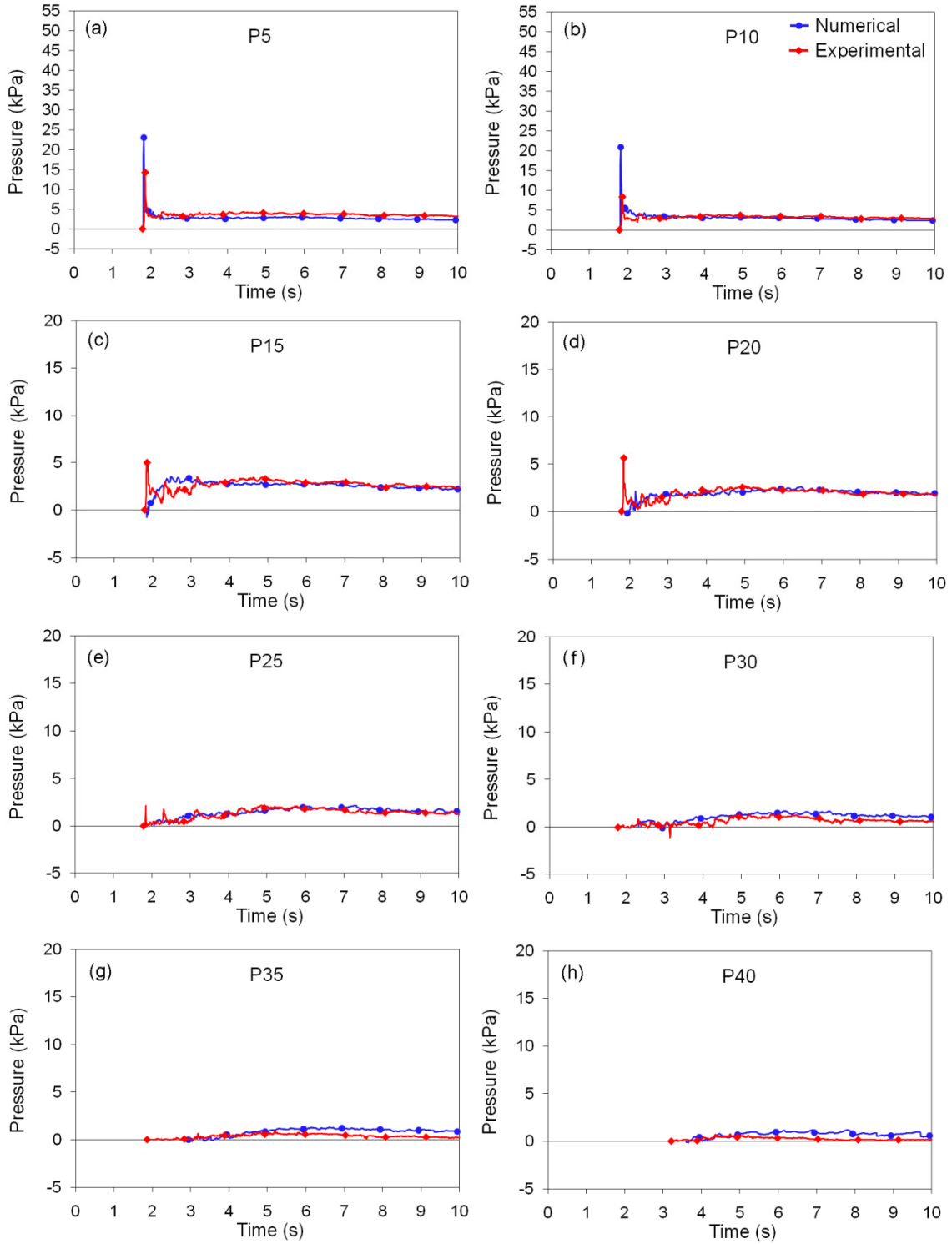


Figure 6.12. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.55\text{m}$ and experimental $h_d \approx 0$ (experimental run E-55-DRY and simulation D2-55).

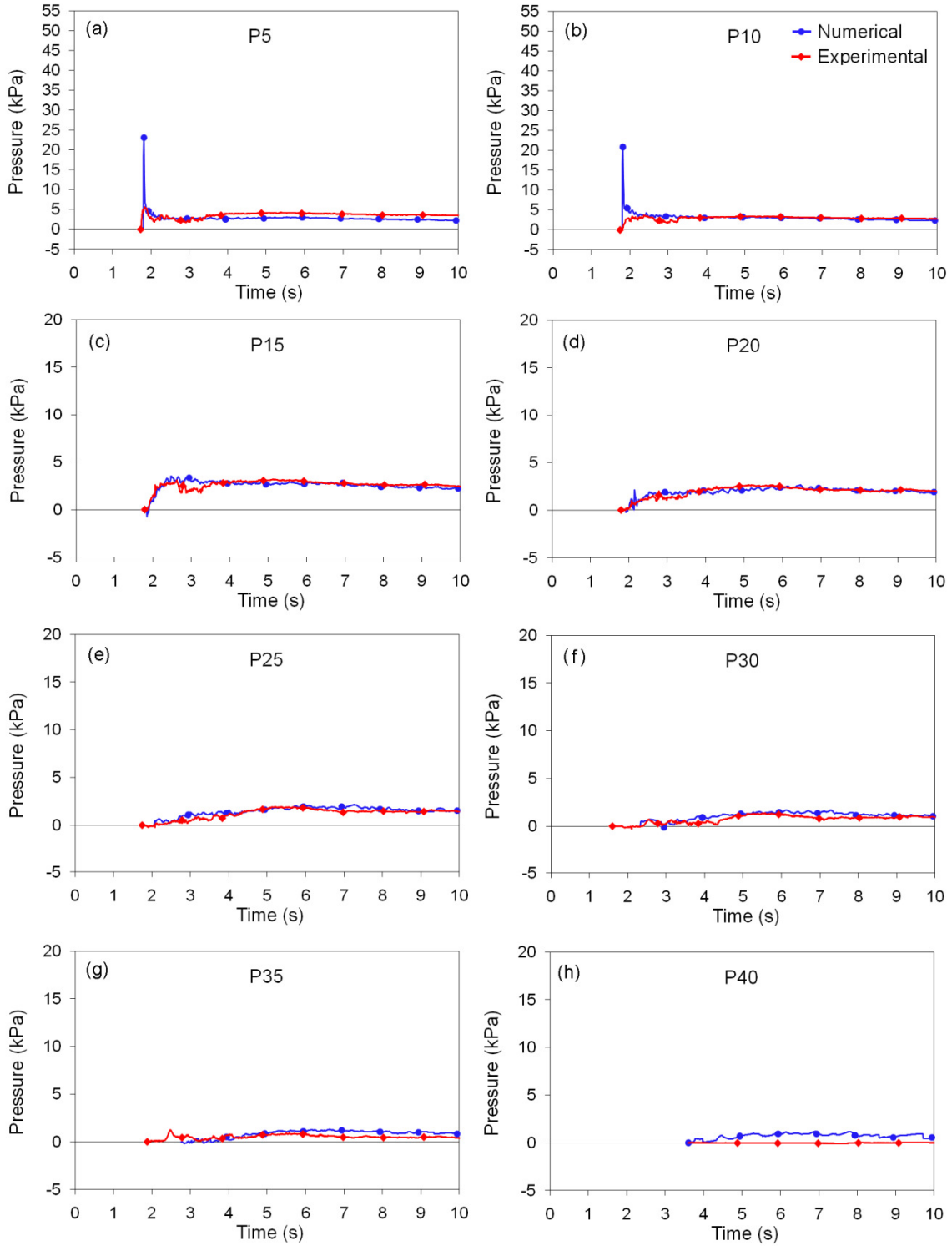


Figure 6.13. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.55\text{m}$ and experimental $h_d \approx 0.005\text{m}$ (experiment E-55-WET and simulation D2-55).

the other hand, as was mentioned in Section 3.3, no shock pressures were observed in the physical experiment performed with a wet-bed condition (Figure 6.13). In an experimental study focusing on the impact of ocean waves breaking directly on vertical walls, Hattori *et al.* [1994] found that the larger the amount of air entrapped in the impinging wave, the lower the magnitude and the longer the rise, or compression time, of the impact pressures. This “cushioning” provided by the entrapped air and its effect on impact pressures was also noticed by Peregrine [2003], who also investigated experimentally the impact of breaking waves on vertical walls. Hence, the believed explanation for the absence of shock pressures at initial impact in the case of physical experiments performed with a wet bed (e.g. E-55-WET) is as follows: As the hydraulic bore advances in the test channel, the still water layer of depth h_d its propagation. This flow resistance and the resulting decrease in the bore-front velocity are well known in the literature (Figure 3.4), even for an h_d of a few millimetres. From Figure 3.8 ($h_d \approx 0.002\text{m}$), in contrast to Figure 3.7 ($h_d = 0$), the momentum transfer from the advancing bore-front to the downstream still water layer seems to generate turbulence which results in considerable air entrainment. It would be this higher concentration of entrapped air in the bore-front that inhibits shock pressures at initial impact. In fact, by comparing still images from the current physical experiments captured with the high-speed video camera (Figure 6.14 and Figure 6.15), this increase in the concentration of entrapped air within the bore-front can be noticed. [The runs shown in these figures were performed with a mitigation wall and such a protection measure is not considered in the current study].

Although both the numerical and experimental runs depicted in Figure 6.12 correspond to a dry-bed condition, there is some discrepancy in the magnitude of the impact pressure at transducer P5. The most probable explanation for this is the fact that, although the physical experiment was performed with a “virtually” dry-bed, due to the friction with the flume’s dry bottom, some air entrainment is still likely to occur. However, the single-phase SPH numerical model does not simulate any air entrainment. Two other possible reasons are the following: Bagnold [1939], who studied wave pressures on coastal structures, pointed out that the magnitude of pressure peaks varies extensively from impact to impact, even when waves are identical. Based on this,

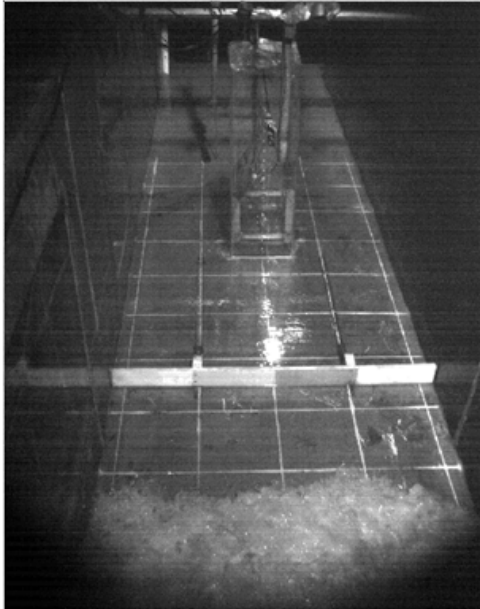


(a)



(b)

Figure 6.14. Experimental still frames of the bore-front propagating over a “virtually” dry bed ($h_d \approx 0$) captured with the high-speed video camera ($h_u = 0.55\text{m}$).



(a)



(b)

Figure 6.15. Experimental still frames of the bore-front propagating over a wet bed ($h_d \approx 0.005\text{m}$) captured with the high-speed video camera ($h_u = 0.55\text{m}$).

Kirkgöz [1991] considered a log-normal probability distribution for the interpretation of the shock pressures resulting from the impact of breaking waves with walls inclined at various angles. Therefore, as only one set of experimental data was provided herein for an experimental run performed with a dry bed (E-55-DRY), the degree of variation in the impact pressures cannot be assessed with confidence. Secondly, with the exception of run E-85-C, the sampling frequency of the pressure transducer was 200Hz. Although further tests would be required to thoroughly demonstrate this, there is a possibility that this frequency might be insufficient to properly capture very short impulses in the pressure time-history. For example, in the case of numerical simulation D2-85, it was noticed that increasing the frequency of numerical outputs from 100Hz to 1000Hz revealed further detail of the shock pressure at initial impact. Considering only a short time interval covering initial impact, Figure 6.16 shows that the time-history for transducer P5 obtained at a frequency of 100Hz matches, but does not fully capture the initial spike and its subsequent oscillation seen in the time-history obtained at 1000Hz. Therefore, it is possible that higher pressure peaks occurred in the physical experiments but these were not captured by the pressure transducers.

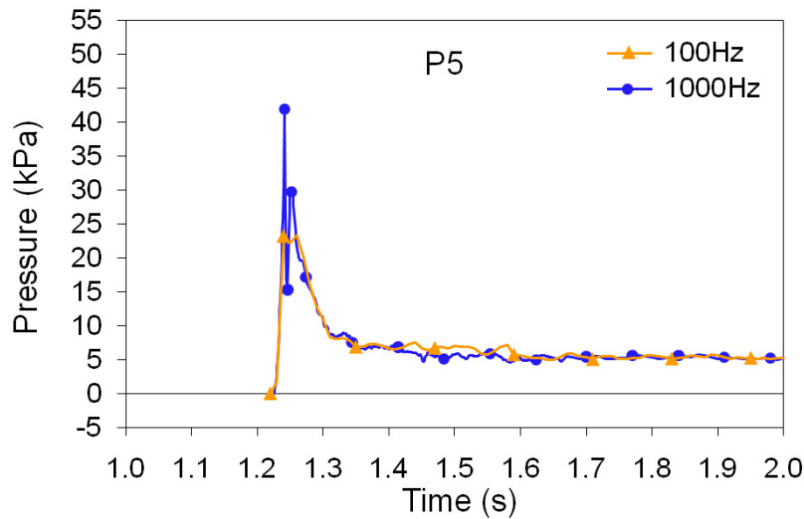


Figure 6.16. Comparison of numerical time-history of pressures acting on the column at an elevation of 0.05m above the flume's bottom with different sampling frequencies for $h_u = 0.85\text{m}$ (simulation D2-85).

Considering the physical experiments for which the bed condition was not assessed, Figures 6.17, 6.18, and 6.19 show the comparison of the numerical and experimental time-history of pressure for E-85, E85-C, and E-115, respectively. Once again, the main discrepancy rests with the prediction of pressure at initial impact. This, in turn, is believed to be related to the bed condition in the physical experiments, which is most likely different from being “perfectly” dry as in the case of the numerical simulations. Interestingly, as seen in Figure 6.19-g, the numerical model is capable of capturing the development of negative pressures exerted at higher elevations on the column occurring at initial impact. These negative pressures can be attributed to water being reflected upstream, along with the suction created as it separates from the column’s surface. Nevertheless, the numerical model generally over-predicts these negative pressures (e.g. Figure 6.19-e). Furthermore, in the case of the circular column (Figure 6.18), such pressures are not experimentally measured. It is foreseeable that, due to the better aerodynamics of a circular column, the upstream flow reflection, which induces negative pressures, is less important. With this in mind, it is plausible that the numerical prediction of the negative pressures is affected by the unwanted formation of the numerical layer of unrealistic pressures discussed in Section 5.3.2 and the post-processing approach that was adopted to work around it. Lastly, a gradually increasing discrepancy can be observed towards the end of the simulations (e.g. Figure 6.19-c). As will be discussed in more detail in Section 7.1, it is suspected that additional inflow was provided during the physical experiments by the flume’s water supply system. As no additional inflow was considered in the simulations, this resulted in numerical bores of slightly shorter duration.

It can be noticed that the magnitudes of the impact pressures predicted by the numerical model are considerably high with respect to subsequent pressures (e.g. Figure 6.19-a). This was initially investigated by verifying the assumption of a weakly compressible fluid (see Section 2.3.2) in the SPH simulations. According to Monaghan [1994], density variations with respect to the reference water density ($\rho_0 = 1000 \text{ kg/m}^3$) should remain within 1% in order to obtain accurate results when considering this assumption. Therefore, the maximum density fluctuations for each of the impoundment depths within the impact region directly upstream of the column were determined. These

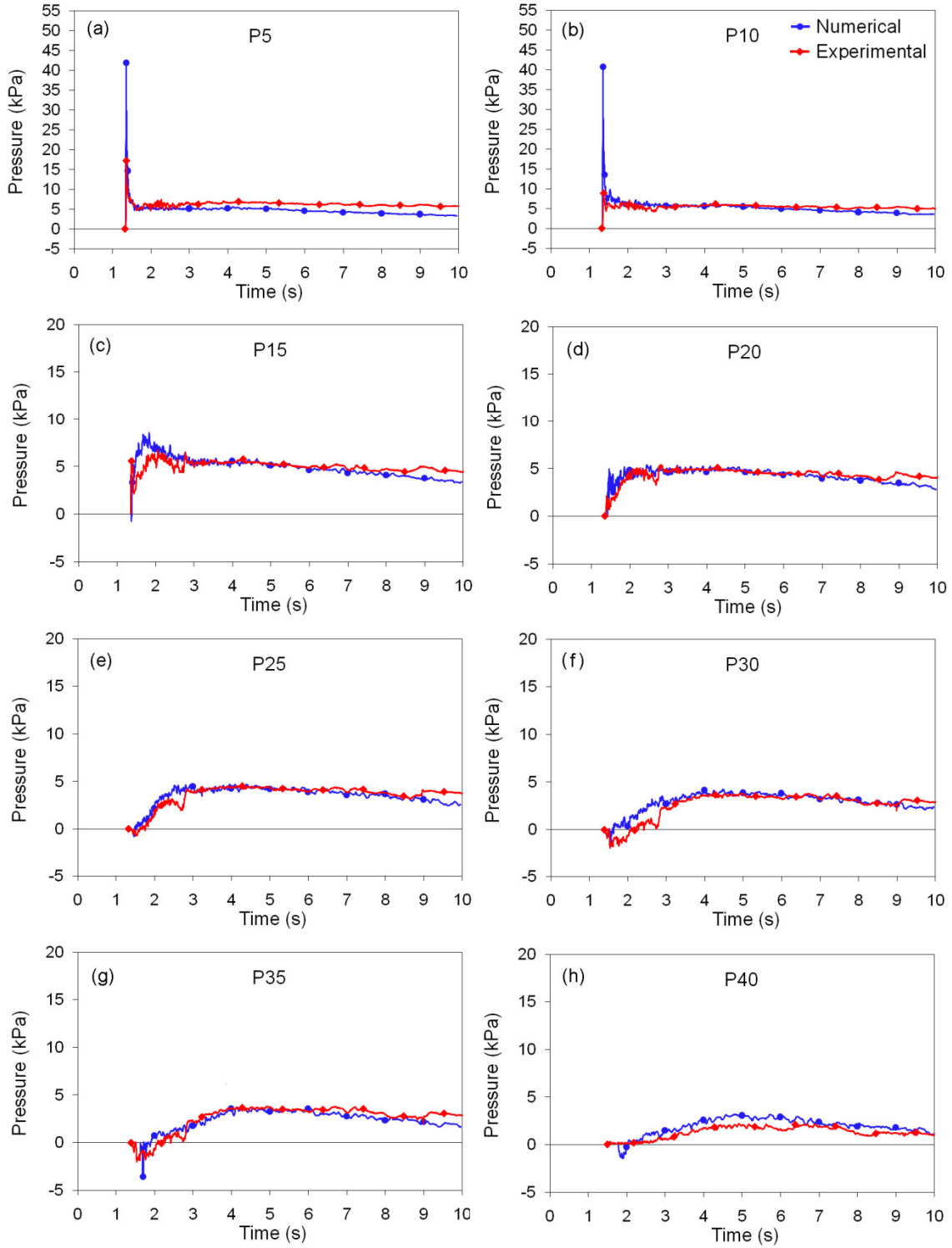


Figure 6.17. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.85\text{m}$ (experimental run E-85 and simulation D2-85).

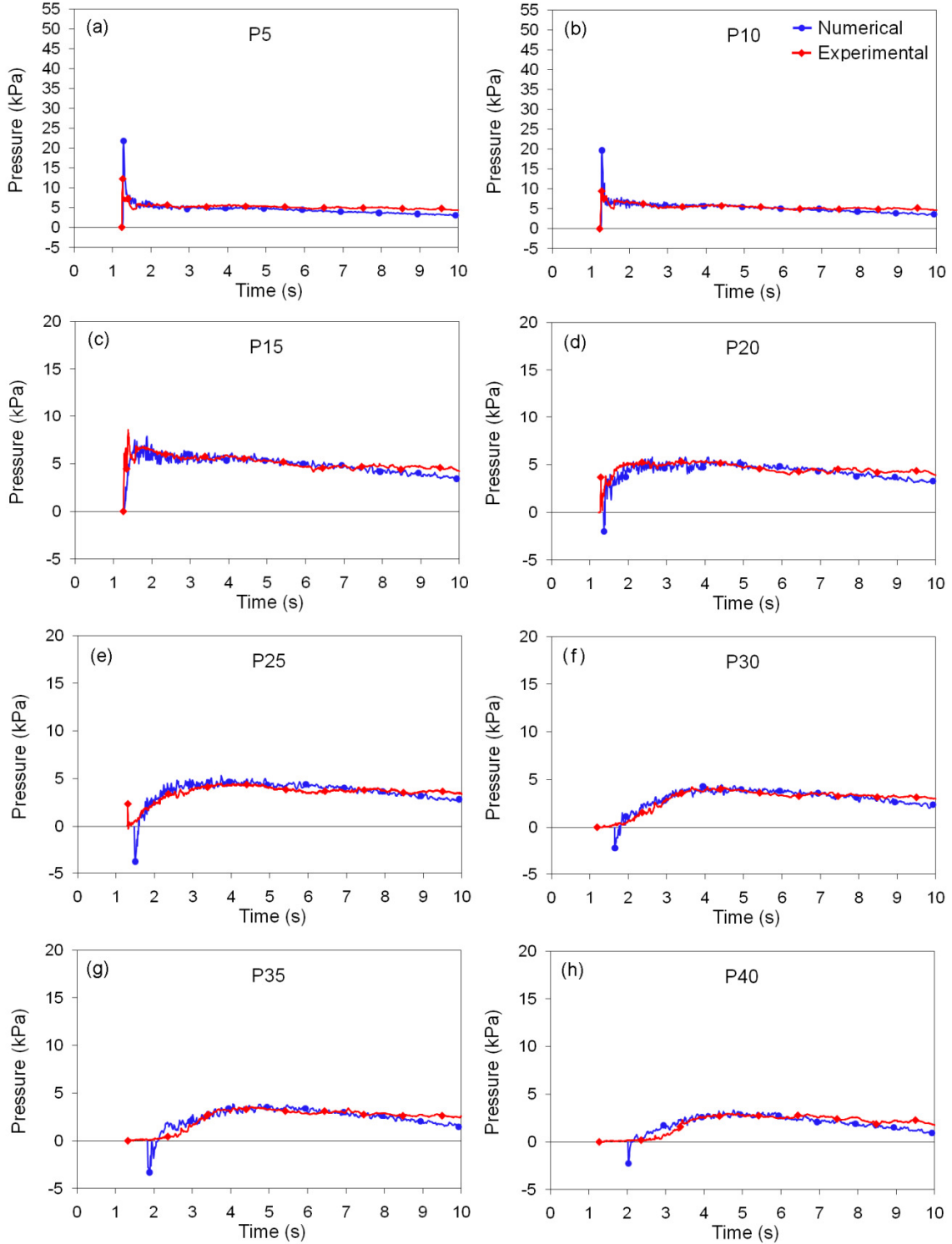


Figure 6.18. Comparison of numerical and experimental time-history of pressures acting on the circular column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 0.85\text{m}$ (experimental run E-85-C and simulation D2-85-C).

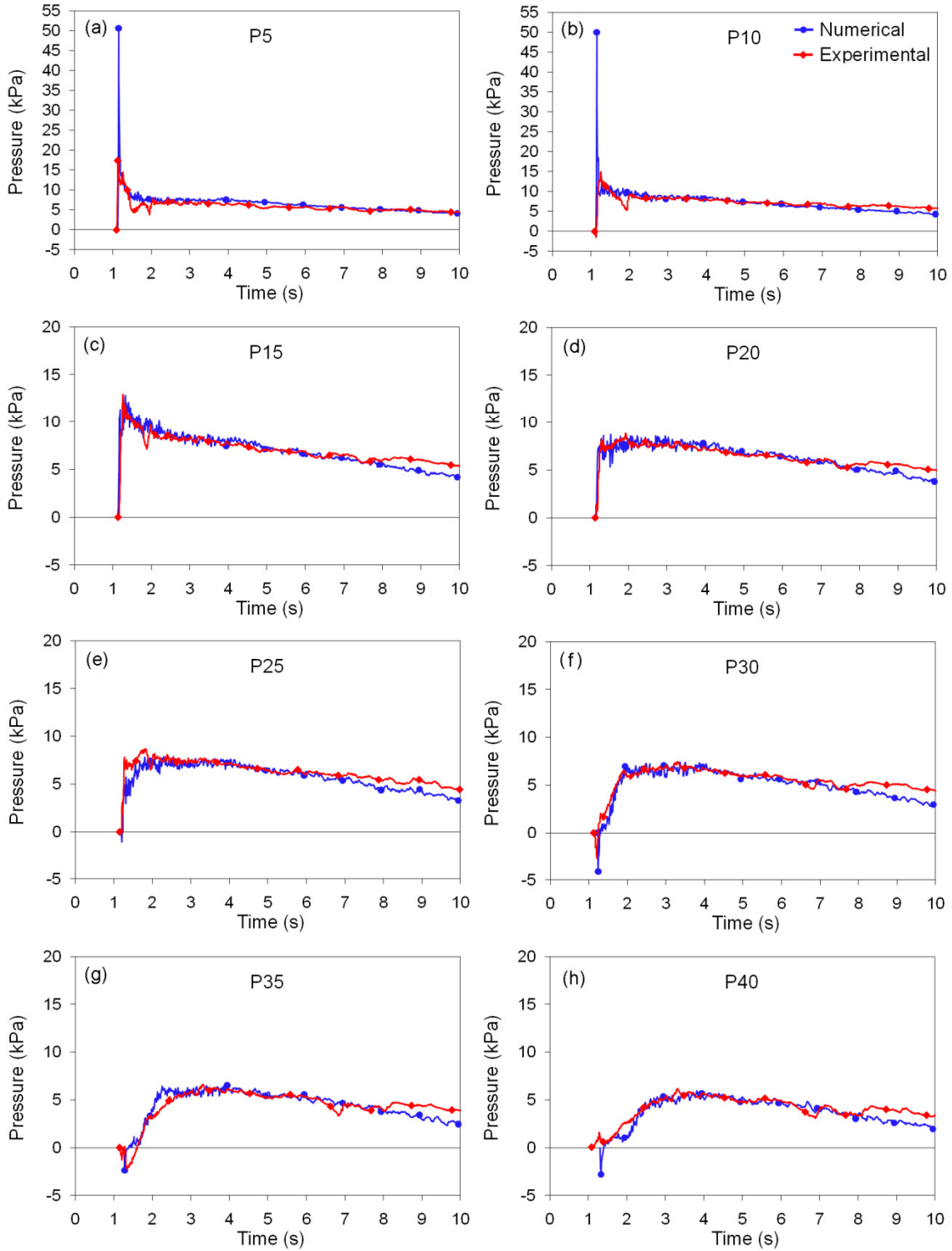


Figure 6.19. Comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 1.15\text{m}$ (experimental run E-115 and simulation D2-115).

fluctuations are presented in Table 6.2. It can be seen that, for all impoundment depths, the maximum density variation which occur at initial impact is slightly greater than 1%. Note that, according to Eq. 4.2, as the speed of sound is increased with the impoundment depth, the compressibility of the simulated fluid decreases accordingly. This explains that, although the impact force increases with the impoundment depth, the maximum density variations remain approximately the same.

Table 6.2. Numerical water density at initial impact and corresponding variation with respect to the reference density ($\rho_0 = 1000 \text{ kg/m}^3$).

Simulation	h_u (m)	Impact Time (s)	ρ_{impact} (kg/m^3)	$ \Delta\rho /\rho_0$ (%)
D2-55	0.55	1.82	1017.06	1.71
D2-85	0.85	1.23	1015.63	1.56
D2-115	1.15	1.12	1017.03	1.70

Considering another moment in the bore-structure interaction where better agreement is generally observed, i.e. when the resulting force on the column reaches its second peak (see Figure 6.22), Table 6.3 shows the density variations in the simulated fluid once again directly upstream of the column. In this second case, it can be observed that these variations remain below 1%.

The latter initially led one to believe that the high numerical pressures at initial impact were perhaps due to excessive increases in the density, with which the pressure is directly computed using the equation of state (Eq. 2.66). However, by comparing in Figure 6.20 the time-history of the pressure at transducer P5 obtained with the SPH method (simulation D2-115) with that obtained by Zhihui Ma, using grid-based FVM along with VOF, very similar magnitudes are observed. As the latter numerical method considers the fluid to be strictly incompressible, it can be concluded that the relatively large density fluctuations presented above do not introduce inaccuracies in the numerical prediction of the pressure at initial impact.

Finally, Figure 6.21 shows the numerical time-histories of the pressure at all transducers and for all impoundment depths. As expected, due to a higher bore-front

Table 6.3. Numerical water density at the moment of maximum force due to runup and corresponding variation with respect to the reference density ($\rho_0 = 1000 \text{ kg/m}^3$).

Simulation	h_u (m)	Runup Time (s)	ρ_{runup} (kg/m^3)	$ \Delta\rho /\rho_0$ (%)
D2-55	0.55	5.73	1002.55	0.26
D2-85	0.85	4.67	1003.00	0.30
D2-115	1.15	3.76	1003.33	0.33

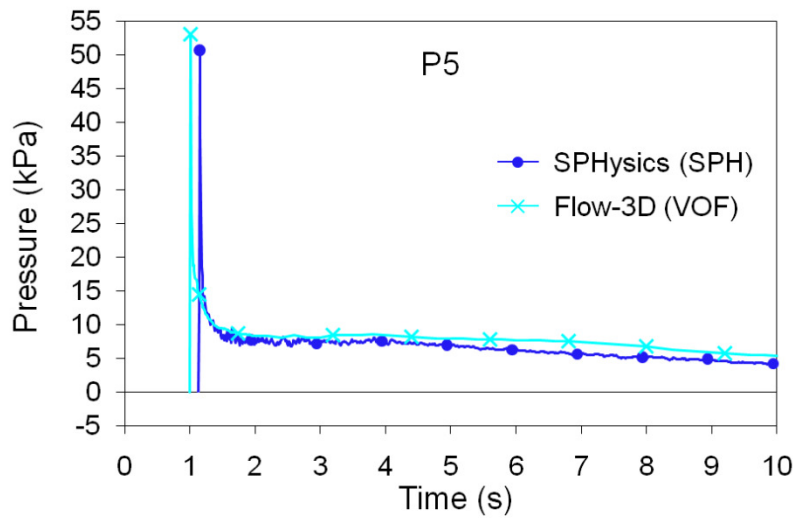


Figure 6.20. Comparison of the numerical time-history of the pressure acting on the column at an elevation of 0.05m above the flume's bottom obtained using the SPH method (simulation D2-115) and FVM with VOF ($h_u = 1.15\text{m}$ and $h_d = 0$). FVM results courtesy of Zhihui Ma.

velocity, the impact pressure increases with increasing impoundment depth. In the case of the negative pressures exerted at higher elevations on the column at initial impact, they also generally increase with increase in the impoundment depth. This could possibly be due to higher impact velocities that result in greater water reflection directed upstream (e.g. Figure 6.9-a). As expected, as flow velocities and runup are also more important, post-impact pressures also increase with increase in the impoundment depth.

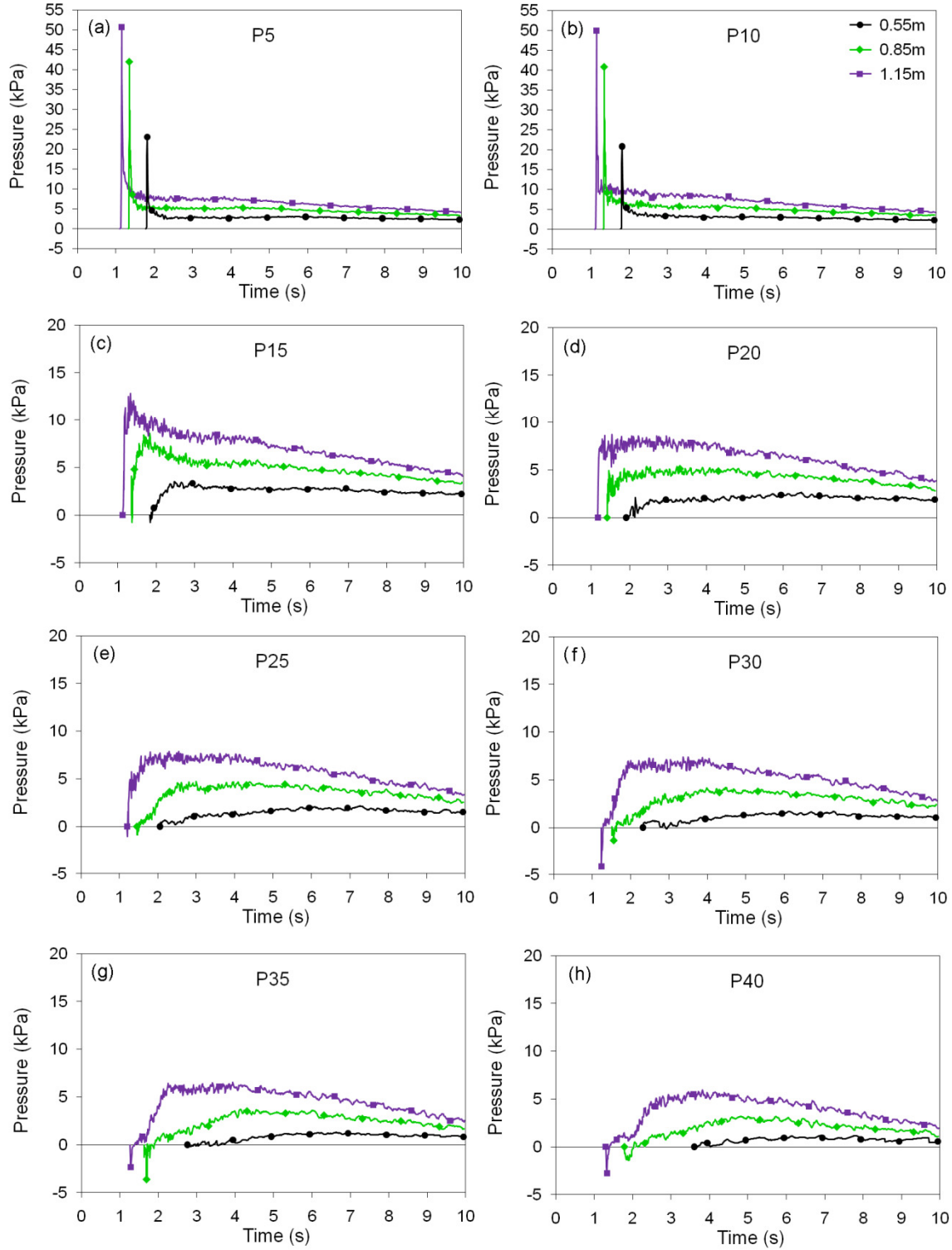


Figure 6.21. Numerical time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).

6.1.4 Force

By integrating the numerical pressure exerted on its entire surface, the numerical time-history of the net base horizontal force acting streamwise on the column (F_X) was obtained and compared with the experimentally-obtained one captured with the dynamometer. Figure 6.22 shows the comparison for all impoundment depths. In the case of the impoundment depth of 0.55m, both experimental dry- and wet-bed conditions are considered (Figure 6.22-a and Figure 6.22-b, respectively).

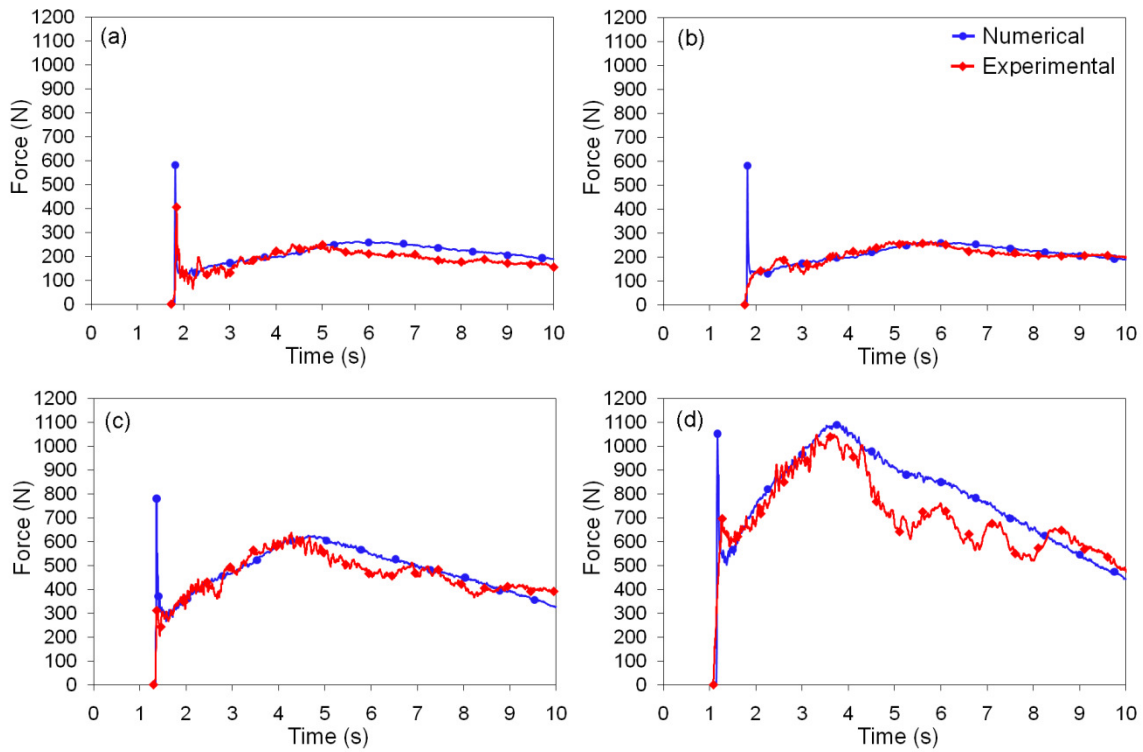


Figure 6.22. Comparison of numerical and experimental time-history of the net base horizontal force acting on the column for impoundment depths (h_u) of (a) 0.55m with a virtually dry bed ($h_d \approx 0$), (b) 0.55m with a wet bed ($h_d \approx 0.005$ m), (c) 0.85m, and (d) 1.15m.

Similarly, as for the case of the pressure, the numerical time-history of the force displays a distinctive peak at initial impact while, depending on the bed condition, the experimental one generally does not. Beyond initial impact and for all impoundment depths, the rate at which the force increases due to the runup of the bore on the column

Chapter 6 Comparison of Numerical with Physical Results - Experiments of Nouri et al. [2010] and Al-Faesly et al. [2012]

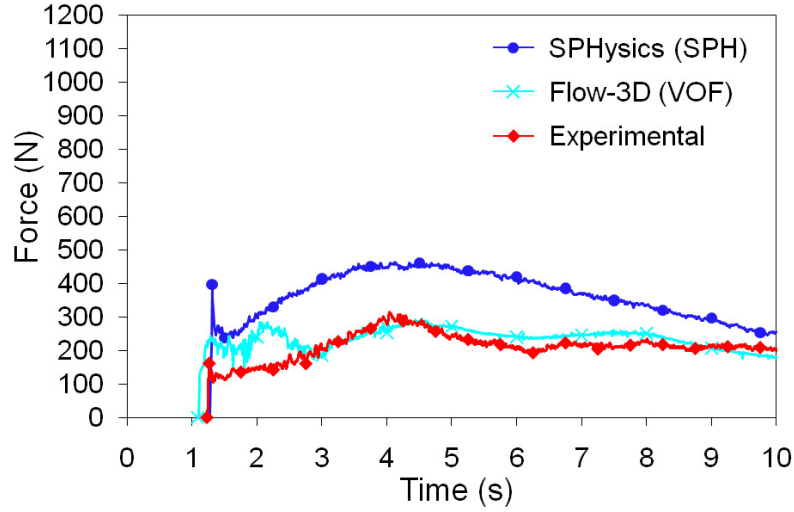


Figure 6.24. Comparison of numerical and experimental time-history of the net base horizontal force acting on the circular column for $h_u = 0.85\text{m}$. $h_d = 0$ for the numerical simulation performed using the SPH method (D2-85-C) while $h_d = 0.001\text{m}$ for the one performed using FVM with VOF. FVM results courtesy of Zhihui Ma.

column by the integration of the pressure was carefully revised (see Section 5.2), although no issue was found. Next, in order to verify the accuracy of the experimental measurements obtained with the dynamometer, the experimental run was also simulated by Zhihui Ma using FVM with VOF (Figure 6.24). Although he considered a wet-bed condition ($h_d = 0.001\text{m}$), which further confirms the reduction in the impact force observed in the physical experiments (e.g. Figure 6.22), relatively good agreement is observed in this case. Strangely, there is good agreement between the numerical (SPH) and experimental results of the pressures acting at the very front of the circular column (Figure 6.18). Since the resulting force is derived from integration of the pressure, one would expect reasonable agreement for it as well. Although the exact reason for this disagreement is unknown, it is suspected to be due to the following: in contrast to the square column (Figure 5.7), the void of fluid particles and numerical layer of unrealistic pressures that develops at the front of the column (see Section 5.3.2) could possibly increase numerically the area of the circular column projected to the flow (Figure 5.8). In other words, the effective diameter of the circular column in the numerical simulation (simulation D2-85-C) could be greater than the actual one of the physical experiment (run

E-85-C). Furthermore, for the numerical predictions of the pressures to be accurate at the very front of the column (i.e. at a stagnation point), this would imply that the column's diameter has limited influence on the pressures at this particular location. Hence, through further investigation, it would be important to also validate the pressures acting at other locations around the circular column (i.e. not just at the front of the column).

As the experimental time-history of the net base horizontal force was obtained at a frequency of 1000Hz (Table 4.1), it is possible to examine the effect of the flume's bed condition on the rise time of the force at initial impact. Over a short time interval covering initial impact, Figure 6.25 shows the experimental time-history of net force acting on the column for both bed conditions (runs E-55-DRY and E-55-WET). The full time-history of the net force corresponding to these experimental runs can be seen in Figure 3.11-b.

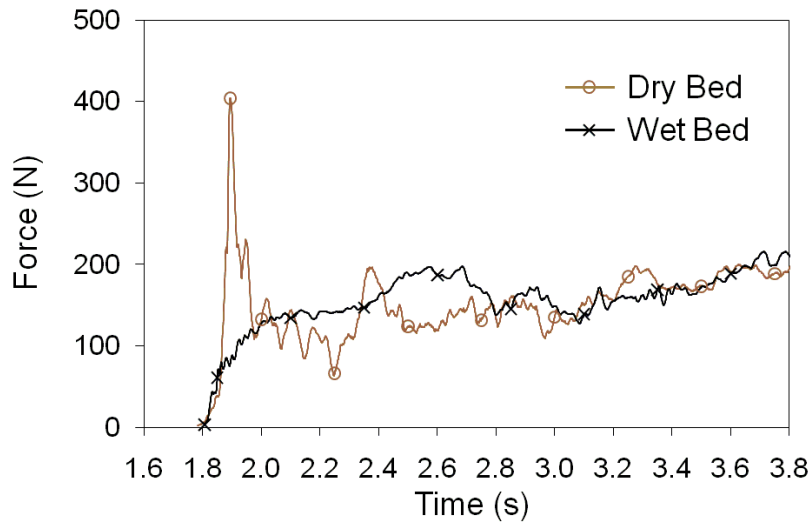


Figure 6.25. Comparison of the experimental rise time of the net base horizontal force acting on the column due to the impact of bores ($h_u = 0.55\text{m}$) propagating on dry ($h_d \approx 0$) and wet ($h_d \approx 0.005\text{m}$) beds (experimental runs E-55-DRY and E-55-WET).

The time required for the first distinct peak to be reached is 0.110 and 0.879 seconds for the dry- and wet-bed conditions, respectively. Having a rise time greater in the case of the wet-bed condition supports the aforementioned hypothesis that more air is entrapped into

the bore-front as the latter propagates over a thin water layer. [Hattori *et al.* [1994] noticed that the time required for air bubbles to compress increases the rise time].

In a similar way, Figure 6.26 shows, over a short time interval covering initial impact, the comparison between the numerical and experimental time-history of the net force acting on the column for $h_u = 0.55\text{m}$. The numerical results correspond to a run with a perfectly dry bed (simulation D2-55), while the experimental ones correspond to a run with a “virtually” dry bed, as previously discussed.

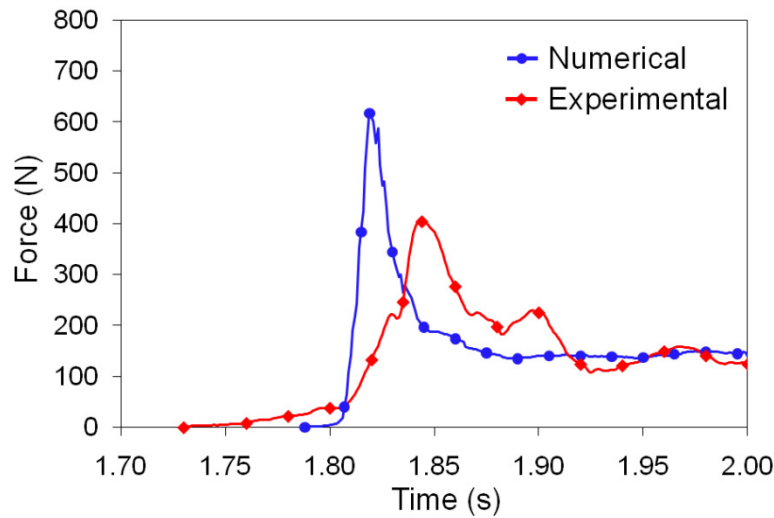


Figure 6.26. Comparison of the experimental and numerical rise time of the net base horizontal force acting on the column due to the impact of a bore ($h_u = 0.55\text{m}$) propagating on a dry bed (experimental $h_d \approx 0$; experimental run E-55-DRY and simulation D2-55).

It can be seen that the rise time of the force impulse at initial impact is longer in the case of the physical experiment (0.110 compared to 0.029 seconds). This demonstrates that, although the bore propagated over a “virtually” dry bed, a certain degree of air entrainment still occurs. As the numerical model does not model the air phase, the predicted magnitude of the force impulse is therefore greater, as the cushioning effect of entrapped air bubbles generally reduces it [Peregrine, 2003].

Figure 6.27 shows the numerical time-history of the net base horizontal force acting on the square column for all impoundment depths considered. As the bore-front

velocity increases with the impoundment depth, the force impulse at initial impact, or the impact force, is also expected to increase. Furthermore, compared to this impact force, the second peak in the time-history, or the runup force, becomes more significant as the impoundment depth increases. This runup force also occurs earlier in the bore-structure interaction. A similar behaviour of the runup force can also be observed in the results of Árnason [2005] (Figure 3.12). [As Árnason's experiments were performed with a wet-bed condition ($h_d = 0.02\text{m}$), he did not observe a large spike in the time-history of the force at initial impact].

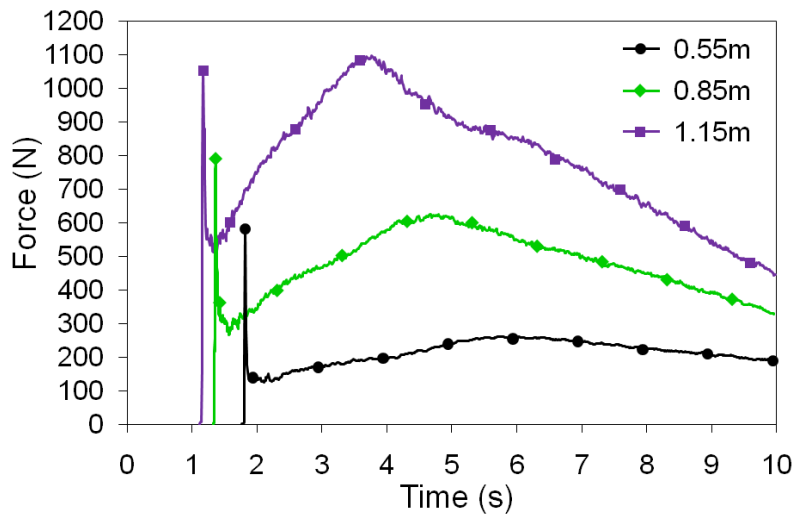


Figure 6.27. Numerical time-history of the net base horizontal force acting on the column for all impoundment depths h_u (simulations D2-55, D2-85, and D2-115).

6.2. Experiment of Kleefsman *et al.* [2005]

The primary goal of incorporating the results of this experiment in the present study was to investigate the influence of particle resolution on the “numerical viscosity” associated with the Riemann solver utilized for removing inaccuracies in the pressure field (see Section 2.3.5). As mentioned in Section 6.1.1, the lack of water splashing at initial impact of the hydraulic bore with the structure is believed to be the main manifestation of this excessive viscosity. Restricted to the serial version of SPHysics, the initial inter-particle spacing (Δ) was decreased from 0.0350 to 0.0155m (see Table 4.12).

6.2.1 Water Surface Elevation

As suspected, compared to the initial water splash observed in the physical experiment (Figure 6.31-d), Figure 6.28 shows that the numerical simulation of such splash is considerably improved with the use of a smaller Δ . Splashing is essentially non-existent for $\Delta = 0.0350\text{m}$, which is the same value that was adopted in simulating the experiments of Nouri *et al.* [2010] and Al-Feasly *et al.* [2012]. This provides a clear explanation for the lack of splashing previously observed in Figure 6.9-a. Nevertheless, the splashing observed in the physical experiment (Figure 6.31) is still more significant, and an even smaller initial inter-particle spacing would be required to reproduce it exactly.

The influence of the particle resolution on the water surface elevation at wave gauges W1, W2, W3, and W4 is shown in Figure 6.29. The locations of these gauges are shown in Figure 4.6. For wave gauges upstream of the obstacle (W2 and W3), the arrival time of the hydraulic bore is not visibly affected as Δ is decreased below 0.0350m . In the sensitivity analysis (Figure 4.8) it was observed that the bore-front velocity slightly converges for Δ below this value. However, since the distances of gauges W2 and W3 from the reservoir gate are relatively small (0.504 and 1.000m , respectively) such convergence is not noticeable in Figure 6.29. On the other hand, the particle resolution has a significant influence on the water surface elevation. This is demonstrated by the more defined time-history of the water surface elevation for $\Delta = 0.0155\text{m}$, for which peaks are generally higher and with a more sudden rise.

Considering the numerical simulation with the highest particle resolution (D5-HIGH), the time-histories of the water surface elevation are compared to experimental ones in Figure 6.30. Magnitude-wise, relatively good agreement is observed. However, a time lag, varying from 0.21 to 0.36 seconds, is observed as the flow propagates back upstream after reflecting on the downstream wall of the tank, to be subsequently reflected back downstream on the upstream wall. Such flow reflection is shown in Figure 6.31, which compares still frames of simulation D5-HIGH to experimental ones. At W1 (Figure 6.30-a), the first sudden rise in the water surface elevation is due to the abrupt front of the bore resulting from the reflection on the downstream wall (Figure 6.31-i).

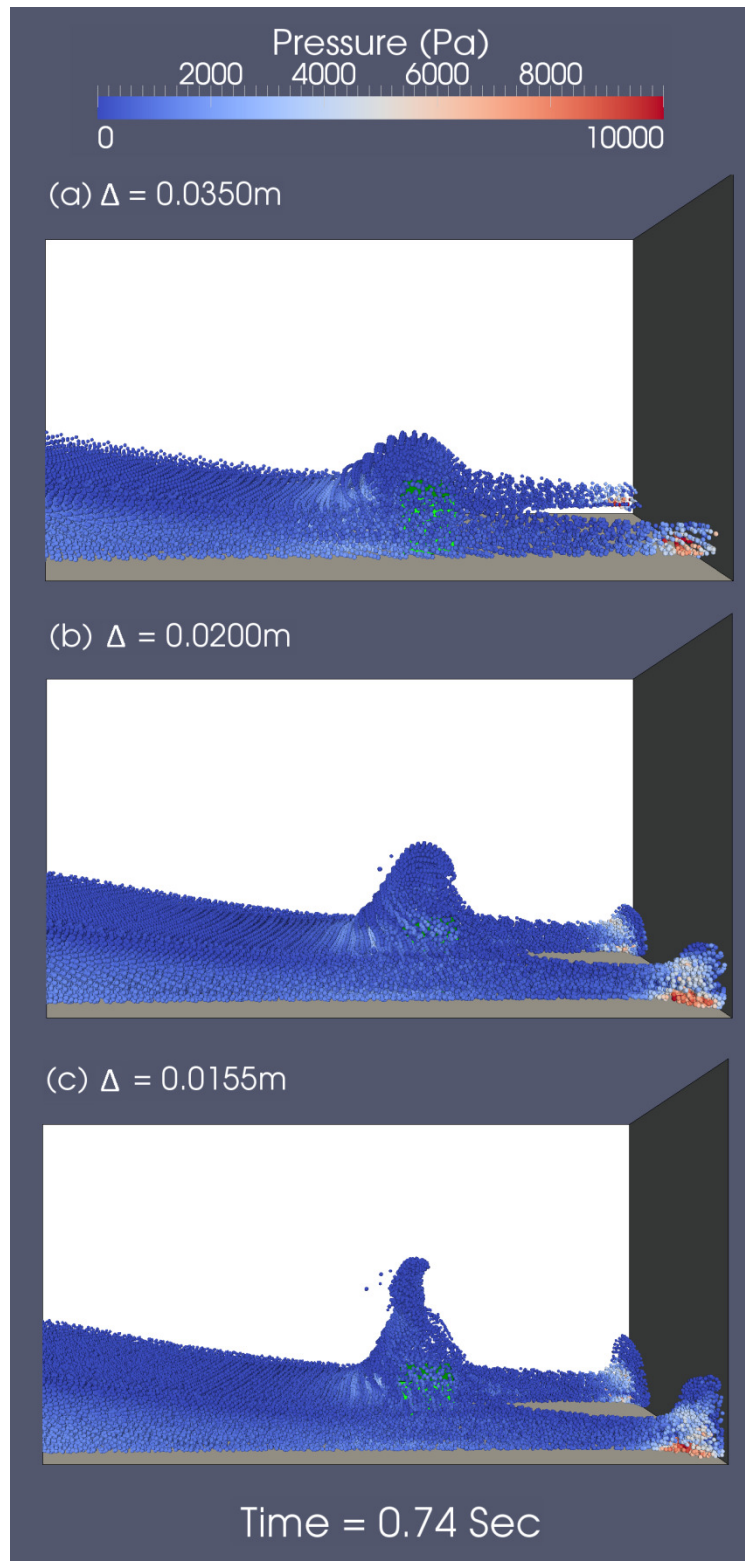


Figure 6.28. Comparison to the numerical water splash at initial wave impact for inter-particle spacings Δ of (a) 0.0350, (b) 0.0200, and (c) 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.

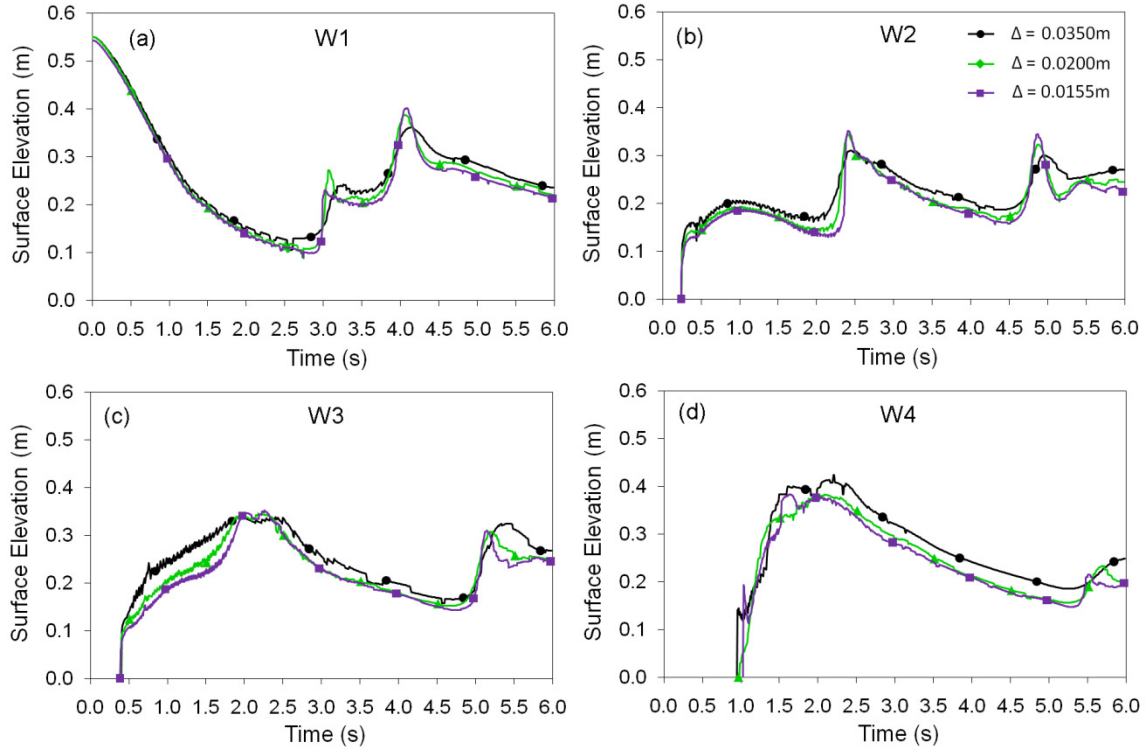


Figure 6.29. Numerical time-history of the water surface elevation at wave gauges (a) W1, (b) W2, (c) W3, and (d) W4 for inter-particle spacings Δ of 0.0350, 0.0200, and 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.

The time lag in the arrival of this reflected bore is 0.21 seconds. As this bore then reflects on the upstream wall, the water elevation suddenly rises again in the reservoir, giving place to the second peak in Figure 6.30-a. The corresponding time lag has slightly increased to 0.25 seconds. At W2 and W3 (Figure 6.30-b), the second sudden rise in the time-history of the water surface elevation shows that, as the wave resulting from the reflection with the upstream wall arrives at the location of the wave gauges, the time lag has further increased to approximately 0.36 seconds. Hence, this discrepancy in the propagation of the reflected wave increases over time. Such lag in the time-history of the water surface elevation has also been observed in previous studies in which the experiment of Kleefsman *et al.* [2005] was considered (see literature review). Simulating the experiment with both WCSPH and ISPH models, Lee *et al.* [2010] attributed this discrepancy to the coarse particle resolution they employed ($\Delta = 0.0183\text{m}$). However, even with the convergence of the wave arrival time observed in Figure 6.29, the time lag

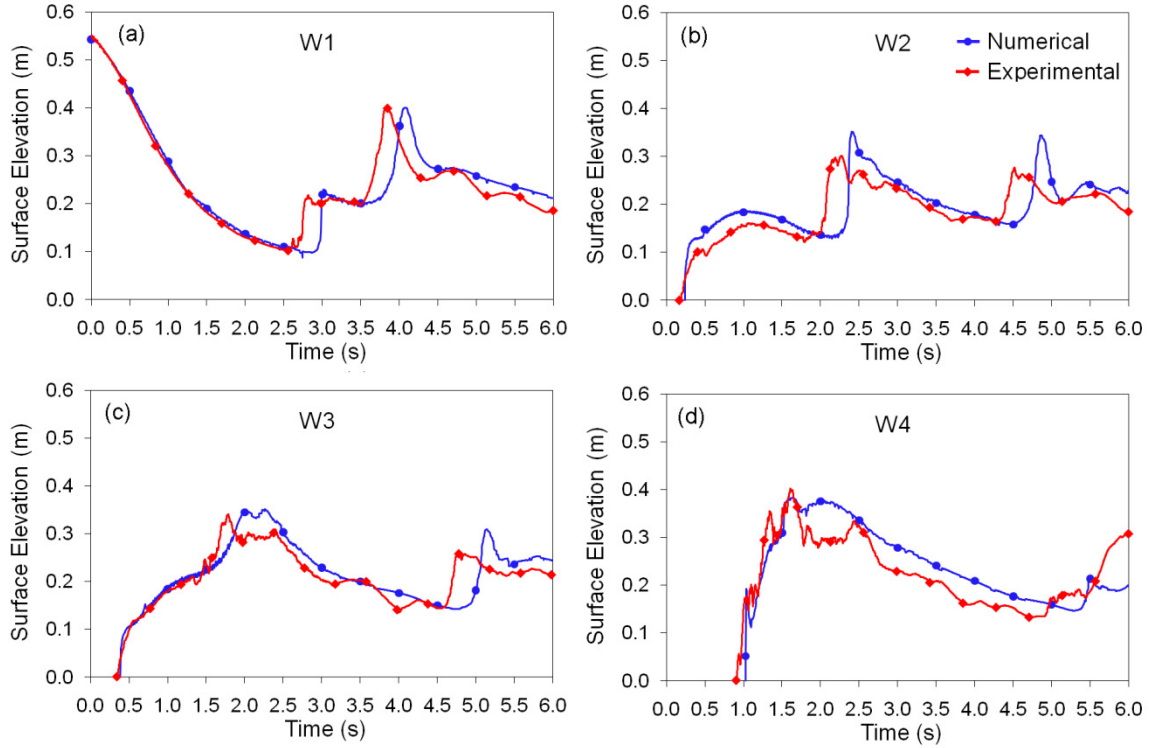
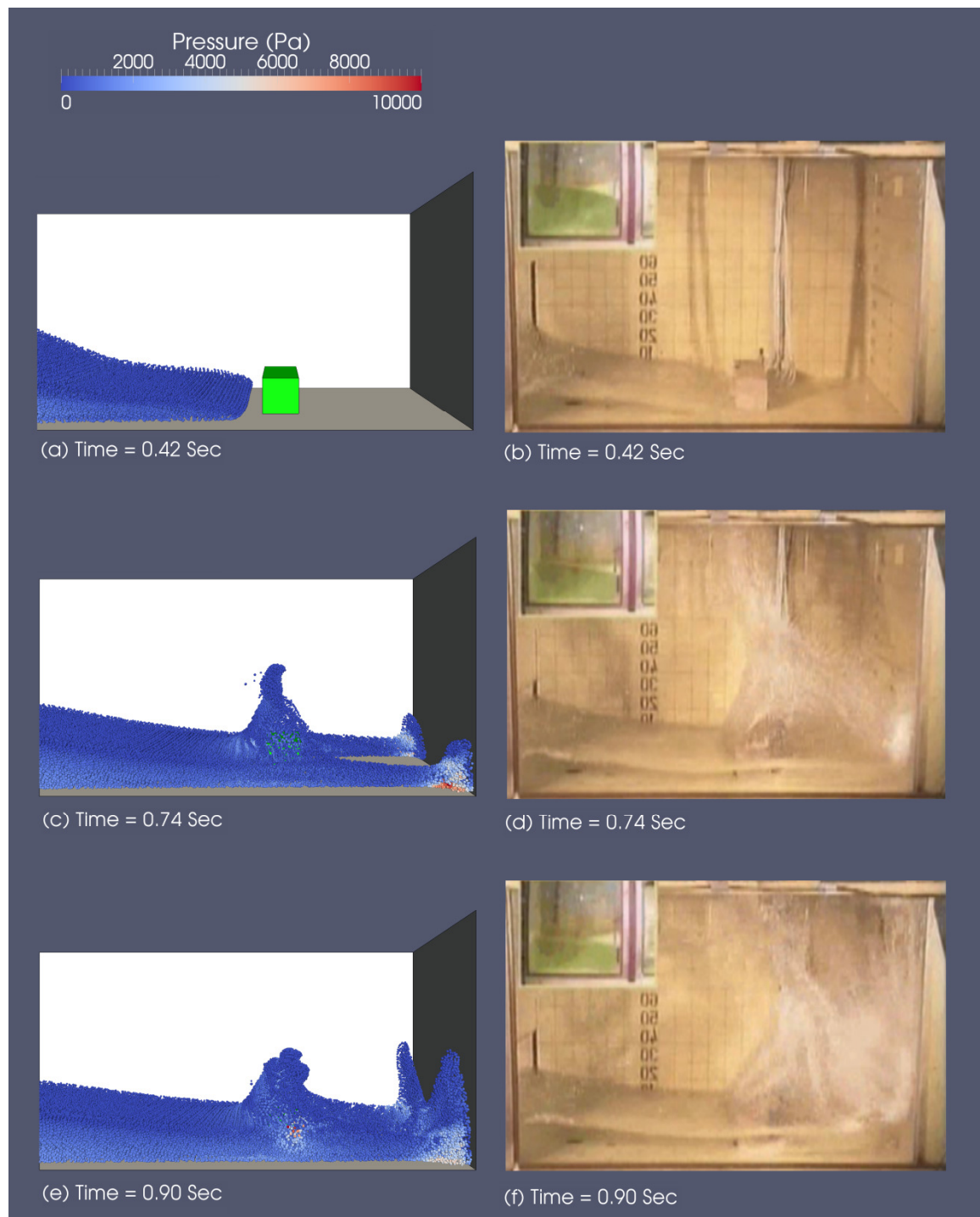


Figure 6.30. Comparison of numerical (simulation D5-HIGH) and experimental (Kleefsman *et al.* [2005]) time-history of the water surface elevation at wave gages (a) W1, (b) W2, (c) W3, and (d) W4.

is still present for the smaller Δ value adopted in the current research. Based on the numerical results of Lee *et al.* [2010] and the ones obtained in this research, it seems that this discrepancy is neither associated with the particle resolution nor the compressibility of the simulated fluid (i.e. compressible or incompressible). Furthermore, as Lee *et al.* did not employ a Riemann solver, the discrepancy cannot be associated with this particular solution approach.

6.2.2 Pressure

In contrast to the water surface elevation, Figure 6.32 shows that, for all pressure transducers presented in Figure 4.7, the overall time-history of pressure on the front and top faces of the obstacle is not significantly improved with further increases in the particle resolution. At initial wave impact however, the pressure impulse at transducer P1 is considerably higher for $\Delta = 0.0155\text{m}$ (simulation D5-HIGH). This difference is perhaps



For caption see page 171.



For caption see page 171.

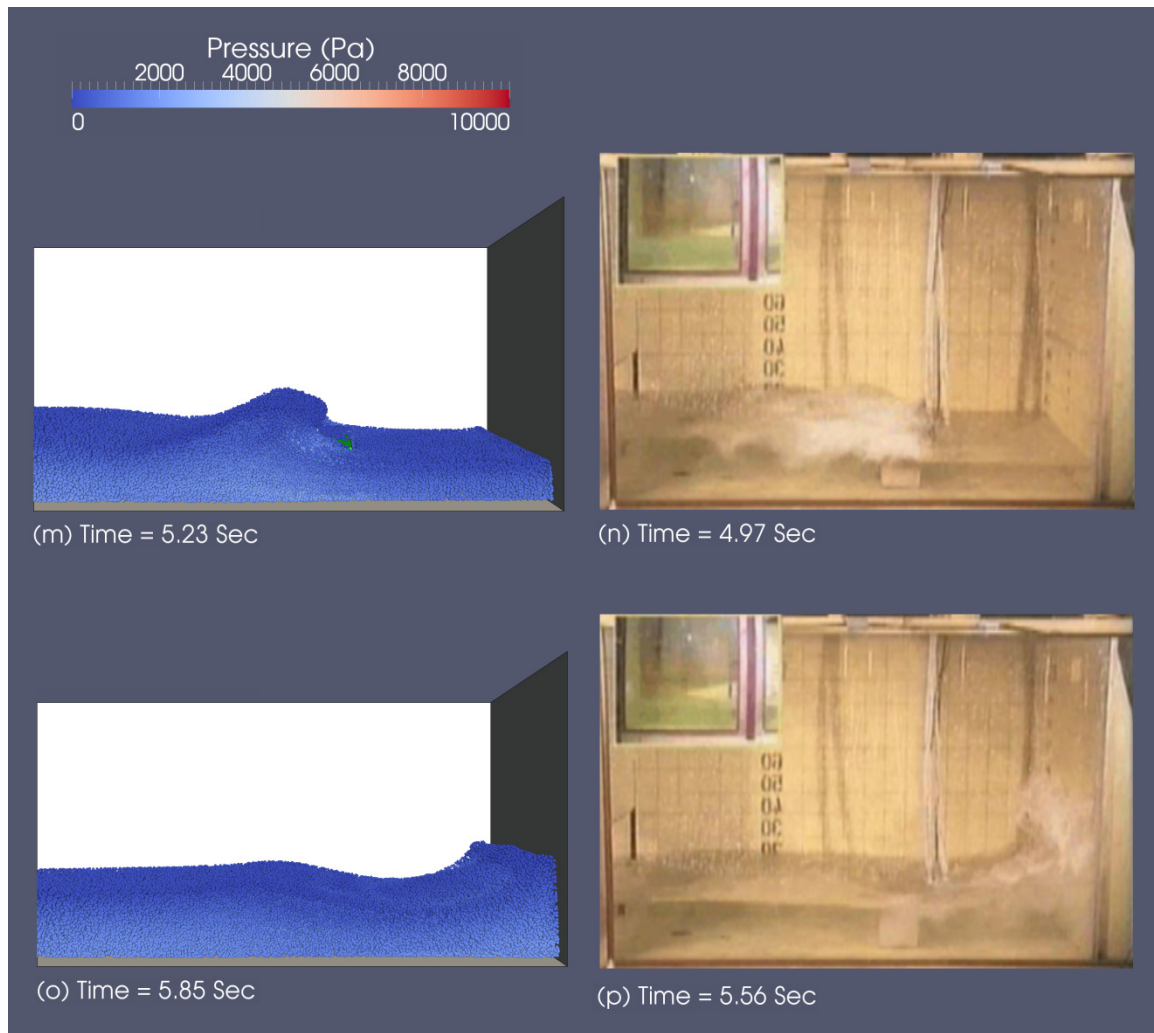


Figure 6.31. Qualitative comparison of numerical (left, simulation D5-HIGH) and experimental (right, Kleefsman *et al.* [2005]) water surface profiles.

due to the fact that the numerical outputs of this simulation were obtained at a frequency of 200Hz compared to 100Hz in the case of the other two. Moreover, in the case of the pressure transducers located on top of the obstacle (P5 to P8) small pressure peaks at approximately 5.25 seconds develop as the particle resolution increases. From Figure 6.31-m, this peaks occurs because, after reflection on the upstream wall, the wave breaks directly on top of the structure.

Considering the simulation with the highest resolution (D5-HIGH) for the comparison of the numerical time-histories of the pressure with the ones measured experimentally, good agreement can be observed in Figure 6.33. As in the time-history of

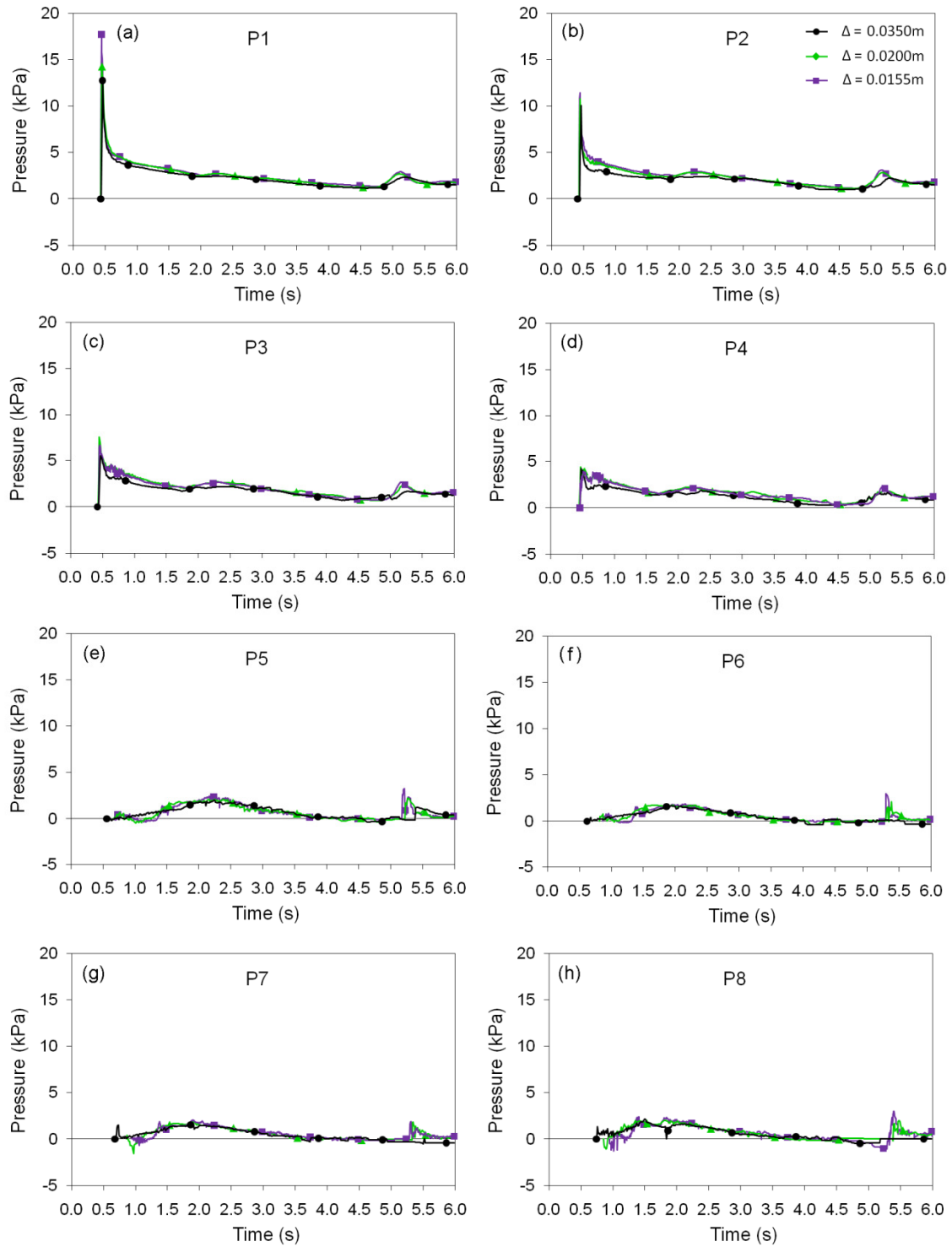


Figure 6.32. Numerical time-history of the pressure at all pressure transducers for inter-particle spacings Δ of 0.0350, 0.0200, and 0.0155m corresponding to simulations D5-LOW, D5-MED, and D5-HIGH, respectively.

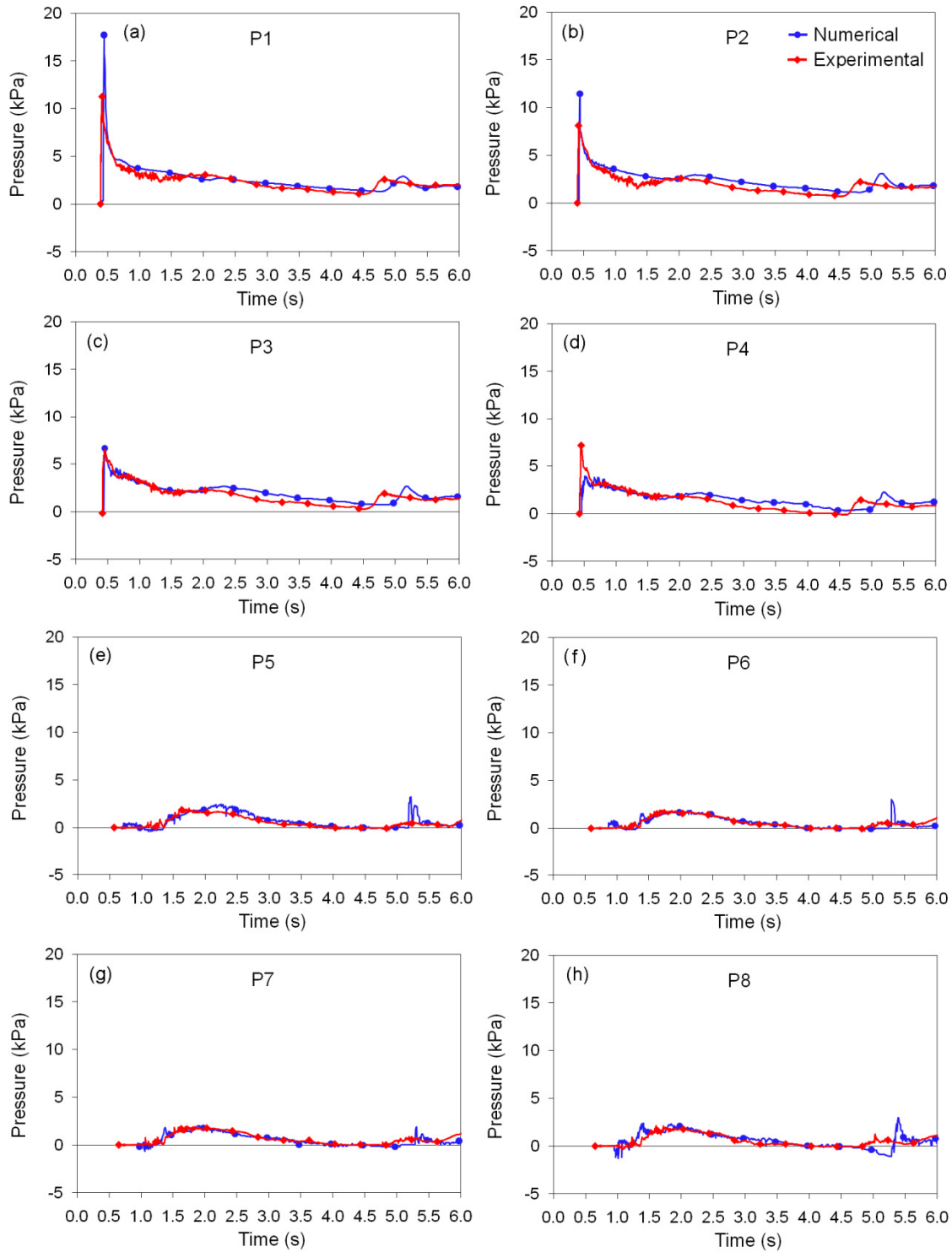


Figure 6.33. Comparison of numerical (simulation D5-HIGH) and experimental (Kleefsman *et al.* [2005]) time-history of the pressure at all pressure transducers.

the water surface elevation, a time lag is visible for the pressure transducers located at the front of the obstacle (P1 to P4). Such a time lag can also be observed in the numerical results of Lee *et al.* [2010] and Crespo *et al.* [2011] in Figure 3.18 and Figure 3.20, respectively. It was also noted that, on the top of the obstacle (pressure transducers P5 to P8), no peaks in the pressure are recorded in the physical experiment. The still frames in Figure 6.31 demonstrate that, after the reflection with the upstream wall, the experimental wave breaks before reaching the obstacle (Figure 6.31-l), while in the numerical simulation (Figure 6.31-m), the wave breaks directly on top of it. Lastly, as in the study of Crespo *et al.* [2011] (Figure 3.20), the numerical model also over-predicts the pressure at initial impact at transducer P1 (Figure 6.33-a).

6.3. Discussion

The quantitative comparison of the numerical with physical results performed in this chapter demonstrates the effectiveness of the WCSPH numerical model. Employing a Riemann solver for the correction of instabilities in the pressure field allowed for good agreement between the observed and simulated hydrodynamic forces, even with a relatively coarse particle resolution ($\Delta = 0.0350\text{m}$). However, by introducing excessive “numerical viscosity”, this Godunov-based correction scheme rendered the reproduction of complex flow features, in particular splashing at initial wave impact, more difficult. To mitigate this detrimental side effect, a particularly high particle resolution is required. Also, with its area projected to the flow believed to be increased numerically (Figure 5.8), agreement in the resulting hydrodynamic forces acting on the circular column was poor (Figure 6.24). This could have been further investigated by adjusting the column’s diameter such that its ‘effective’ area projected to the flow is the same as the one in the physical experiment.

For the experiments of Nouri *et al.* [2010] and Al-Feasly *et al.* [2012], the bed condition (i.e. dry vs. wet) was shown to have a significant influence on the magnitude of the force at initial wave impact. As most experimental runs were performed with a wet- or partially wet-bed condition, such impulsive forces were generally nonexistent in the experimental data. The assessment of the rise time of this impact force would suggest that the entrapped air in the bore-front plays a major role in its inhibition. Since the numerical

model only simulates the water phase and since only simulations with a dry bed were possible (due to the difficulty in reproducing the exact thickness of the experimental downstream water depth), considerably large numerical impact forces were generally observed.

Lastly, considering the experiment of Kleefsman *et al.* [2005], the propagation and reflection process of the numerical hydraulic bore in the closed test tank lagged in time. This behaviour was also observed in previous SPH-related studies. Although the exact cause for this discrepancy is unknown, the results presented in this research, together with those of earlier related studies, leads one to conclude that it was not due to the particle resolution, the compressibility of the simulated fluid, or the use of a Riemann solver. Time-related discrepancies in the upstream propagation of the obstruction-induced wave and in the formation of the downstream wake were also noticed in the experiments of Nouri *et al.* [2010] and Al-Feasly *et al.* [2012]. However, based on additional numerical simulations undertaken in this study, plausible explanations for these cases are presented in the next chapter.

Chapter 7. Results of Numerical Experiments

Validated with a reasonable level of confidence, the numerical model is further utilized in this chapter to investigate important aspects of the physical experiments. These aspects are (i) the reservoir geometry, (ii) the width of the test channel, (iii) the reservoir size, and (iv) the downstream bed condition (i.e. dry vs. wet). These additional numerical simulations had a two-fold purpose: (i) to investigate further and improve our understanding of the underlying physical phenomena involved in the bore-structure interaction, and (ii) to assist in explaining some of the discrepancies observed in the qualitative comparison exercises performed between the numerical simulations and the experimental data.

7.1. Influence of Reservoir Geometry

Figure 4.2 shows a schematic of the experimental setup of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012]. In order to maximize the duration of the hydraulic bores, which were generated based on the dam-break approach, the upstream reservoir was partitioned as shown. With this modification, the reservoir geometry has a sharp 180° bend, rather than a straight section (which is the case in the ideal dam-break scenario). Furthermore, there is an expansion and subsequent constriction of the flow width in this bend. The purpose of this particular modeling exercise is to evaluate the influence of such a bend in the reservoir unit on the overall results of the physical experiments. To do this, a numerical simulation (D3-L2) was performed with a hydraulic bore generated from a straight reservoir having equal surface area (see Figure 4.14). The dimensions of this ‘equivalent’ reservoir were selected as $L = 11.20\text{m}$ and $W = 1.30\text{m}$.

7.1.1 Runup and Force

Figure 7.1 demonstrates that the reservoir geometry has a considerable influence on the time-history of the runup (wave gauge W9) and therefore, on the net base horizontal force acting streamwise on the column. As the reservoir with a sharp 180° bend (hereafter referred to as the “bend” reservoir), has a 3.28m straight section upstream of the gate (see Figure 4.2), no differences during the initial stages of the bore-structure interaction are expected. However, beyond ~3 seconds into the simulation, significant differences can be observed. First, a higher runup is reached in the case of the bend reservoir and consequently a higher runup force is also reached (625.76 compared to 591.92N). Secondly, in the case of the straight reservoir, the maximum runup is attained more gradually and also later in the simulation. The same is true for the time-history of the force (4.67 compared to 6.22 seconds). Furthermore, the rate at which the runup and the force decrease once their maximums have been reached is more gradual in the case of the straight reservoir.

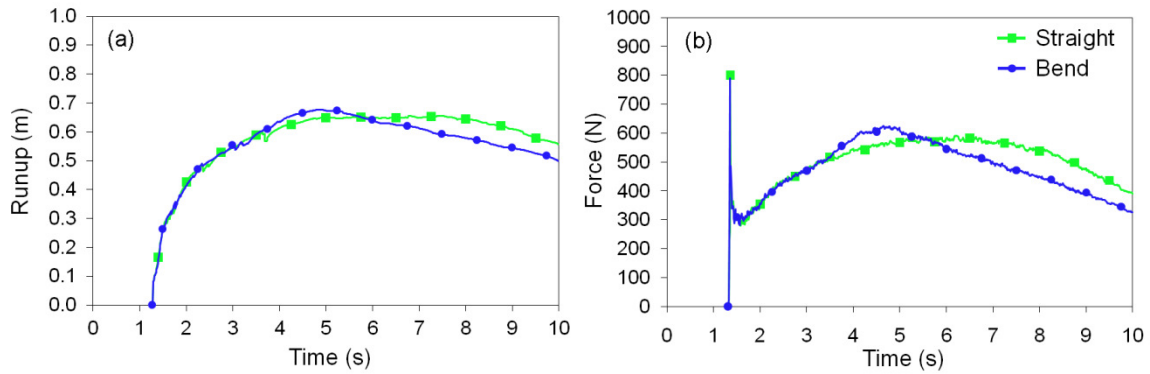


Figure 7.1. Time-history of (a) the bore runup and (b) the net base horizontal force acting streamwise on the column for straight (simulation D3-L2) and bend (simulation D2-85) reservoirs.

7.1.2 Water Surface Elevation and Velocity

To help explain the above-noted differences in the time-history of the runup and of the resulting force acting on the column, Figure 7.2 shows the water surface profile along the centerline of both reservoir geometries at various times after gate opening. [Note that, for

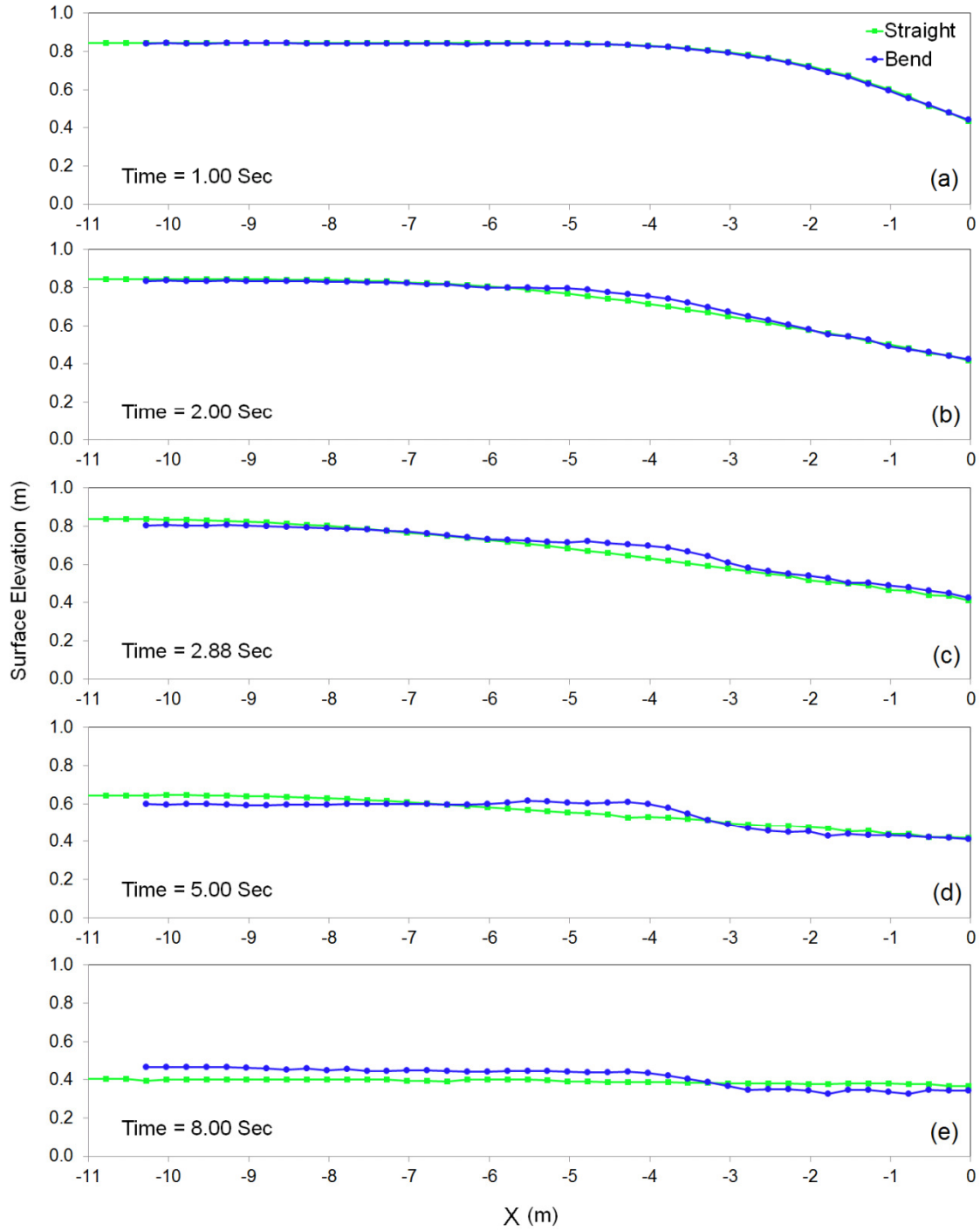
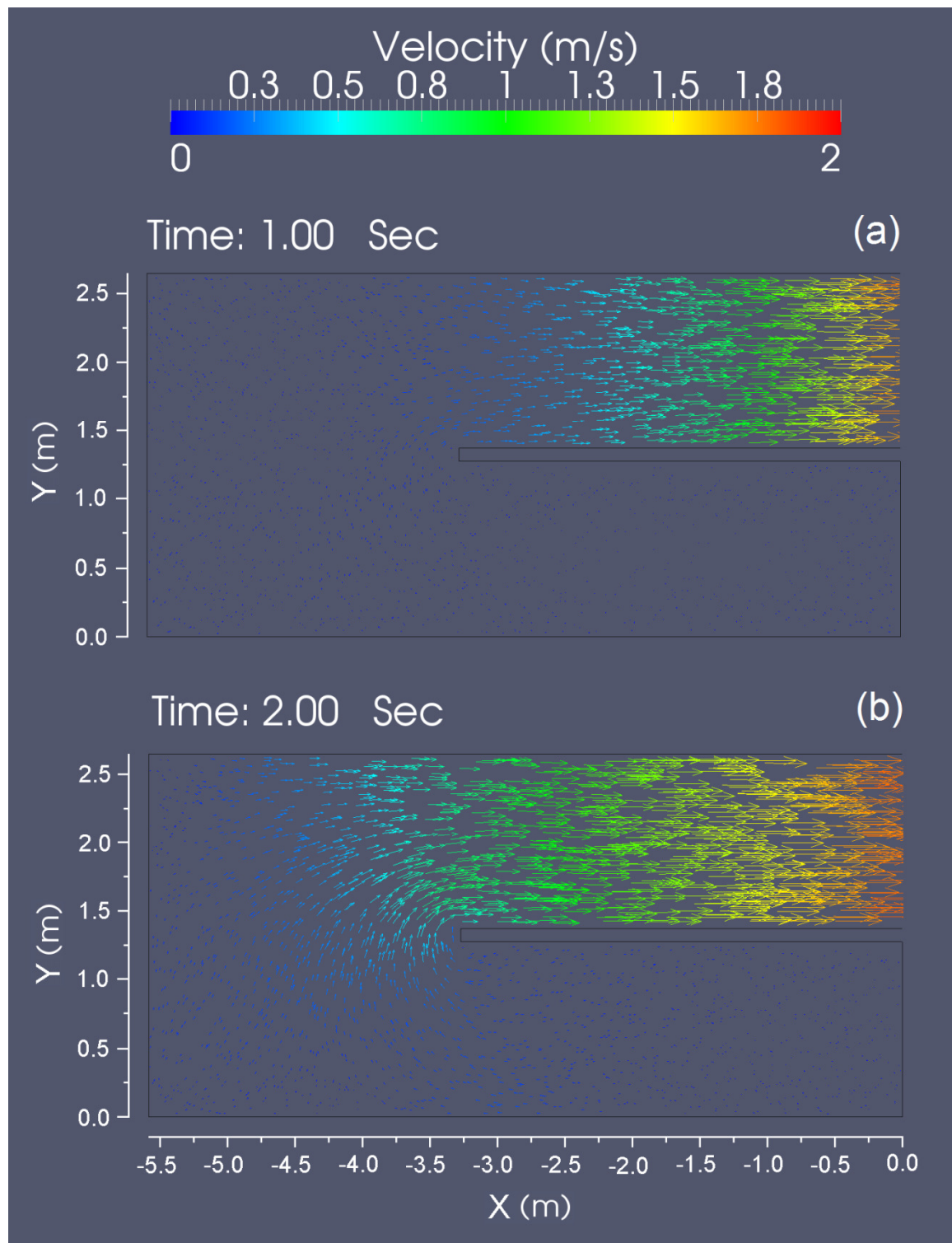


Figure 7.2. Water-surface profile along the centerline of the bended (simulation D3-L2) and straight (simulation D2-85) reservoirs (a), 1.00, (b) 2.00, (c) 2.88, (d) 5.00, and (e) 8.00 seconds after gate opening.

all figures in which an X axis is displayed, the origin corresponds to the downstream end of the reservoir (i.e. the location of the gate)]. Initially, up until the negative dam-break wave reaches the upstream end of the partition wall in the bend reservoir ($X = -3.28\text{m}$), the water surface profile is identical (Figure 7.2-a). Beyond this point, the water depth at $X \approx -4.00\text{m}$ remains greater in the case of the bend reservoir (e.g. Figure 7.2-b). By comparing the water surface profiles in Figure 7.2-c and Figure 7.2-d, it can be deduced that the bend reservoir initially empties more rapidly, as the water depth for $X < -6.00\text{m}$ is lower. However, as time advances, the upstream section of the straight reservoir ($X < -3.28\text{m}$) eventually empties quicker than in the case of the bend reservoir (Figure 7.2-e). Figure 7.2-c shows that the water surface in the bend reservoir drops at $X \approx -6.00\text{m}$ but then increases again at $X \approx -4.00\text{m}$. The latter shows that, as the flow enters the channel ‘test’ section, the constriction in the flow width from 2.70 to 1.30m (see Figure 4.2) forces the water level to increase at this location.

To better visualize the impact of this constriction in the flow width, Figure 7.3 displays, in a plane close to the flume’s bottom, the velocity field inside the bend reservoir at times corresponding to those considered in Figure 7.2. When the negative dam-break wave reaches the upstream end of the partition wall at $X = -3.28\text{m}$ (Figure 7.3-b), the effective width of the dam-break flow becomes greater than 1.30m. Hence, the discharge increases suddenly and because of the width constriction along the centerline of the reservoir, the flow accelerates as it enters the straight test section. This flow acceleration is also apparent from the sudden drop in the water surface elevation near this location (Figure 7.2-c). Up to ~5 seconds into the simulation, when compared to the straight geometry, the volume of water in the bend reservoir has decreased more rapidly. From this point in time, it is believed that the combined head loss, attributed to the initial flow expansion, the sharp 180° bend, and the eventual flow constriction seen in Figure 7.3-d, governs the flow supply, rather than the remaining potential energy of the water that remains in the reservoir. As a result, the flow exiting the reservoir from there on is shallower (Figure 7.2-e) and slower (Figure 7.4-b).



For caption see following page.

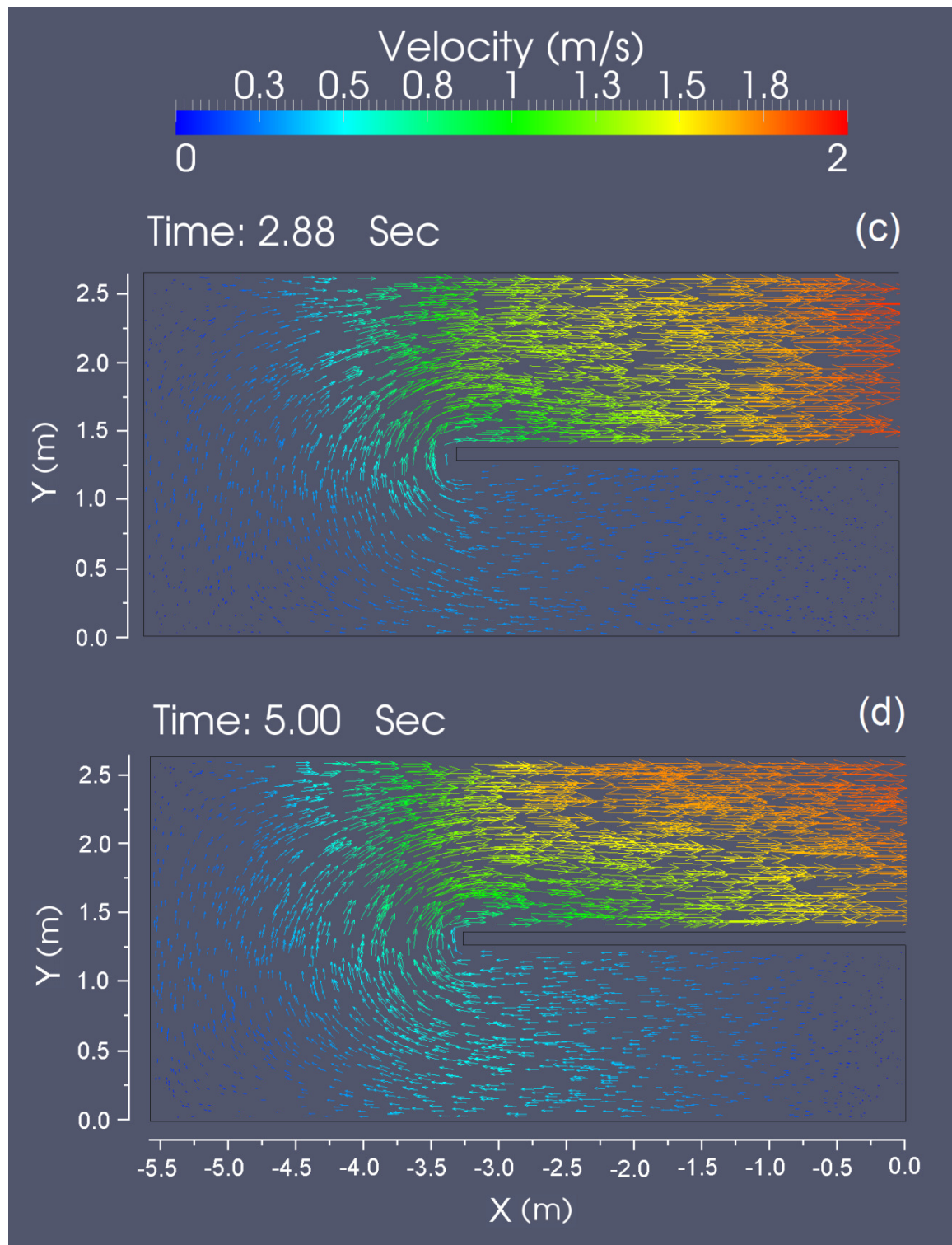


Figure 7.3. Plan view of velocity field inside the bended reservoir (a) 1.00, (b) 2.00, (c) 2.88, and (d) 5.00 seconds after gate opening (simulation D2-85).

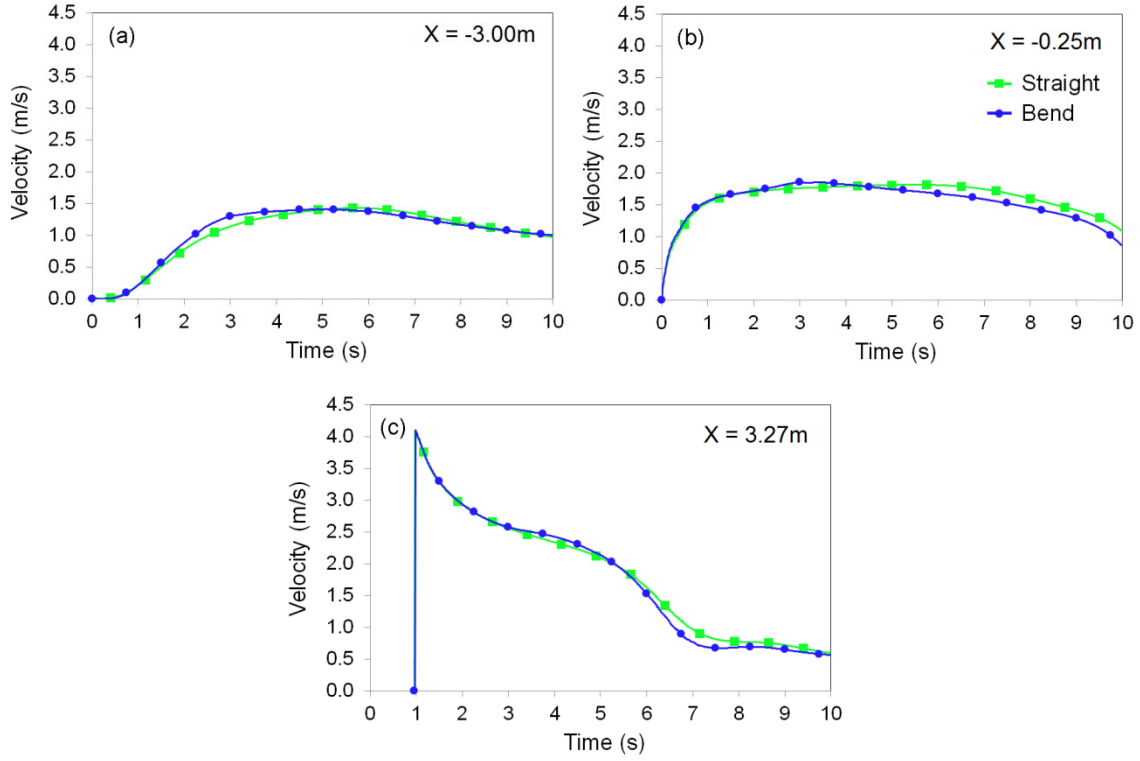


Figure 7.4. Time-history of the depth-averaged flow velocity (a) -3.00m, (b) -0.25m, and (c) 3.27m from the gate for straight (simulation D3-L2) and bended (simulation D2-85) reservoirs.

Figure 7.4 shows, for both reservoir geometries, the time-history of the depth-averaged streamwise velocity 3.00 and 0.25m upstream of the gate (i.e. inside the reservoir), as well as 3.27m downstream of the gate. The depth-averaged velocity (v_a) is computed as follows:

$$[7.1] \quad v_a = \sum_{h=0}^h v \frac{\Delta z}{h}$$

where h is the water depth, v is the streamwise flow velocity at the middle of a tributary depth Δz approximated according to Eq. 5.7. For the bend reservoir, it can be seen from Figure 7.4 that, at all locations shown, the velocity momentarily exceeds that for the straight reservoir case. This acceleration in the bore velocity, due to the previously discussed wider dam-break flow just upstream of the narrower test section, would explain

the higher runup of water on the column shown in Figure 7.1-a. Consequently, a higher runup induces a higher runup force (Figure 7.1-b). At later times however, as more water remains in the straight reservoir and since the latter induces less head losses, the corresponding flow velocity is greater. This explains the more sustained runup of water and the resulting force on the column.

Figure 7.5 shows the time-history of discharge from the reservoir near the location of the gate ($X = -0.25\text{m}$). The discharge is computed as the product of the depth-averaged flow velocity (v_a) and the rectangular cross-sectional flow area ($W \times h$). The observed behaviour is very similar to that for the velocity (Figure 7.4-b). For both geometries, the sudden increase in discharge after gate opening is essentially identical. In the case of the bend reservoir, the discharge momentarily increases as the width of the dam-break flow expands (e.g. Figure 7.3-b). Once its peak is reached (3.42 seconds), the discharge then decreases at an approximately constant rate until the end of the simulation. In the case of the straight reservoir, the discharge increases at a low but constant rate after the sudden rise. It is only much later in the simulation (5.89 seconds) that it begins to decrease at an approximately constant rate as well.

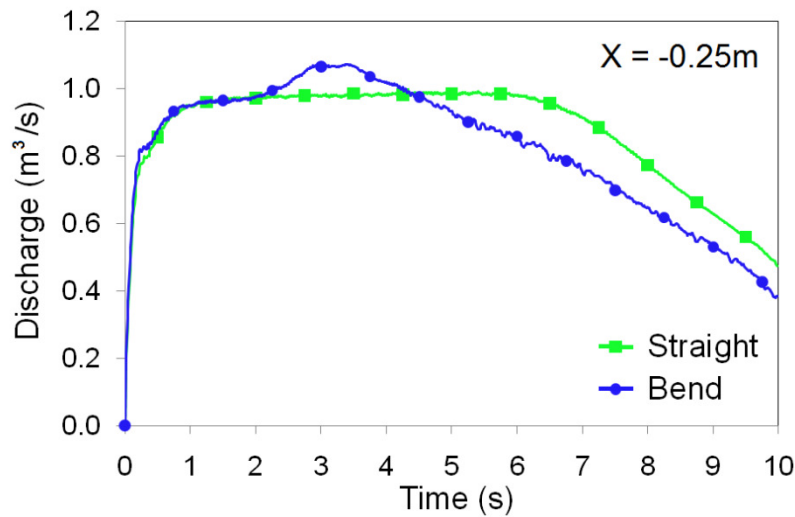


Figure 7.5. Time-history of the discharge calculated 0.25m upstream from the gate for straight (simulation D3-L2) and bended (simulation D2-85) reservoirs.

7.1.3 Cross-Waves Formation

As discussed in Section 6.1.1, diagonal cross-waves formed in the physical experiments at the entrance of the test section. However these are not reproduced by the SPH numerical model (see Figure 6.2 and Figure 6.3). Also, in contrast to the experimental results, a gradual decrease in the numerical time-history of the pressure at various elevations on the column was observed towards the end of the simulations (e.g. Figure 6.19-c). Although not explicitly mentioned by either Nouri *et al.* [2010] or Al-Faesly *et al.* [2012], it is suspected that additional inflow was provided during the physical experiments by the pumps supplying water to the flume. According to the layout of this laboratory flume (see Nouri *et al.* [2010]), the inflow provided by the pumps during the reservoir-filling stage comes from a 2.30m long opening in the left lateral wall of the flume (Figure 7.6-b). As no additional inflow was considered in the SPH simulations, the resulting hydraulic bores were of slightly shorter duration. In his numerical simulations using grid-based FVM with VOF, Zhihui Ma also noticed this discrepancy. As the software package that Ma used (Flow-3D) allows open boundaries with inflow, he adjusted the ‘inflow’ value such that his numerical results matched the experimental ones at times beyond 10 seconds. Depending on the experimental run being simulated, the inflow values determined in this way varied between 0.2 and 0.3m³/s. Figure 7.6 clearly shows the effect of this lateral inflow on the formation of the cross-waves in the simulations performed using FVM with VOF. When there is no inflow (Figure 7.6-a), the flow around the sharp 180° bend induces higher outward velocities, thereby creating a change in the water surface elevation next to the left lateral wall in the upstream portion of the test section. As shown in Figure 7.3-d, these higher velocities are also observed in the SPH simulations. On the other hand, when there is lateral inflow upstream of the test section (Figure 7.6-b), water from opposite transversal directions (along the Y axis) accelerates and converges inside the test section, perhaps resulting in the formation of diagonal cross-waves.

Alternatively, these cross-waves observed in the physical experiments could be the result of shock waves induced by lateral disturbances to the supercritical dam-break flow (Dr. Colin Rennie, personal communication, 2012).

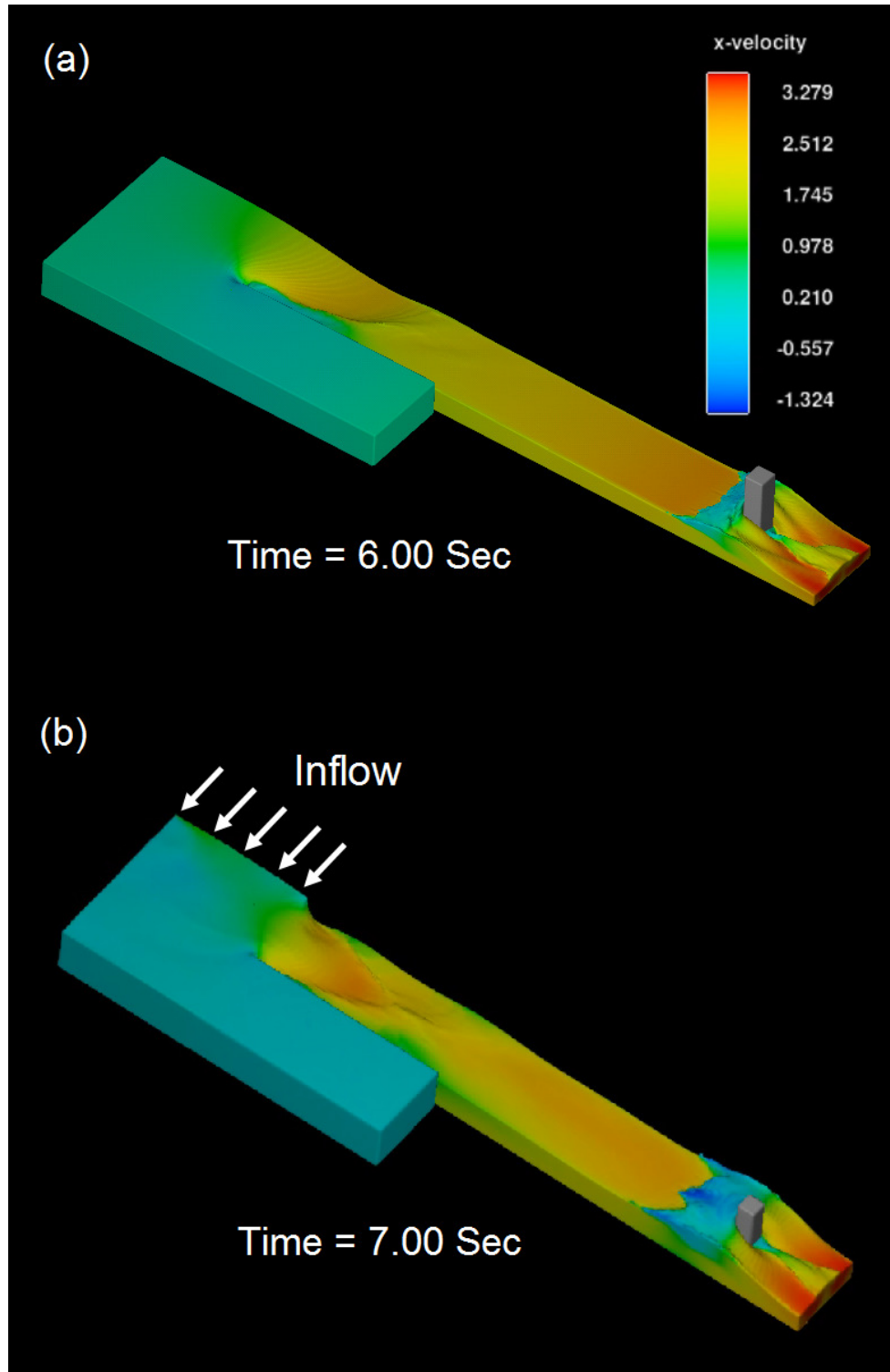


Figure 7.6. Effect of additional lateral inflow on the formation of diagonal cross-waves at the entrance of the test section ($h_u = 0.85\text{m}$). Numerical results obtained using FVM with VOF courtesy of Zhihui Ma.

7.1.4 Discussion

The numerical tests performed in this section clearly demonstrate that the unique (bend) geometry of the reservoir unit of the experimental flume in this study has an impact on the flow entering the flume's test section. Although the initial stages of the bore-structure interaction is the same for both the 'straight' and 'bend' geometries, the bend reservoir induces a higher but less sustained runup force. Also, it seems that the diagonal cross-waves at the entrance of the test channel in the physical experiments are largely due to opposite 'transverse-directed' currents that converge together.

7.2. Influence of Channel Width

In order to examine the flume 'wall' effect on the bore-structure interaction, the width of the test section, was increased numerically. Considering the computational domain with a straight reservoir shown in Figure 4.14 with $L = 6.0\text{m}$, the width (W) was doubled from 1.3 to 2.6m.

7.2.1 Runup and Force

Figure 7.7 shows the resulting influence that the flow width has on both wave runup and the net base horizontal force acting streamwise on the column structure. As the dam-break flow in a straight channel is generally two-dimensional, the resulting force on the column is essentially the same for both widths, up until wall effects occur after approximately 2 seconds in the case of $W = 1.3\text{m}$. However, it is noted that the peak in the time-history of the force at initial impact is greater for $W = 2.6\text{m}$. Since the depth of the bore-front and its velocity are the same for both widths (Figure 7.9), and given that a numerical output frequency of 1000Hz was considered from 1 to 2 seconds for the simulation with $W = 1.3\text{m}$, it is not known with certitude why such a difference in the initial impact force occurs. At the moment of impact, the latter is believed to be caused by a slightly different distribution of water particles within the sampling volumes. This in turn would result in a different kernel approximation of the pressure. [The approach taken for determining the resulting force acting on the column, by the integration of the pressure, is described in Section 5.2.1]. After ~2 seconds, the runup on the column

(Figure 7.7-a) is higher for $W = 1.3\text{m}$ than if is for $W = 2.6\text{m}$. Consequently, the runup force is also higher (Figure 7.7-b).

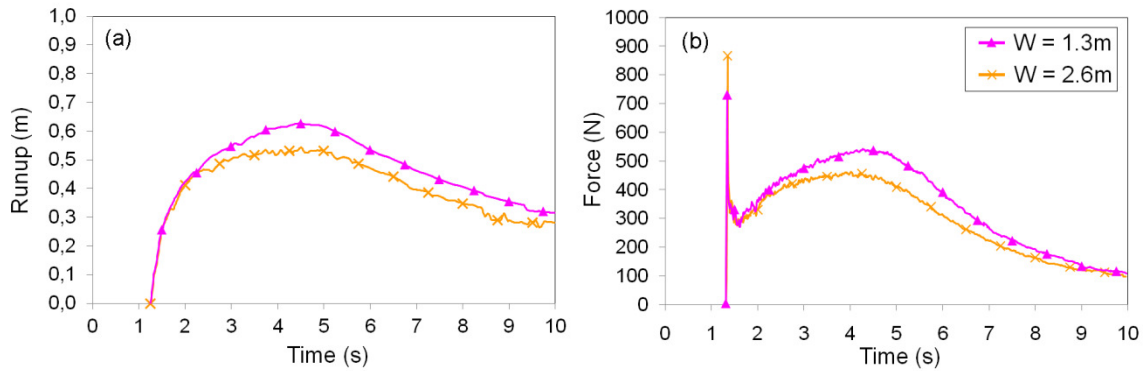
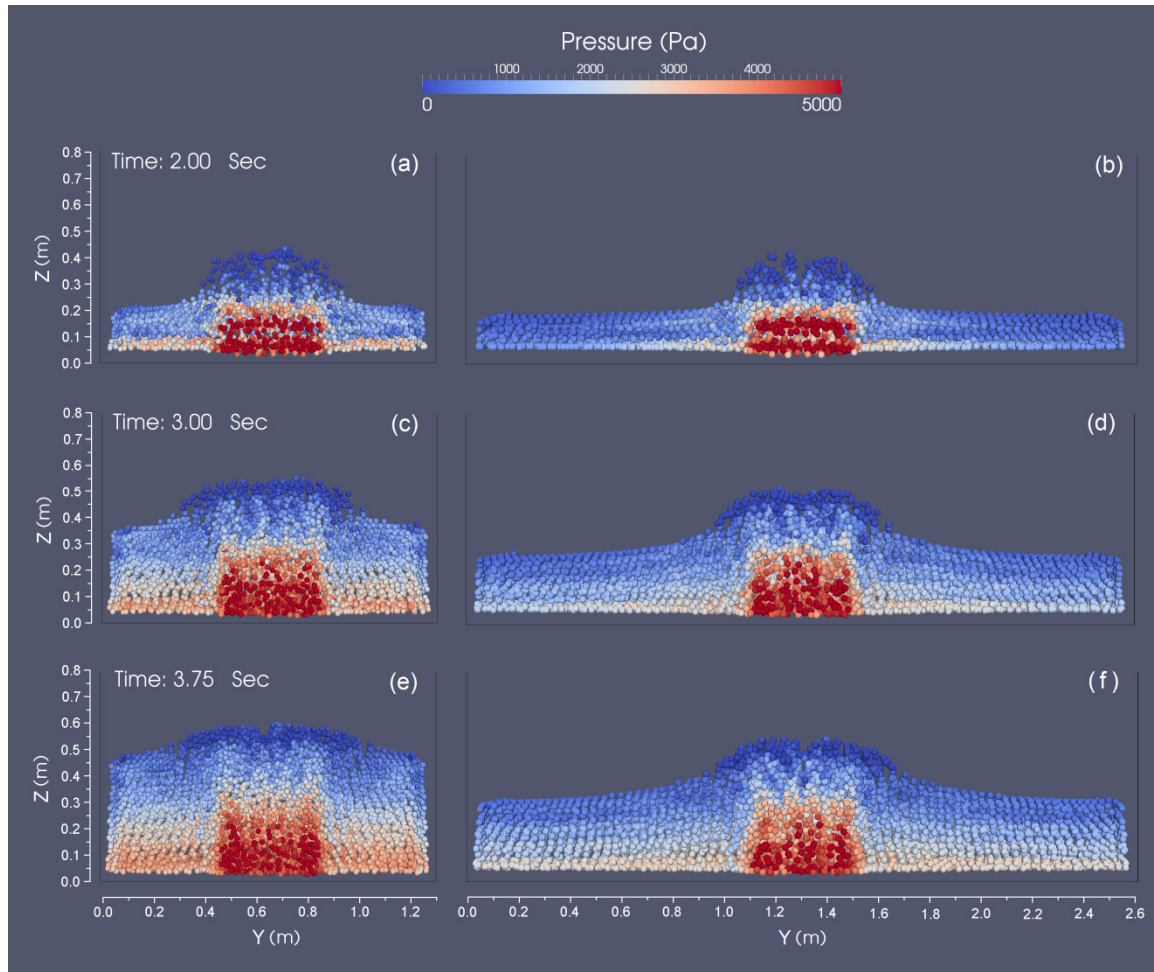


Figure 7.7. Time-history of (a) the bore runup and (b) the net base horizontal force acting streamwise on the column for channel widths (W) of 1.3 (simulation D3-L1) and 2.6m (simulation D3-WIDE).

7.2.2 Water Surface Elevation and Velocity

Providing cross-sectional views of the test section just upstream of the column structure ($X = 4.67\text{m}$), Figure 7.8 shows the influence of the flow width on the water surface elevation at this location. At 2 seconds (Figure 7.8-a and Figure 7.8-b), the water surface elevation in front of the column (i.e. the runup) is approximately the same for both widths. As time advances, the obstruction of the column in the narrower channel causes the water level across the flume to increase (Figure 7.8-c). On the other hand, in the case of the wider channel, the water level away from the channel's centerline does not seem to be significantly affected by the presence of the column (e.g. Figure 7.8-h).

Looking at the time-history of the water surface elevation and the depth-averaged flow velocity 1.77 and 3.77m downstream of the gate, Figure 7.9-c shows that the main difference between the two widths is that, in the case of $W = 2.6\text{m}$, the smaller flow obstruction caused by the column does not induce a wave that propagates upstream once the maximum runup is reached. In addition, the sudden velocity reduction associated with the propagation of such wave (i.e. the increase in water depth) is clearly noticeable in the case of $W = 1.3\text{m}$ (e.g. Figure 7.9-b).



For caption see following page.

To investigate further the nature of this obstruction-induced wave observed in the physical experiments ($W = 1.3\text{m}$), simulation D2-85 is considered and the Froude number at locations sufficiently away from one another ($X = 1.77\text{m}$ and $X = 3.77\text{m}$) is calculated as:

$$[7.2] \quad Fr = \frac{v_a}{\sqrt{gh}}$$

where v_a is the depth-averaged flow velocity calculated with Eq. 7.1 and h is the flow depth. The time-history of the Froude number at these locations is shown in Figure 7.10,

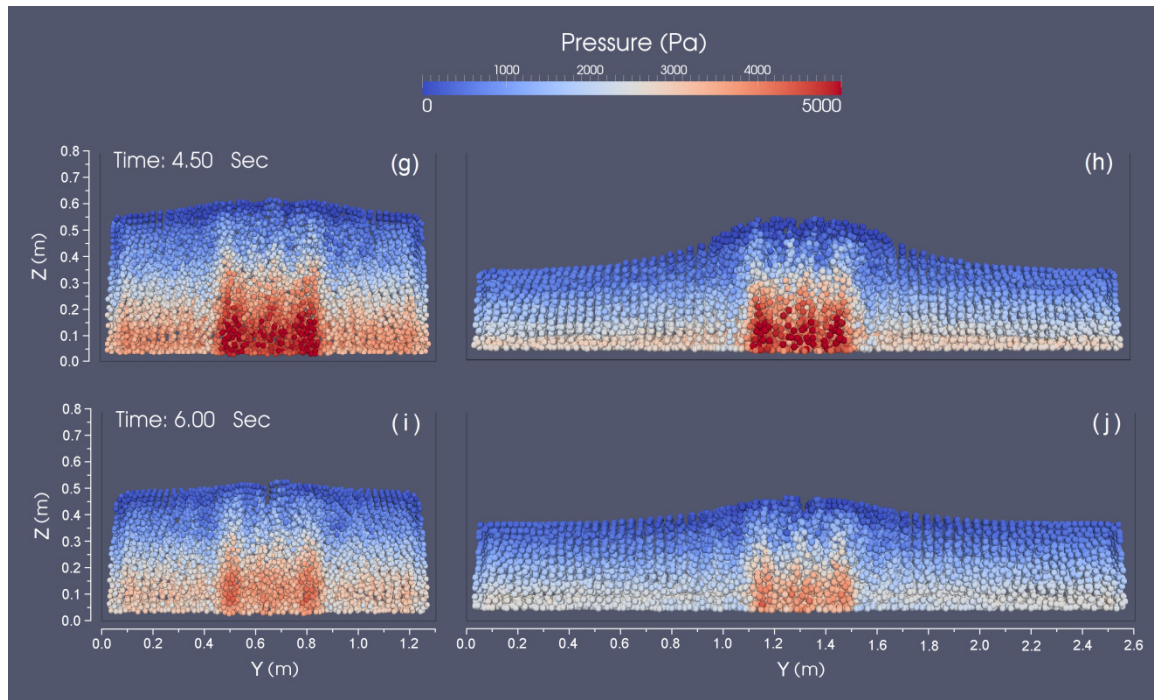


Figure 7.8. Cross-sectional view of bore-structure interaction just upstream of column (a, b) 2.00, (c, d) 3.00, (e, f) 3.75, (g, h) 4.50, and (i, j) 6.00 seconds after gate opening for channel widths (W) of 1.3 (left, simulation D3-L1) and 2.6m (right, simulation D3-WIDE).

where a red horizontal line highlights the critical value of unity. At impact with the column structure, as the velocity is high and the flow depth shallow, the flow is considerably supercritical ($Fr > 1$). Shortly after, as the flow depth associated with the dam-break increases, the related velocity decreases and so does the Froude number. At 4.73 seconds after gate opening, just moments after the maximum runup is reached, the flow at $X = 3.77\text{m}$ becomes subcritical as the obstruction-induced wave propagates upstream. At $X = 1.77\text{m}$, the flow becomes subcritical at 6.44 seconds, due to the later arrival of this wave. Hence, according to this numerical simulation (D2-85), there is a small possibility for a hydraulic jump to occur, since at some instant the flow is supercritical upstream while subcritical further downstream. However, since the upstream Froude number is close to unity, this jump would be relatively weak and characterized by simple undulations of the free-surface [Chow, 1973]. Based on this, the wave observed in the physical experiment (Figure 6.4) could possibly be a hydraulic jump, as previously

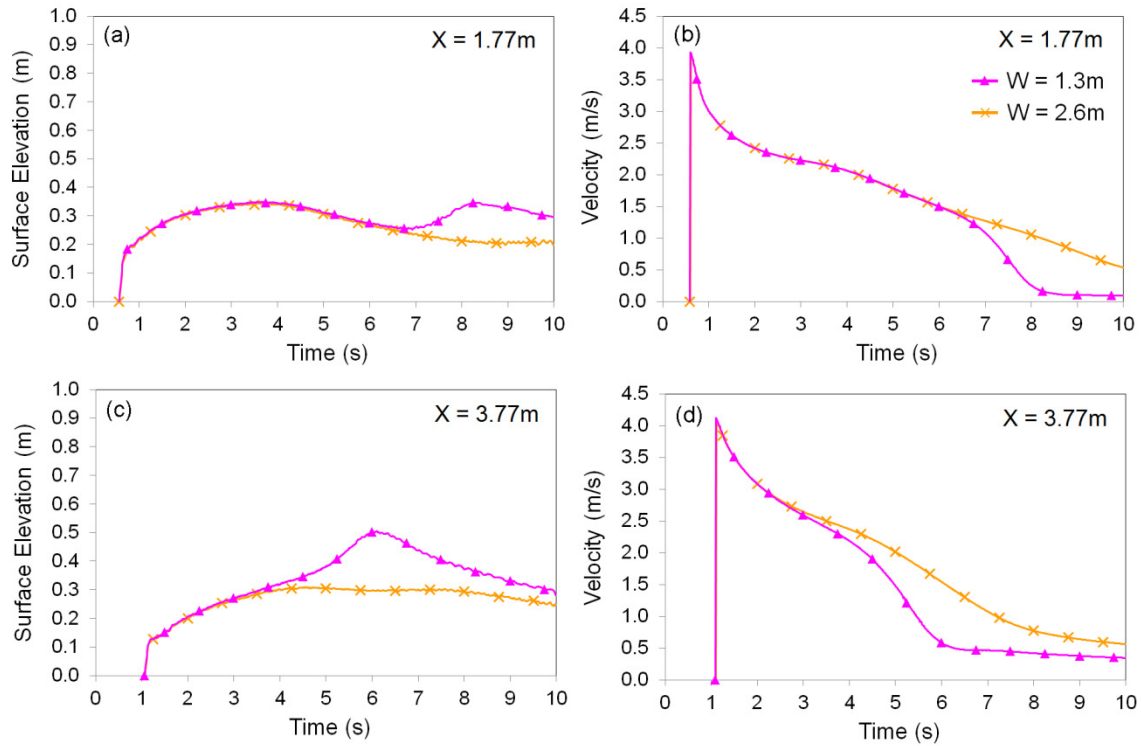


Figure 7.9. Time-history of the water-surface elevations (left) and depth-averaged flow velocities (right) for channel widths (W) of 1.3 (simulation D3-L1) and 2.6m (simulation D3-WIDE) at locations (a, b) 1.77m, and (c, d) 3.77m from the gate.

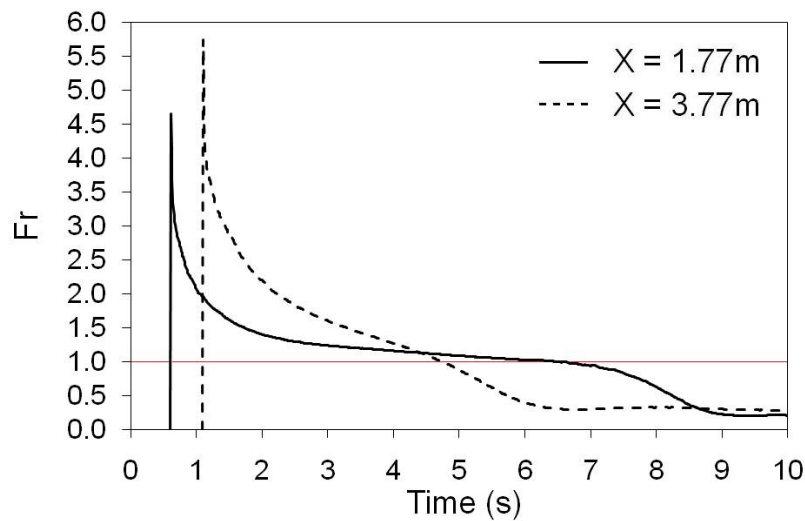


Figure 7.10. Froude number 1.77 and 3.77m downstream of the gate for simulation D2-

suspected in Section 6.1.1. Because of the visible turbulence however, this experimental hydraulic jump seems stronger than a simple undular jump, according to the definition of Chow.

Figure 6.7-b shows that the experimental obstruction-induced wave exhibits delay in comparison the numerical one. As the two waves had approximately the same celerity, it was concluded that this delay was a consequence of the times at which their propagation was initiated. To investigate this discrepancy, Figure 7.11 shows the time-history of the water surface elevation 1.77m downstream of the gate for simulations D3-L1 and D3-HIGH, for which the initial inter-particle spacing is 0.035 and 0.025m, respectively.

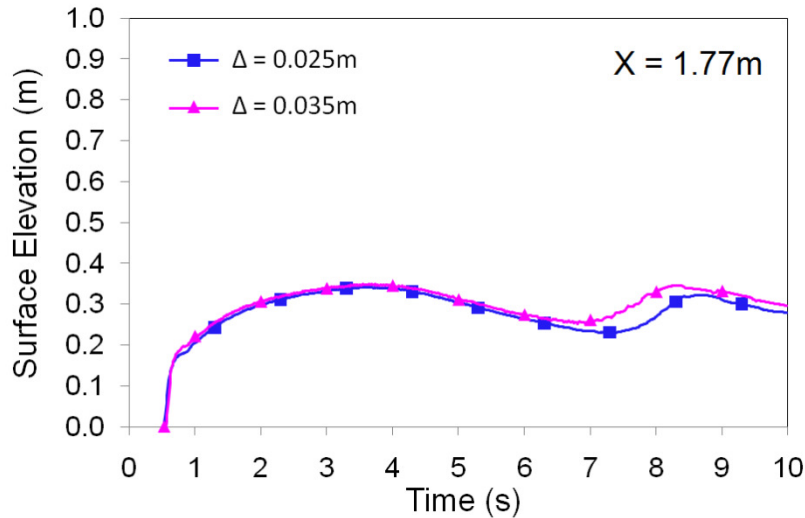


Figure 7.11. Time-history of the water-surface elevation 1.77m downstream from the gate for numerical simulations (D3-L1 and D3-HIGH) with different initial inter-particle spacings Δ .

It can be observed that, as in the physical experiments, the arrival of the wave at this location occurs later in the case of the simulation with the higher particle resolution. Figure 7.12 demonstrates that, as the void between fluid particles and repulsive boundary particles decreases with a higher particle resolution, the effective flow area at the location of the column increases. Consequently, the flow obstruction is less significant and the aforementioned wave forms later. [Note that, no advanced post-visualization was

performed in Figure 7.12; the fluid particles are simply coloured blue and repulsive boundary particles red].

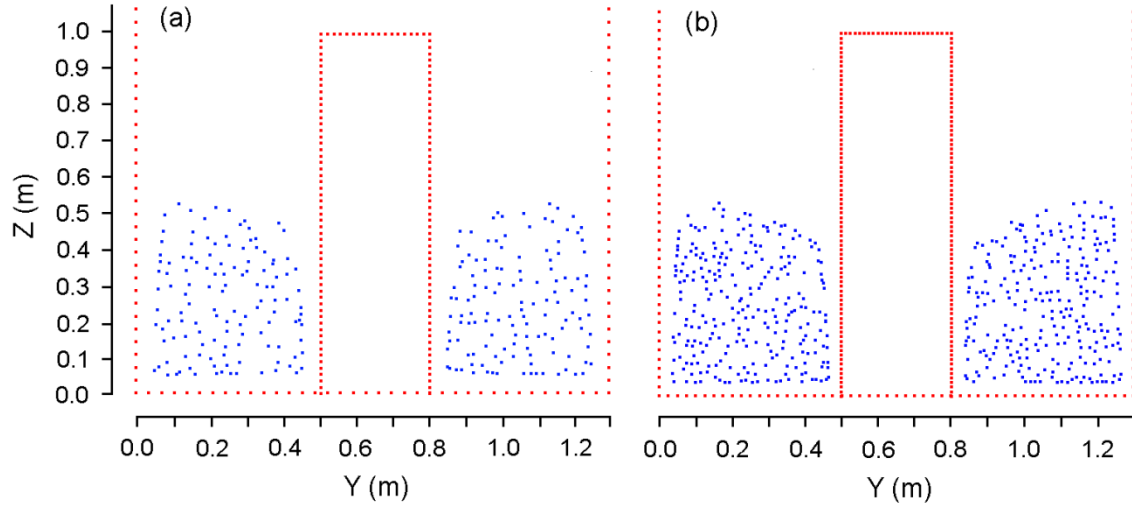


Figure 7.12. Cross-sectional view at $X = 4.92\text{m}$ demonstrating the reduction of the effective flow area caused by the void of fluid particles near solid boundaries for the initial inter-particle spacing Δ of (a) 0.035m (simulation D3-L1) and (b) 0.025m (simulation D3-HIGH).

7.2.3 Bore-Structure Interaction

Figure 7.13 shows the numerical time-history of the net base horizontal force acting on the column for simulation D3-L1 ($L = 6.0\text{m}$ and $W = 1.3\text{m}$). The red points correspond to key moments in the bore-structure interaction for which velocity vectors in an X-Z plane just upstream of the column are shown sequentially in Figure 7.14. To further illustrate the following description of the bore-structure interaction, corresponding still frames of the physical experiments of Árnason [2005] are also presented in Figure 7.15.

At initial impact (Figure 7.14-a), the shallow bore-front impacts the column at very high velocity. As the momentum of the water is either suddenly reduced to zero or redirected upwards, a large impulsive force is exerted on the column. Shortly after

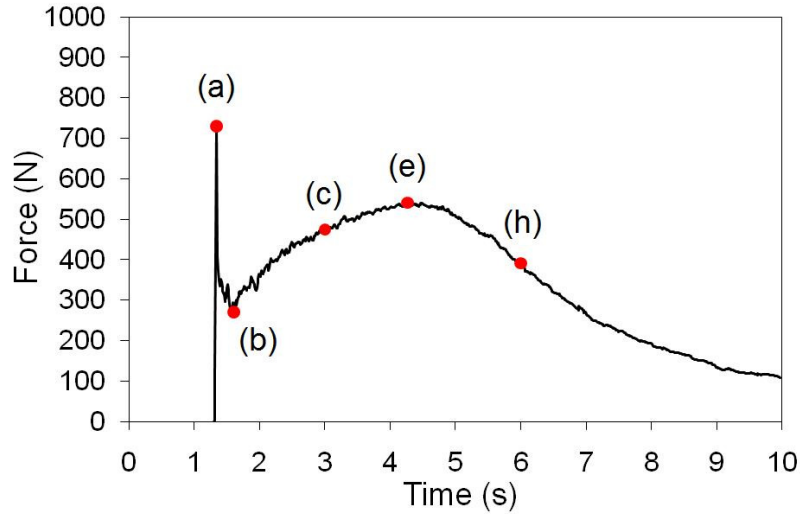
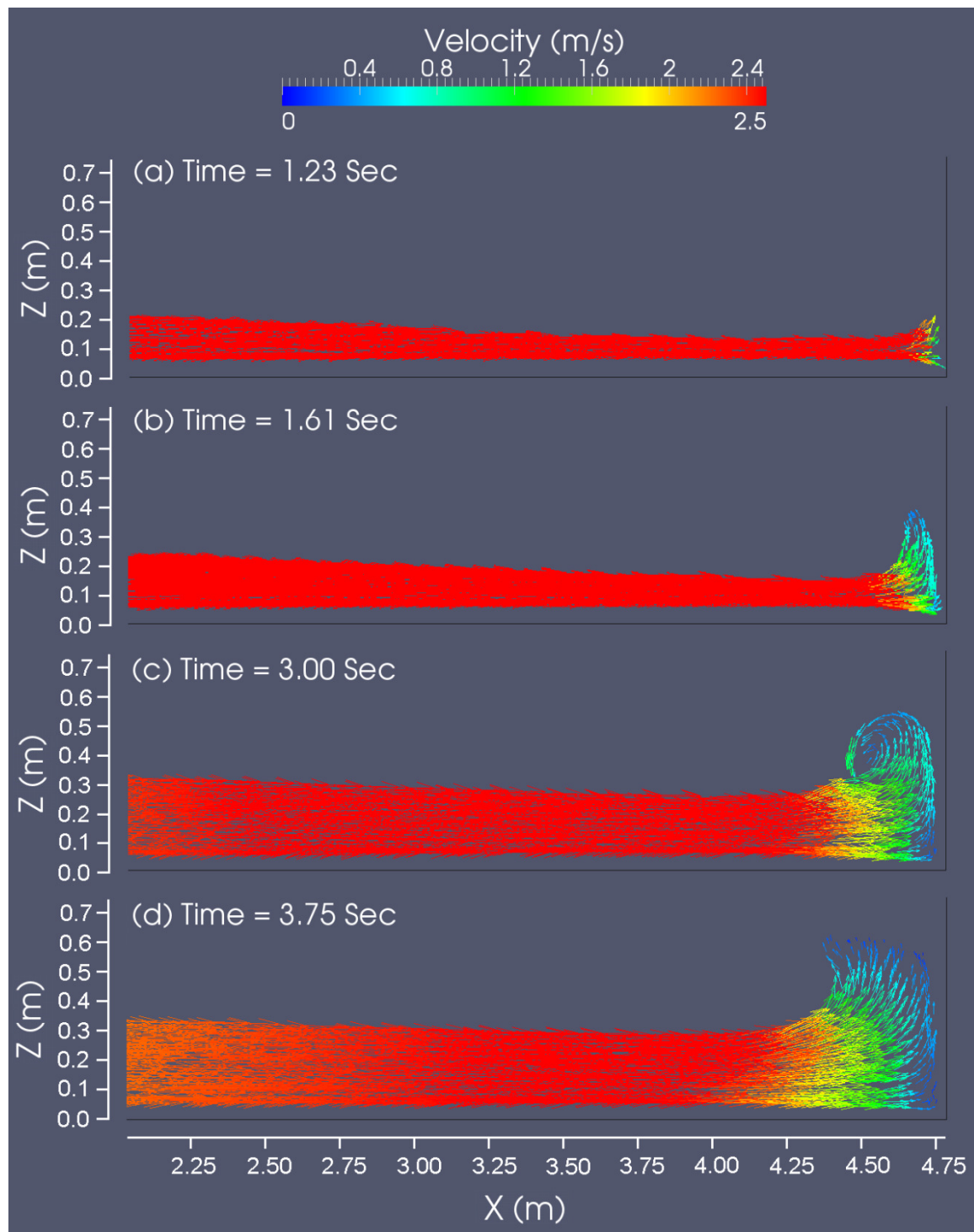
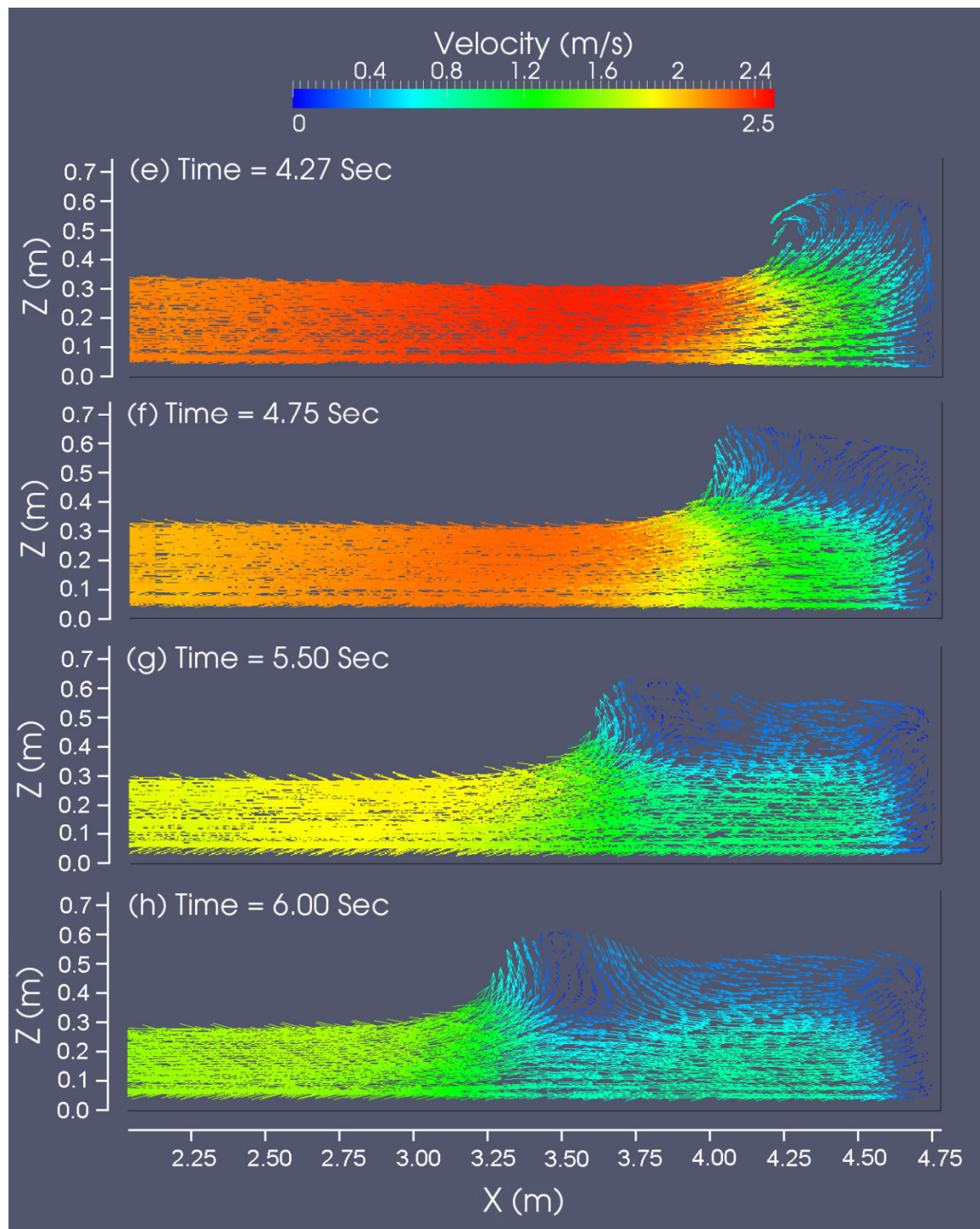


Figure 7.13. Numerical time-history of the net base horizontal force acting on the column with points associated to Figure 7.14 (simulation D3-L1).

(Figure 7.14-b), the resulting force on the column drops significantly since, as a region of lower velocity begins to form directly upstream of the column, the incoming water flows around the column with a less drastic change in its momentum. Figure 7.15-a shows a still frame from the experiments of Árnason at a similar instant. Beyond this point, although the velocity of the incoming flow decreases, its depth increases (Figure 7.14-c), resulting in a gradually increasing force acting on the column. [Note that this decrease of velocity is hardly noticeable in Figure 7.14, as the velocity scale was selected in order to improve the visibility of velocity gradients at locations of low velocity]. As the runup of water on the column increases, a surface roller forms (Figure 7.14-c). This surface roller is characterized by water being redirected upstream and falling on the incoming flow. Figure 7.15-b shows a still frame showing such flow feature in the experiments of Árnason. As the water level increases further, this surface roller expands towards the walls of the flume, as seen in the transversal cross-sections taken just upstream of the column shown in Figure 7.8. At the moment when this surface roller reaches the walls, as seen in Figure 7.15-c for the experiments of Árnason, the second peak in the force (i.e. the runup force) occurs (Figure 7.14-e). Furthermore, because of the flow obstruction caused by the column, as discussed previously a wave initiates its propagation upstream. With a closer inspection of the velocity field within this wave (Figure 7.14-h), the



For caption see page 196.



For caption see page 196.

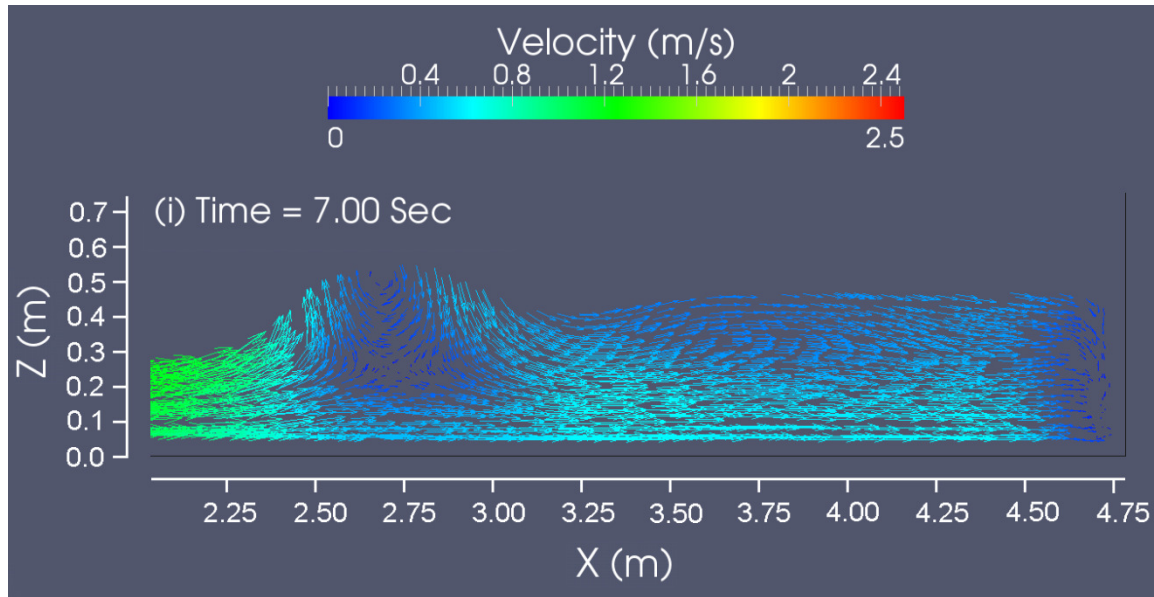


Figure 7.14. Velocity field upstream of column's center (a) 1.23, (b) 1.61, (c) 3.00, (d) 3.75, (e) 4.27, (f) 4.75, (g) 5.50, (h) 6.00, and (i) 7.00 seconds after gate opening for the channel width (W) of 1.3m (simulation D3-L1).

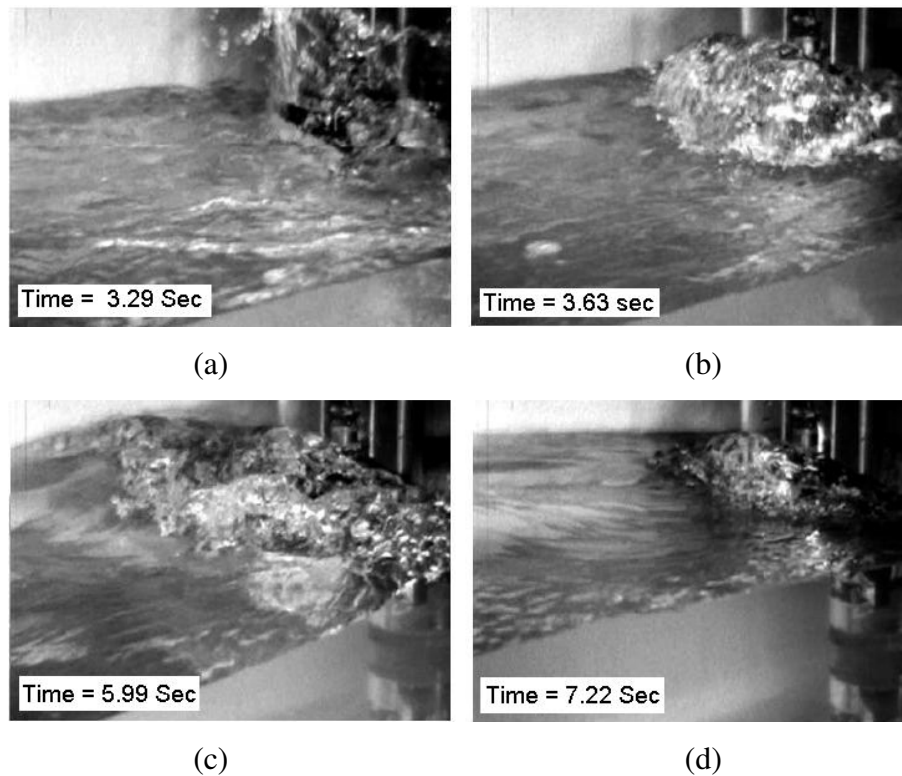


Figure 7.15. Still frames from the physical experiments of Árnason [2005] showing the interaction of a hydraulic bore ($h_u = 0.250\text{m}$) with a circular column.

following is observed. The upper portion of the incoming flow at the front of the wave is redirected upwards and slightly backwards as well, although no wave breaking occurs. The water particles then flow downwards and downstream on the back of the wave. In addition, weak clockwise orbital flow is observed at the very peak of the wave, bringing some water particles from the downstream side of the wave to the front (Figure 7.14-i). As this wave propagates upstream from the column, the resulting force on the latter gradually diminishes as a consequence of the falling water surface elevation and the lower flow velocity in front of it. This reduction in the water depth results in the majority of the velocity vectors next to column being directed downwards (e.g. Figure 7.14-i) compared to upwards, as in the earlier stages of the bore-structure interaction (e.g. Figure 7.14-e).

In a plane close to the channel's bottom, Figure 7.16 and Figure 7.17 show the numerical velocity in the X and Y directions, respectively, as the wake downstream of the column develops (simulation D2-85). As the hydraulic bore impacts the column, it divides into two streams (e.g. Figure 7.16-a). Because of the flow reflection caused by the flume walls, these streams are reoriented towards the centerline of the flume (Figure 7.17-a) and, as the components of the velocity in the Y direction are small compared to the ones in the X direction, they eventually reconnect at a distance much further downstream (not shown). With time, the reconnecting point gradually migrates upstream (e.g. Figure 7.16-b), until it reaches the back of the column (e.g. Figure 7.16-c). At that instant, some of the fluid particles flowing around the column flow back upstream behind the column, as shown by the negative X velocities in Figure 7.16-c. In addition, as the flow circulating around the column reconnects behind it with moderated Y velocity components with opposite directions (Figure 7.17-c), the water surface is forced upwards along the centerline of the flume. The latter can clearly be seen in Figure 6.3-e. Beyond this point in time, it seems that, as a void region develops, the water particles with very low velocities directly behind the column do not mix with the ones that are flowing around it (e.g. Figure 7.16-d).

With regards to the influence of the channel width on the formation of the downstream wake, Figure 7.18 shows (plan view) the evolution of the velocity field around the column in simulations D3-L1 and D3-WIDE, corresponding to W of 1.3 and

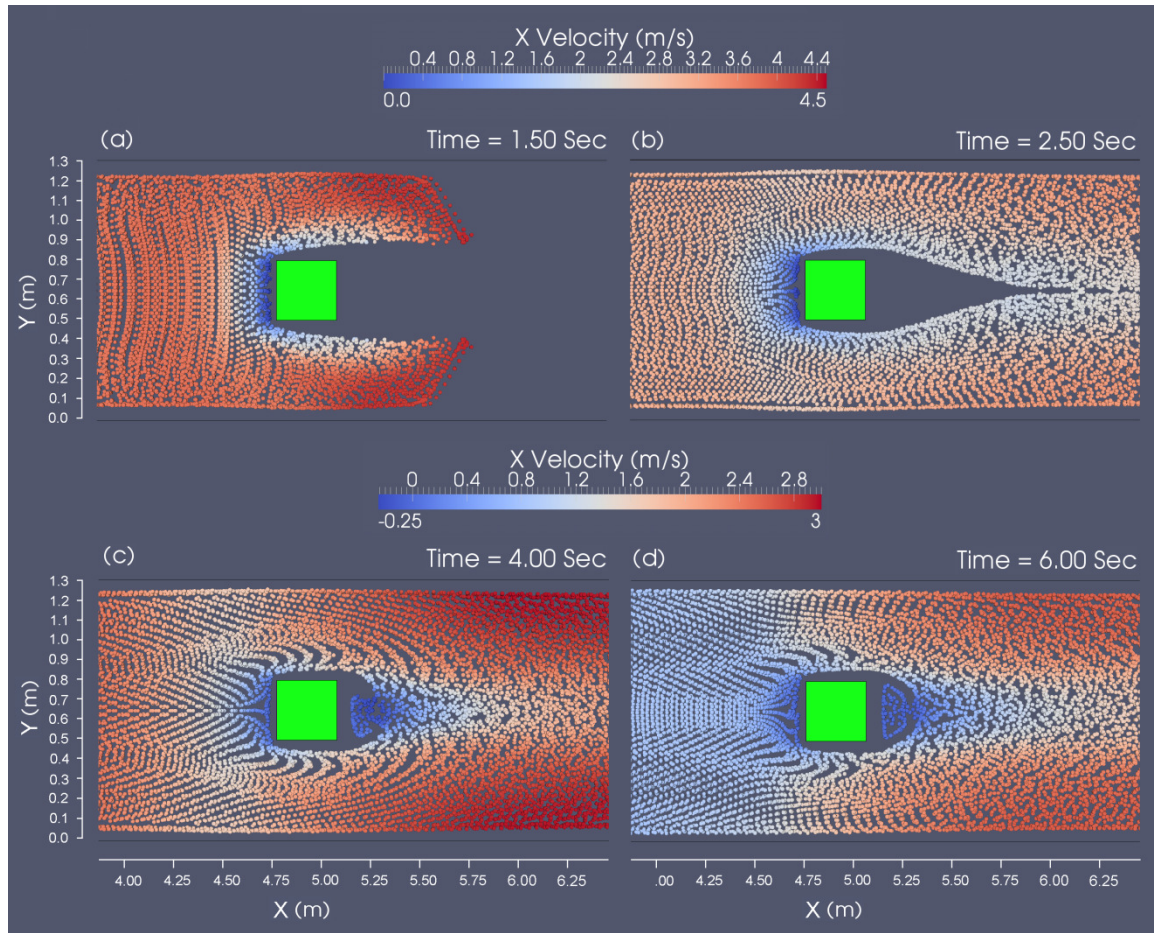


Figure 7.16. Plan views of the X velocity field in a plane close to the channel's bottom at (a) 1.50, (b) 2.50, (c) 4.00, and (d) 6.00 seconds after gate opening for simulation D2-85.

2.6m, respectively. In the case of $W = 2.6\text{m}$, it can be observed that the two streams flowing around each side of the column reconnect later in the simulation and further downstream. The latter can be explained by the fact that, as the bore impacts the column, the divided streams tends to flow away from the column towards the sides of the flume. Hence, in the case of the narrower flume, the reflection of these streams on the lateral walls is more pronounced and the streams reconnect sooner along the flume's centerline.

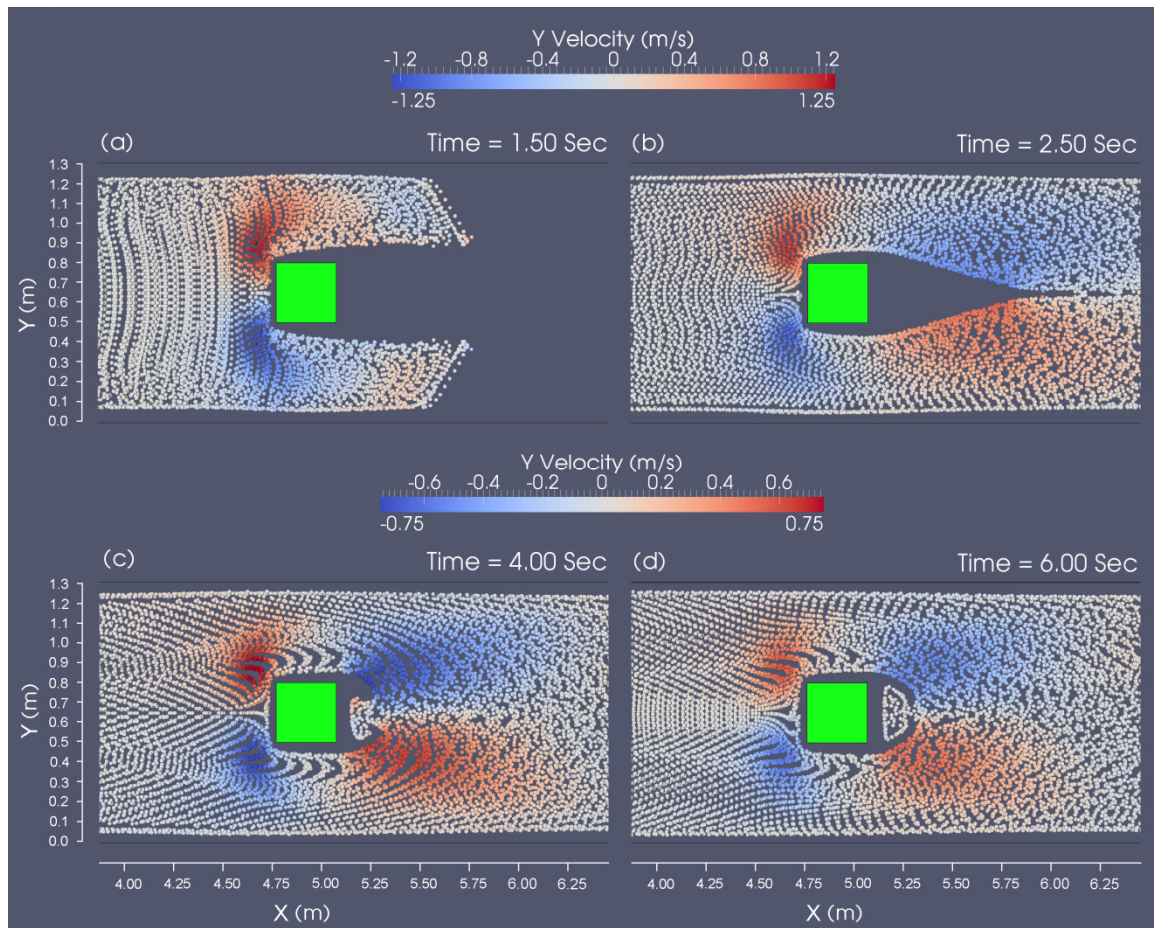


Figure 7.17. Plan views of the Y velocity field in a plane close to the channel's bottom at (a) 1.50, (b) 2.50, (c) 4.00, and (d) 6.00 seconds after gate opening for simulation D2-85.

7.2.4 Discussion

The numerical results presented in this section show that the width of the channel has a marked influence on the bore-structure interaction. However, this influence is mainly observed in the later stages of the interaction. The most important effect of the relatively narrow width considered in the actual physical experiments ($W = 1.30\text{m}$) is the obstruction of the flow field in the vicinity of the 0.3m wide column. As a result of the latter, the water depth at the front of the column reaches higher elevations, thereby exerting greater post-impact hydrodynamic forces on the column. Based on the increasing runup force with respect to the initial impact force shown in Figure 6.27, this would suggest that the greater the impoundment depth, the greater the influence of the channel width will be.

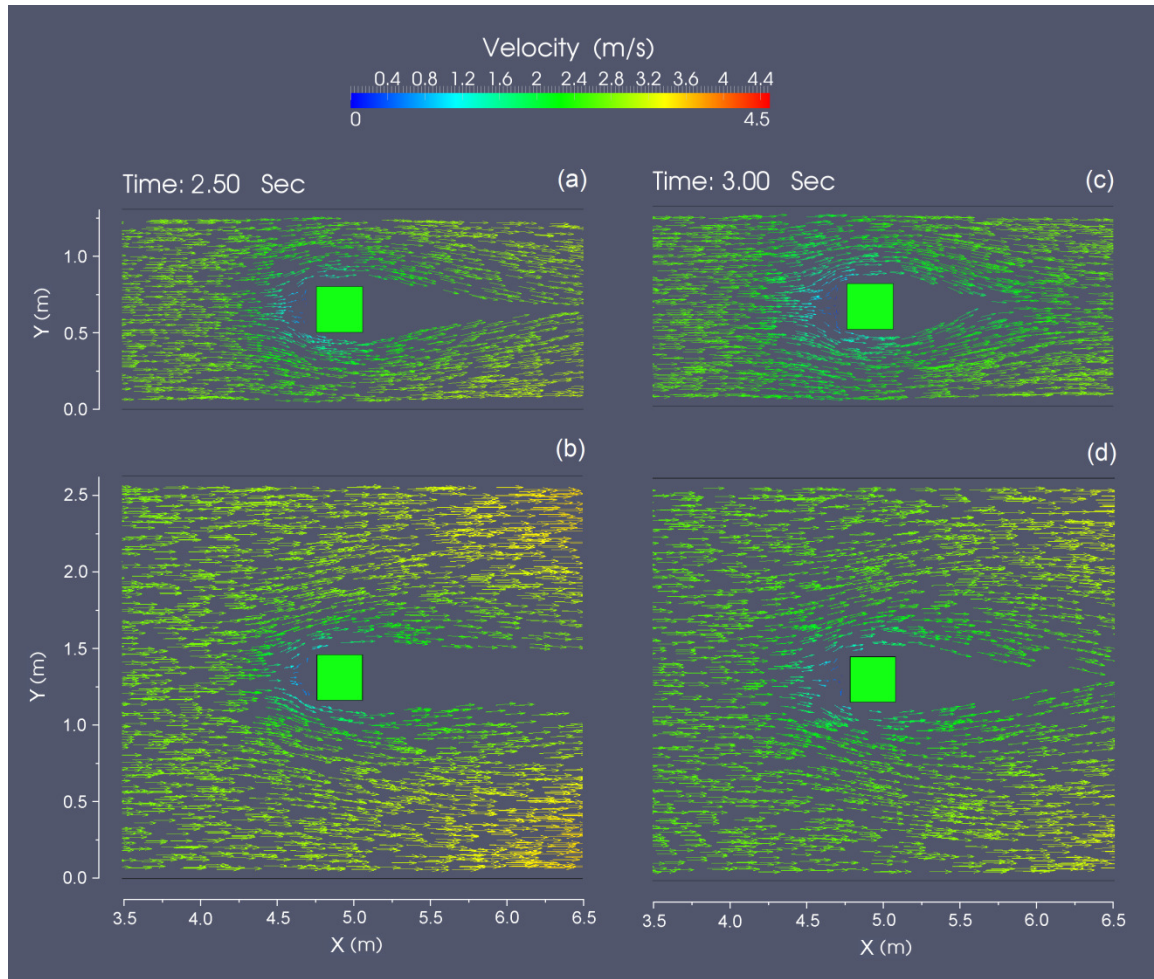


Figure 7.18. Plan view of velocity field around the column (a, b) 2.50 and (c, d) 3.00 seconds after gate opening for channel widths (W) of 1.3 (left, simulation D3-L1) and 2.6m (right, simulation D3-WIDE).

Also, it is obvious that the obstruction-induced wave observed in the physical experiments is due to the relatively narrow width of the test section, when compared to the width of the column structure. An analysis of the flow criticality for simulation D2-85 showed that, while the Froude number is close to unity, the flow upstream of this wave was supercritical yet subcritical downstream. Theoretically, a weak hydraulic jump could form but the numerical velocity field revealed that this wave was more like a simple wave propagating over an opposite current. On the other hand, based on the visible turbulence and air entrainment, such a wave in the video images of the physical experiments (e.g. Figure 6.4) is more indicative of a hydraulic jump. Due to the relatively

coarse resolution of the numerical simulations and the excessive numerical diffusion induced by the Riemann solver scheme (see Section 6.2.1), it is possible that the complex flow features of a hydraulic jump were not successfully reproduced in the numerical simulation.

Lastly, due to the reflection of the divided hydraulic bore from the flume's side walls, the development of the wake downstream of the column is also affected by the width of the test section. In the case of a narrower section, reflection off the flume's side walls of water flowing around the sides of the column is stronger than it would be for the case of a wider section. Consequently, the downstream wake forms earlier in the case of a narrower channel.

7.3. Influence of Reservoir Size

To examine the influence of reservoir size (i.e. its surface area) on the resulting hydrodynamic forces, a computational domain with a straight reservoir is considered (Figure 4.14). As was the case in the physical experiments, the flow width (W) is fixed at 1.3m, however the reservoir length (L) is varied (6.0, 11.2, and 15m). To further examine the influence of reservoir size, the numerical results are also compared with the empirical and analytical relationships that follow.

In their laboratory investigation on dam-break flow mechanics, Lauber and Hager [1998] examined, among other things, the effect of the reservoir length. One of their main conclusions was that the flow depth and velocity field close to the front of a dam-break wave is not affected by reservoir length if $L/h_u > 2$. Taking into account the length of the reservoir, Lauber and Hager formulated the following empirical relationships for estimating maximum flow depth and its occurrence time at any location downstream of the gate:

$$[7.3] \quad Z_M^* = \frac{4}{9} \left\{ 1 + [(\lambda^*)(X^*)^{-2/3}]^{-1} \right\}^{-5/4}$$

$$[7.4] \quad T_M^* = 1.7 [\lambda^* + (X^*)^{2/3}]$$

with the following dimensionless parameters:

$$[7.5] \quad X^* = X/h_u$$

$$[7.6] \quad Z^* = h/h_u$$

$$[7.7] \quad T^* = t(g/h_u)^{1/2}$$

$$[7.8] \quad V^* = v/(gh_u)^{1/2}$$

where X is the distance downstream from the gate, h is the flow depth, t is the time after gate opening, v is the mean flow velocity, and $\lambda^* = L/h_u$ is the relative length of the straight water reservoir used to generate the dam-break wave.

As one of the pioneers on the topic, Ritter [1892] provided an analytical solution for the dam-break problem in a dry horizontal straight channel of infinite length. He assumed hydrostatic pressure and the fluid to be inviscid with no bottom friction. Despite all these assumptions, physical experiments have shown that, except for the bore-front and during wave initiation, Ritter's solution can predict the general characteristics of the dam-break flow reasonably well [Lauber and Hager, 1998]. With the dimensionless parameters presented above, Ritter's solution takes the form:

$$[7.9] \quad Z^* = \left[\frac{1}{3} \left(2 - \frac{X^*}{T^*} \right) \right]^2$$

$$[7.10] \quad V^* = \frac{2}{3} \left(1 + \frac{X^*}{T^*} \right)$$

with a constant bore-front velocity $U = 2\sqrt{gh_u}$ ($V^* = 2$).

Eq. 7.3 and Eq. 7.6 shows that, when the reservoir is infinitely long, the maximum flow depth (h) at any location X approaches $(4/9)h_u$ asymptotically. Similarly, the prediction of Ritter's analytical solution at any location X (Eq. 7.9 and Eq. 7.6)

approaches $(4/9)h_u$ asymptotically, but as time goes to infinity. In addition, Ritter's solution of the mean flow velocity (Eq. 7.10 and Eq. 7.8) indicates that at any location X , the latter approaches the value of $(2/3)\sqrt{gh_u}$ as time goes to infinity.

7.3.1 Runup and force

Figure 7.19 shows the influence of the reservoir size on the time-history of the runup and the net base horizontal force acting streamwise on the column. Up to when the runup force for $L = 6.0\text{m}$ is reached (4.27 seconds), the time-history of the resulting force is very similar for all reservoir lengths, with the exception of the initial impact force. As explained previously in Section 7.2, it is believed that the initial impact force should be the same for all reservoir lengths, as the velocity and the depth of the bore-front at impact are the same (Figure 7.20). This would also agree with the observations of Lauber and Hager [1998] indicating that the flow characteristics of the bore-front are not affected when $L/h_u > 2$, which is the case for all lengths considered here. For the greater reservoir lengths of 11.2 and 15.0m, a quasi-steady force of 571N is reached, on average, between 5 and 9 seconds. In the case of $L = 15.0\text{m}$, this force is sustained much longer, although it begins to diminish near the end of the simulation. As expected, the time-history of the water runup on the column follows the same trend as the force time-history.

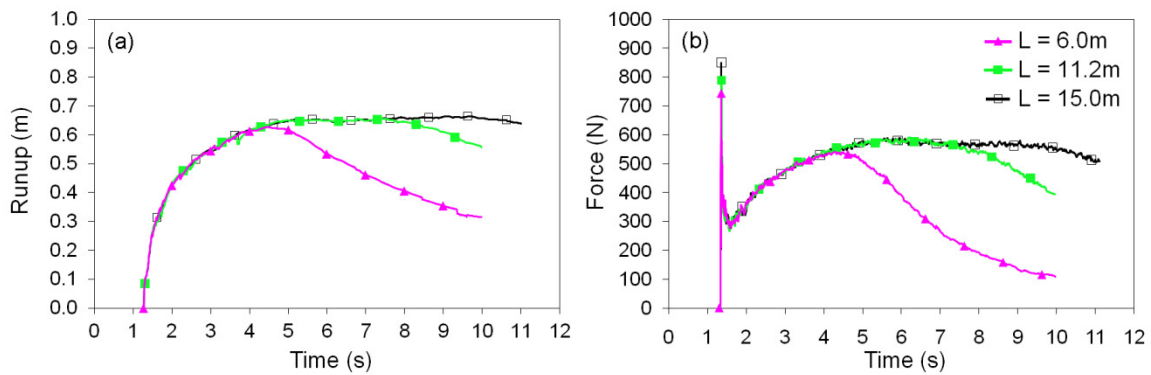


Figure 7.19. Time-history of (a) the bore runup and (b) the net base horizontal force acting streamwise on the column for reservoir lengths (L) of 6.0, 11.2 and 15.0m corresponding to simulations D3-L1, D3-L2, and D3-L3, respectively.

7.3.2 Water Surface Elevation and Velocity

Figure 7.20 shows the time-history of the water surface elevation and the depth-averaged flow velocity 0.25m upstream, as well as 0.77, 1.77 and 3.77m downstream of the gate. The depth-averaged flow velocity is calculated according to Eq. 7.1. For the locations downstream of the gate ($X > 0$), the analytical predictions of the flow depth and of the flow velocity provided by Ritter's [1892] analytical solution (Eq. 7.9 and Eq. 7.10) are also shown. In addition, the maximum flow depths calculated from the empirical equations of Lauber and Hager [1998] are plotted at their corresponding occurrence times (Eq. 7.3 and Eq. 7.4). For the latter, the numbers 1, 2, and 3 correspond to the reservoir lengths of 6.0, 11.2, and 15m, respectively. As mentioned previously, it can be noticed that, for all reservoir lengths, both the numerical flow depth and flow velocity are initially identical. This is until either the arrival of the aforementioned obstruction-induced wave, or the weakening of the hydraulic bore due to the emptying of the reservoir. In the second case, the associated gradual reduction in both the flow depth and velocity is clearly observed in Figure 7.20-a at 3.19 and 6.32 seconds in the case of the reservoir lengths of 6.0 and 11.2m, respectively. These times were determined as the instant when the deviation with the time-history associated to $L = 15.0$ m is greater than 0.005m. At the location of the gate ($X = 0$), the dimensionless time corresponding to this decline can be estimated as $T^* = 1.7\lambda^*$, according to Lauber and Hager [1998]. The latter corresponds to the time required for the negative dam-break wave to reach the upstream end of the straight reservoir. For the impoundment depths of 6.0m and 11.2m, this instant correspond 3.53 and 6.58 seconds, respectively. In the case of $L = 11.2$ m and at the shown locations with $X > 0.77$ m, the arrival of the obstruction-induced wave occurs before the expected decrease in bore intensity (e.g. Figure 7.20-e). For $L = 6.0$ m however, the latter is only observed for $X = 3.77$ m.

Inspecting the time-history of the water surface elevation, there is relatively good agreement with Ritter's analytical solution until the arrival of the obstruction-induced wave (Figure 7.20-e). This agreement is particularly true for the rate at which the water surface elevation suddenly increases after the bore-front has reached a particular location,

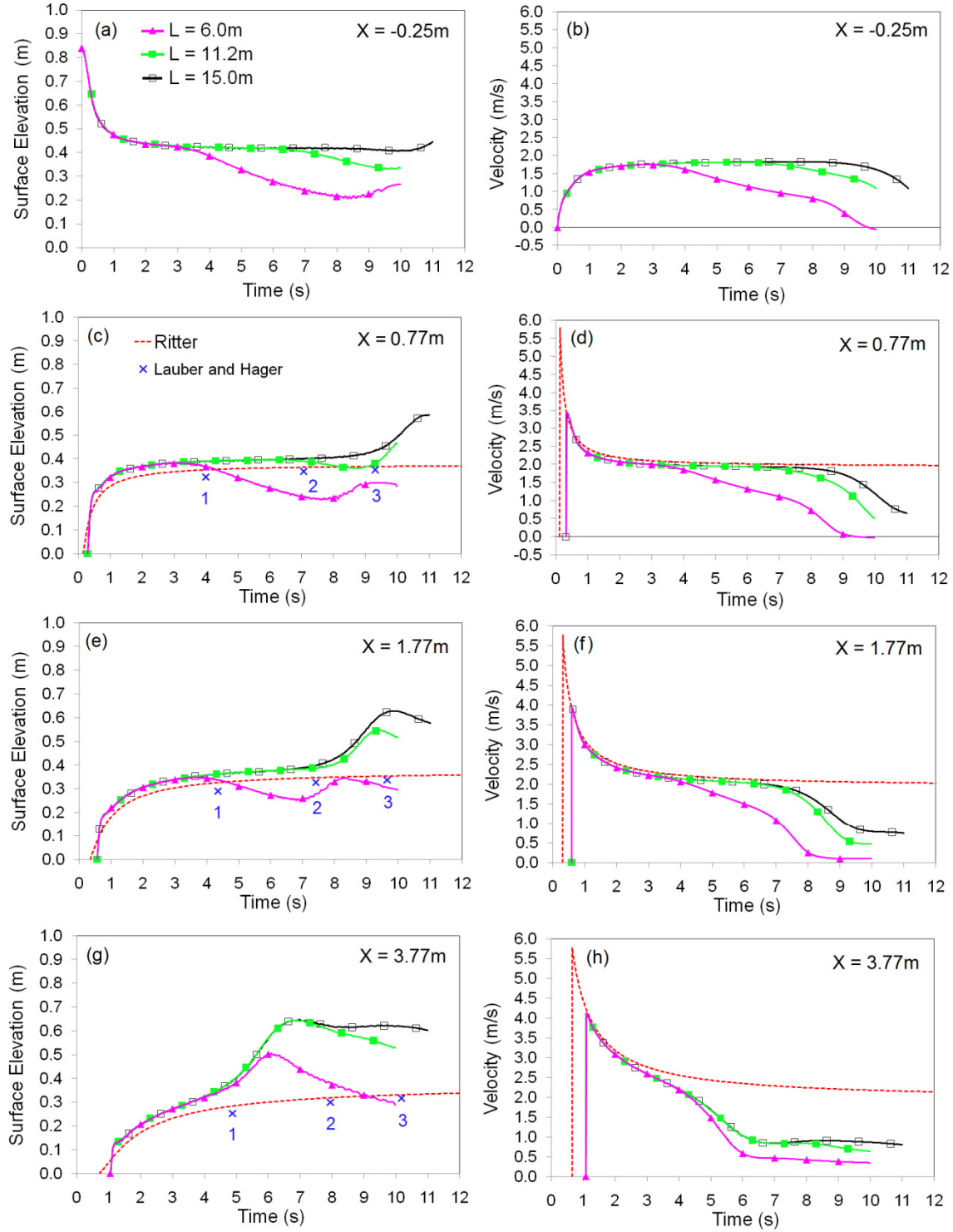


Figure 7.20. Time-history of the water-surface elevations (left) and depth-averaged flow velocities (right) for reservoir lengths (L) of 6.0, 11.2 and 15.0m at locations of (a, b) -0.25m , (c, d) 0.77m , (e, f) 1.77m , and (g, h) 3.77m from the gate.

although this rise is generally more sudden in the case of the numerical simulation. Also, where the dam-break flow is not altered by the obstruction-induced wave, there is a constant difference of approximately 0.038m between the numerical results and Ritter's solution of the flow depth. It can be seen that the maximum flow depth predictions by the empirical relationship of Lauber and Hager [1998] (Eq. 7.3) are slightly below the one provided by Ritter's analytical solution (Figure 7.20-c). Also, their prediction of the occurrence time of the maximum flow depth (Eq. 7.4) is approximately 1 second later than was observed in the numerical simulations. For both reservoir lengths of 6.0 and 11.2m, only Figure 7.20-c shows this last discrepancy, although for $L = 6.0\text{m}$, it is also observed in Figure 7.20-e.

The behaviour of the obstruction-induced wave is also affected by the length of the reservoir. Closer to the column, the arrival of the wave, which inflects a deviation from Ritter's solution of the flow depth, occurs at approximately the same time for all L (see Figure 7.21-g). In the case of $L = 6.0\text{m}$ however, the height reached by this wave is not as high as was the case for the greater reservoir lengths. In the case of the greater L , the same wave height is reached although it is only for $L = 15.0\text{m}$ that the flow depth beyond the passing of the wave remains approximately constant. As you move further upstream, the height reached by the wave increases as the reservoir length increases (Figure 7.22-e). This is explained by the fact that, further upstream, the wave arrives later and the more time the reservoir has to empty. Hence, the weaker dam-break flow upstream of the wave is not as strong as the flow around the column downstream. This results in a decrease in elevation of the wave and of the "plateau" directly downstream of the wave.

An initial discrepancy can be observed between the time-history of the flow velocity predicted by Ritter's solution and the one obtained in the numerical simulations (Figure 7.20-d). The latter can be explained knowing that, assuming an inviscid fluid and neglecting bottom friction, Ritter considered a constant bore-front velocity $U = 2\sqrt{gh_u} = 5.78\text{m/s}$. This is well known to be an over-prediction [Leal *et al.*, 2006], and it explains the quicker arrival time of the bore-front at locations $X > 0$. This discrepancy was expected as Ritter's solution is known to provide a poor prediction of dam-break flow characteristics during wave initiation just after the release of the reservoir and

particularly at the bore-front [Lauber and Hager, 1998]. As seen in the numerical simulations, the velocity of the bore-front (initial peak in the time-history of the flow velocity) actually accelerates with distance, and if a longer distance was considered, a deceleration would be observed due to the effect of bottom friction. Nevertheless, the rate at which the numerical flow velocity approaches the value of $(2/3)\sqrt{gh_u} = 1.93\text{m/s}$, beyond the passing of the bore-front, agrees very well with Ritter's solution (e.g. Figure 7.20-d). As expected, because of the decrease in the flow strength due to either the limited reservoir volume or the arrival of the obstruction-induced wave, there is a departure from the analytical prediction (Figure 7.20-f).

7.3.3 Steady-State Flow Condition

Based on the quasi-constant net base horizontal force acting on the column for $L = 15.0\text{m}$ (simulation D3-L3), one could say that, in the vicinity of the column, a steady-state condition occurs momentarily. According to Iizuka and Matsutomi [2000], the resulting force acting horizontally on a structure due to a steady tsunami flow can be estimated using the following relationship:

$$[7.11] \quad F_x = \frac{1}{2} \rho C_D u^2 h_f B_h$$

where C_D is a drag coefficient, u is the inland tsunami flow velocity, h_f is the inundation depth at the front of the structure, and B_h is the width of the structure. For the drag coefficient, they proposed 1.1 to 2.0. Interestingly, this relationship is identical to the classical relationship for form drag, namely:

$$[7.12] \quad F_D = C_D A_p \rho \frac{v_0^2}{2}$$

where A_p is the projected area of the structure in the direction of flow and v_0 is the free-stream velocity. According to Crowe *et al.* [2005], when the Reynolds number is greater

than 10^4 the approximate value of C_D for a square rod or column is 2.0. The latter is given by the following:

$$[7.13] \quad Re = \frac{v_0 L_c}{\nu}$$

where L_c is the characteristic length (being set equal to the side dimension of the square column), and $\nu = 10^{-6} \text{ m}^2/\text{s}$ is the kinematic viscosity of water at 20°C. Here, v_0 is taken as v_a at a distance 1.00m upstream of the column ($X = 3.77\text{m}$).

At a time of 8 seconds after the opening of the gate in simulation D3-L3:

$$Re = \frac{(0.89\text{m/s})(0.3\text{m})}{10^{-6}\text{m}^2/\text{s}} = 267,000 > 10,000$$

According to either Eq. 7.11 or Eq. 7.12 and assuming $C_D = 2.0$, the resulting force acting streamwise on the column is calculated as:

$$F_x = \frac{1}{2} \left(1000 \frac{\text{kg}}{\text{m}^3} \right) (2.0) (0.89\text{m/s})^2 (0.62\text{m}) (0.30\text{m}) = 147.33\text{N}$$

On the other hand, the corresponding force predicted by the numerical model is 558.57N, which gives a difference of 411.24N between the two values.

However, in the form drag relationship (Eq. 7.12), a steady flow is assumed around the obstacle. Hence, in addition to dynamic components, the derivation considers equal hydrostatic pressure distributions on both the front and back sides of the obstacle. As is clear from Figure 7.23, this condition is not present in our case. The inundation depth at the front of the column, or runup, is 0.62m, whereas it is 0.17m at the back of the column. Using the relationship below, the numerical hydrostatic forces acting at the front and back of the column are calculated.

$$[7.14] \quad F_H = \frac{1}{2} \gamma_w b h^2$$

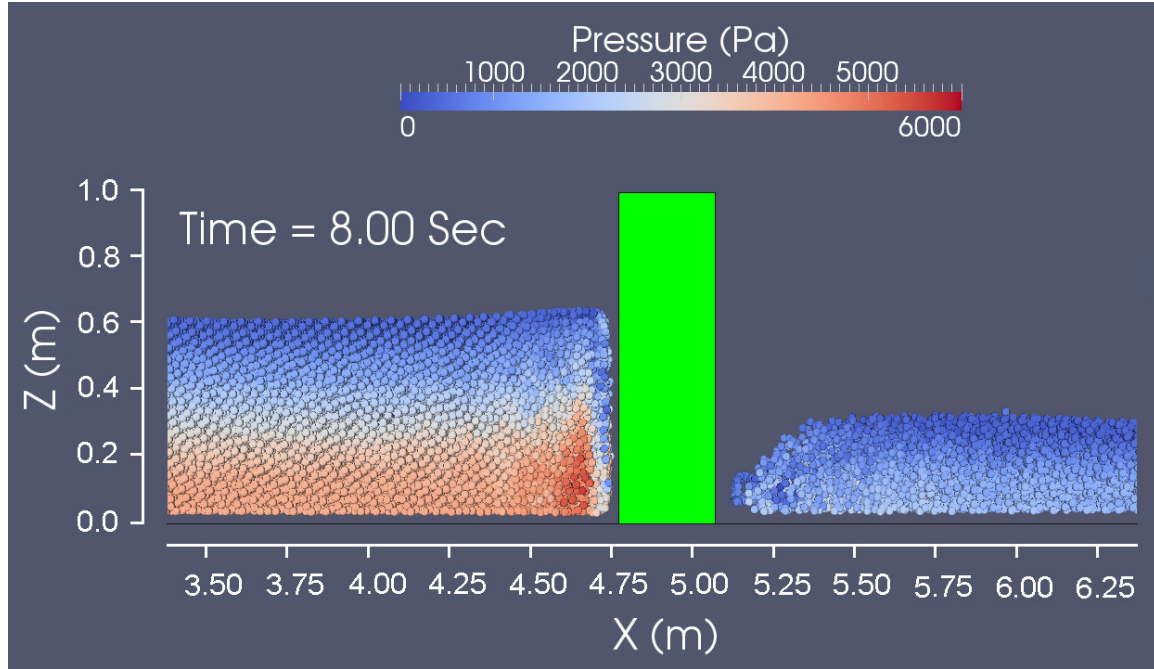


Figure 7.23. Difference in the water surface elevation at the front and back of the column at steady-state for $L = 15.0\text{m}$ (simulation D3-L3).

$$F_{H,front} = \frac{1}{2} \left(9810 \frac{\text{N}}{\text{m}^3} \right) (0.30)\text{m} (0.62\text{m})^2 = 565.65\text{N}$$

$$F_{H,back} = \frac{1}{2} \left(9810 \frac{\text{N}}{\text{m}^3} \right) (0.30)\text{m} (0.17\text{m})^2 = 42.53\text{N}$$

The difference in the hydrostatic force acting at the front and back of the column is then 523.12N. This difference in the hydrostatic forces acting on the front and back faces of the column is of the same order of magnitude as the difference between the net horizontal force estimated by the classical form drag relation and that predicted by the numerical model.

7.3.4 Discussion

The larger the reservoir size (or length in the case of a straight reservoir) is, the more sustained the hydraulic bores generated therein (and hence the hydrodynamic forces exerted on a structure) will be. As predicted by Ritter [1892], the numerical results show

that, for a sufficiently long reservoir, the flow depth and velocity at any location downstream of the gate approach the values $(4/9)h_u$ and $(2/3)\sqrt{gh_u}$, respectively, as time advances, given that the reservoir length is sufficiently long. However, the latter does not hold when there is disturbance to the dam-break flow, such as the increase in water depth that results from the flow obstruction caused by the column.

For the longest reservoir length considered ($L = 15.0\text{m}$), the resulting force on the column eventually reached a quasi-steady value. However, this numerical value is much greater than that calculated based on classical fluid mechanics. This is because the hydrostatic pressure distributions on the front and back of the column are not the same in the current experiments. Our experiments show that, unless the tsunami inundation depths at the front and back of a structure are approximately the same, such a relationship is not appropriate.

7.4. Influence of Bed Condition

The experimental results of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] presented in Section 6.1 demonstrated that the bed condition, whether it is wet or dry, has a significant influence on the resulting hydrodynamic force at initial impact of hydraulic bores with columns. According to these results, large initial impulses in the pressure and force are inhibited (e.g. Figure 6.22) when the hydraulic bore propagates over a wet bed ($h_d \approx 0.005\text{m}$). As noted in the literature review (Section 3.3), Árnason [2005], who only considered a wet-bed condition ($h_d = 0.02\text{m}$), also did not observe such peaks in the force at the instant of impact of hydraulic bores with columns (Figure 3.12). However, Ramsden [1993] observed that the initial impact pressure acting on a vertical wall is higher (Figure 3.10) in the case of a bore propagating over a wet bed ($h_d = 0.003\text{m}$) compared to a bore propagating over a dry bed ($h_d = 0$). To investigate further the influence of the bed condition on the development of the hydrodynamic forces and elucidate the above disagreement, the impact on structures of bores propagating on both dry and wet beds is investigated numerically in this section. Two structure types are considered, a square column with a side dimension of 0.30m (simulations D3-DRY and D3-WET) and a vertical wall that expands across the width of the test channel

(simulations D4-DRY and D4-WET). The computational domains corresponding to these structures are shown in Figure 4.14 and Figure 4.15, respectively, with $L = 5.0\text{m}$ and $W = 1.3\text{m}$.

As discussed in Section 4.3.2, some difficulties related to the repulsive boundary condition were encountered when trying to simulate the thin downstream water depth (h_d) observed in the physical experiments. Since h_d has a strong influence on both the bore propagation and the resulting hydrodynamic forces generated at initial impact, a quantitative comparison of experimental results with numerical results corresponding to an equivalent wet-bed condition was not feasible. Therefore, the numerical wet-bed condition considered here corresponds to an h_d greater than was observed in the physical experiments of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] (0.080m vs. 0.005m). A qualitative comparison between only numerical results is performed for this reason. In fact, as a result of the high strength of the repulsive forces exerted by boundary particles on the fluid particles, the numerical h_d employed herein was the smallest one possible.

7.4.1 Bore Propagation

In order to verify their propagation characteristics, the bore profiles produced by the model are compared to the analytical solution of the shallow water equations (SWE) using the method of characteristics [Stoker, 1957]. As described in the literature review (Section 3.1), the SWE are a simplified form of the incompressible Navier-Stokes equations, for which viscosity and turbulence are neglected and for which hydrostatic pressure and uniform fluid velocity over depth are assumed. The comparisons of the entire bore profile for the dry- and wet-bed conditions are shown in Figure 7.24 and Figure 7.25, respectively, at various times after the release of water in the reservoir. The agreement with the analytical solution, shown in Figure 7.24 and Figure 7.25, is relatively good and comparable to the level of agreement observed by Stansby *et al.* [1998], who compared small-scale experimental bore profiles with the same analytical solution of the shallow water equations (see Figure 3.5 and Figure 3.6). Nevertheless, with regards to Figure 7.25, it seems that the bore-front velocity U in the case of the wet-bed condition is less than the constant analytical one of 2.88m/s. Figure 7.26 shows the time-history of U , for both dry- and wet-bed conditions, up to the time of impact of the

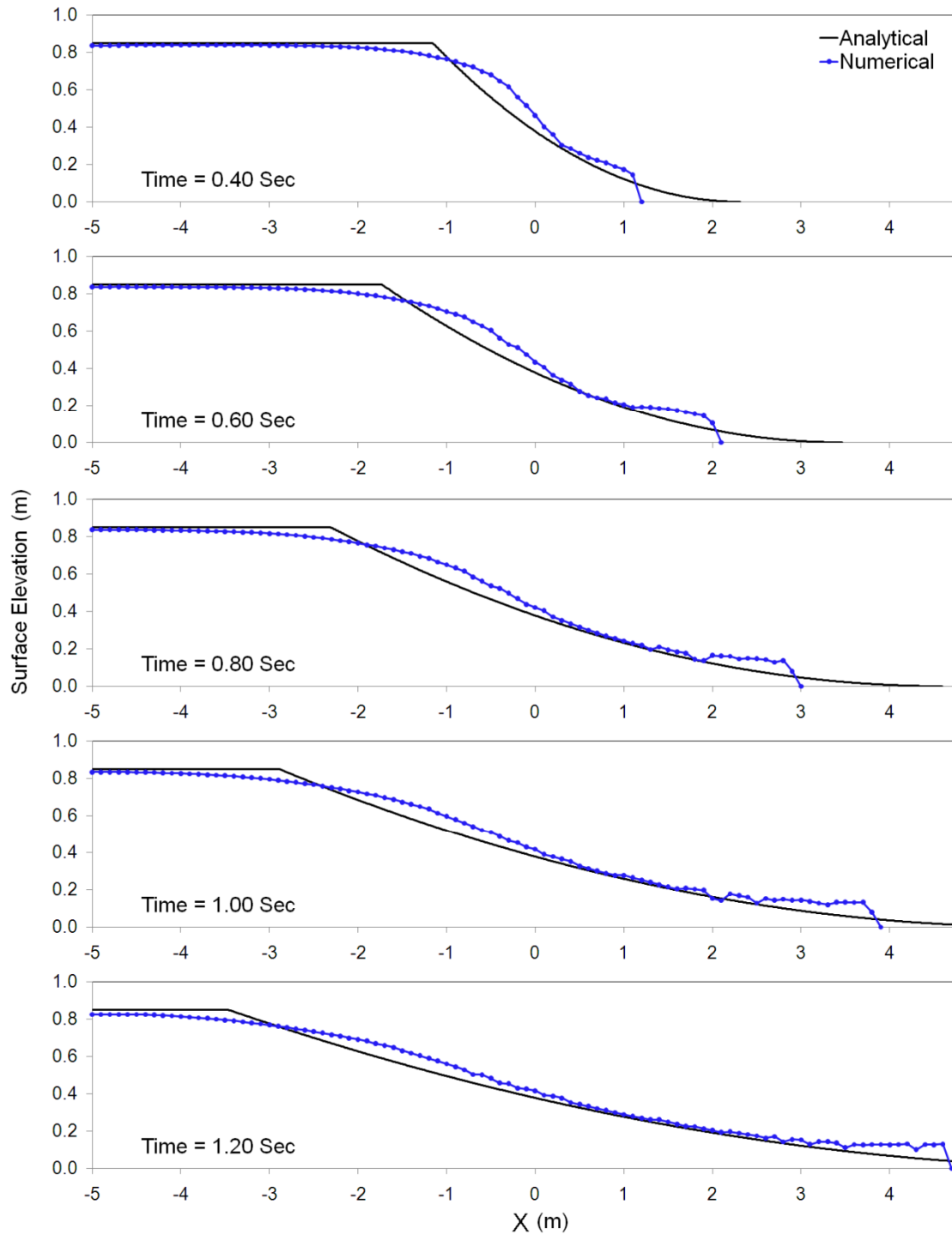


Figure 7.24. Comparison of numerical and analytical (SWE) bore profiles for the dry-bed condition (simulation D3-DRY).

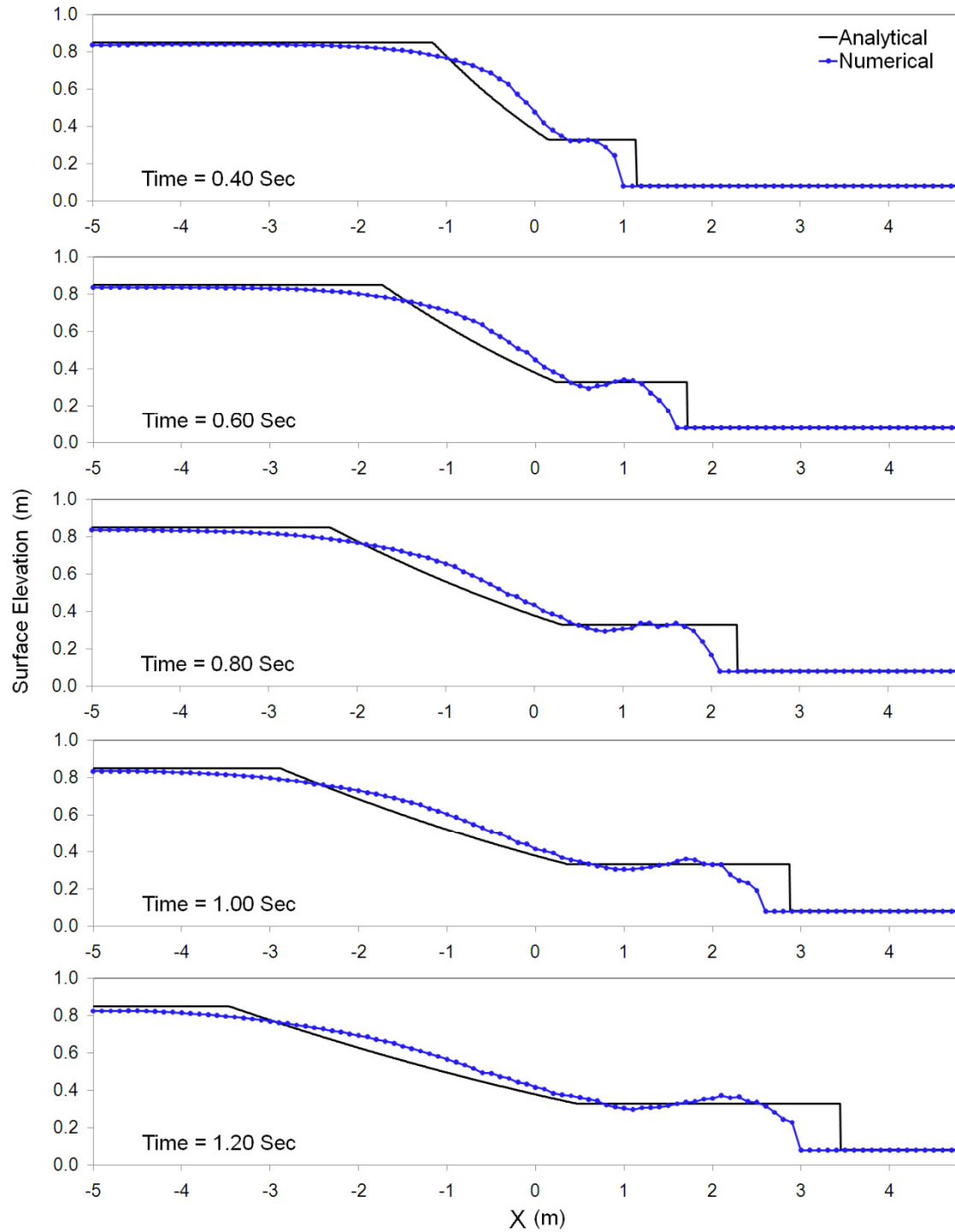


Figure 7.25. Comparison of numerical and analytical (SWE) bore profiles for the wet-bed condition (simulation D3-WET).

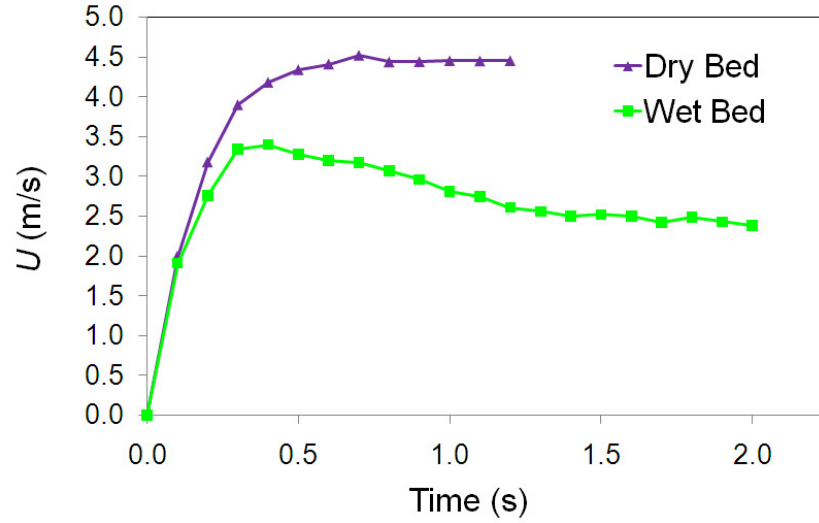


Figure 7.26. Time-history of the bore-front velocity U for the dry- and wet-bed conditions until impact with structures (simulations D3-DRY and D3-WET).

bores with the structures, which occurs at 1.23 and 2.04 seconds, respectively. It can be observed that, in the case of the dry bed, the bore-front keeps accelerating until a quasi-steady velocity of approximately 4.45m/s is reached. However, in the case of the wet bed, the bore-front initially accelerates to a maximum velocity of 3.40m/s but then it continues to decelerate until impact with the structures. At that instant, the bore-front has a velocity of 2.38m/s. This deceleration and the difference in the location of the bore-front can be attributed to the fact that the analytical solution neglects the effects of turbulence. Indeed, Figure 7.27-b demonstrates that strong turbulence seems to be created in the region of the flow where some of the kinetic energy, or momentum, of the advancing bore-front is transferred to the initially still fluid particles located on the channel's bottom. Hence, dissipation of energy due to turbulence could explain this deceleration of the bore-front when propagating on a wet bed.

Figure 7.27 shows side and plan views of the numerical simulations for which the bore-fronts are located at a distance of 3.60m from the downstream end of the reservoir. [Note that, for this figure and all others showing images from the numerical results, the black lines represent solid boundaries]. The previously discussed (Section 4.3.2) effect of the repulsive forces of the boundary particles on the still fluid particles can be clearly seen over the flume's bottom in Figure 7.27-b and next to the solid boundaries (i.e. lateral

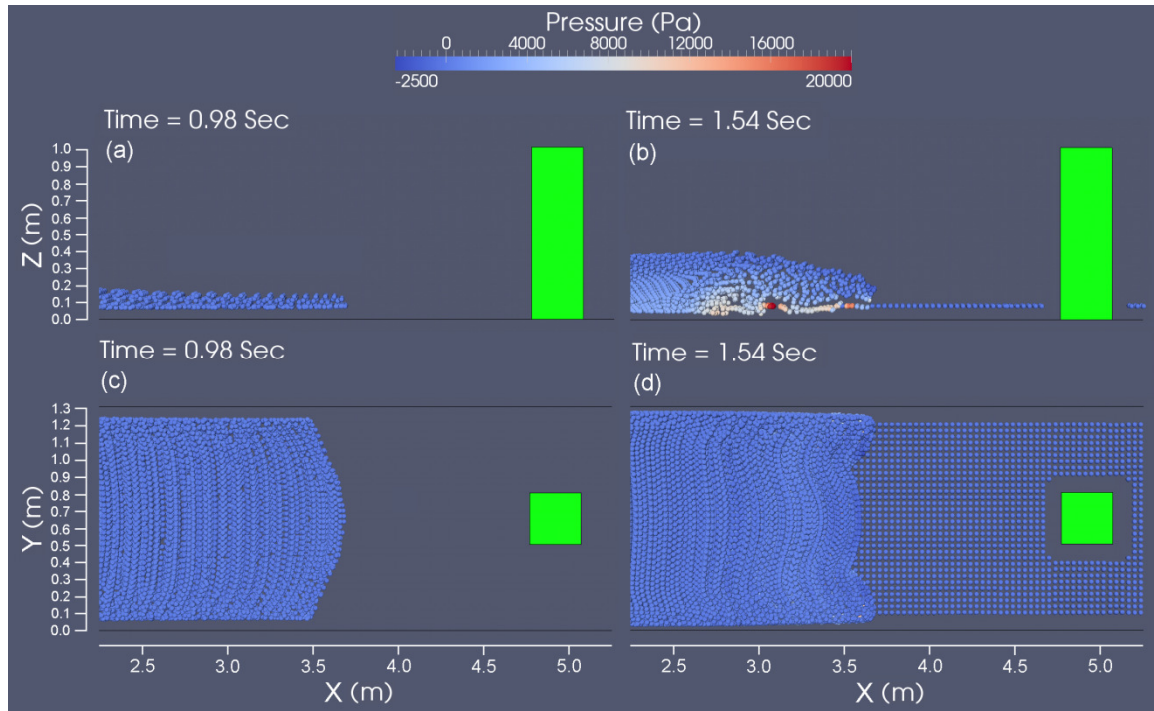


Figure 7.27. Side (top) and plan (bottom) views of bore-front for the dry- (left, simulation D3-DRY) and wet-bed (right, simulation D3-WET) conditions at a distance of 3.60m from the downstream end of the reservoir.

walls and column) in Figure 7.27-d. Upon inspection of the plan views of the bores in Figure 7.27-c and in Figure 7.27-d, it is clear that the shape of the bore-front across the width of the channels is remarkably different. In the case of the dry bed, the bore-front is furthest downstream at the center of the flume, which could possibly be explained by the additional friction caused by the lateral walls. However, in the case of the wet bed, the bore-front is the furthest downstream next to the flume walls. In the latter case, further analysis showed that the bore-front profile becomes higher at the center of the channel. This causes accelerated flow motion towards the walls, which eventually reduces the bore-front profile at the center of the channel.

Figure 7.28 shows the profile of the bore-fronts at a section 3.60m from the downstream end of the reservoir. For both dry- and wet-bed conditions, the very front of the bores is almost vertical. This steep section reaches an elevation of 0.13m in the case of the dry-bed condition (compared to an elevation of 0.21m in the case of the wet-bed one). Furthermore, the overall slope of the bore profile along the ~0.60m reach just

upstream of the steep front section is much milder in the case of the dry-bed condition (0.00277) compared to the wet-bed one (0.30544).

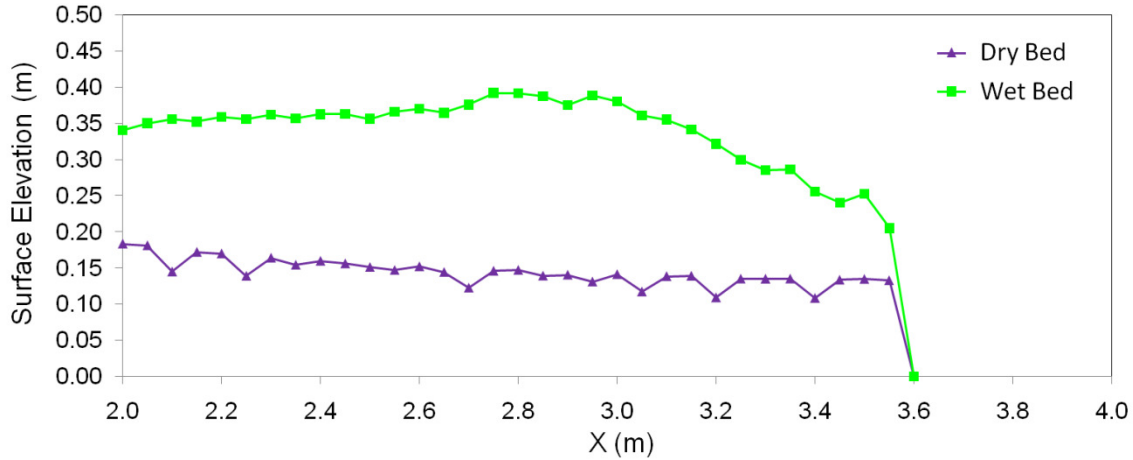


Figure 7.28. Comparison of bore-front profile for the dry- (simulation D3-DRY) and wet-bed (simulation D3-WET) conditions at a distance of 3.60m from the downstream end of the reservoir.

7.4.2 Bore Runup and Velocity Field

Figure 7.29 shows the time-history of the bore runup on the impacted structures. For both dry- and wet-bed conditions and for both structures, the rate at which the runup suddenly increases moments after initial impact is essentially the same. Nevertheless, for the entire duration of the simulations, the runup associated with the wet-bed condition reaches elevations that remain higher than the one associated with the dry-bed condition. In the case of the column (Figure 7.29-a) and for the wet-bed condition, the rates at which the runup varies beyond the initial sudden rise are very similar to those observed for the dry-bed condition. Also, it seems that the time-histories of the runup would converge beyond a 5 seconds period. The runup on the column for the dry-bed condition continues to increase beyond the first sudden rise until a maximum value of 0.59m is reached at 3.87 seconds, after which a decrease is observed. Conversely, in the case of the wet-bed condition, this additional increase beyond the first sudden rise, until a maximum value of 0.68m is reached at 3.57 seconds, is not as significant. On the other hand, in the case of the vertical wall structure (Figure 7.29-b), for both dry- and wet-bed conditions, the value

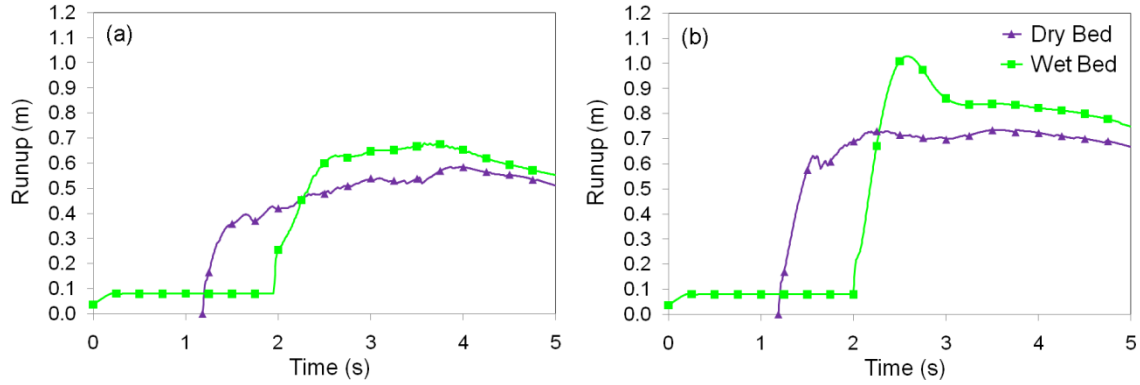


Figure 7.29. Time-history of the bore runup for dry- and wet-conditions on the (a) column (simulations D3-DRY and D3-WET) and (b) wall (simulations D4-DRY and D4-WET).

of the runup beyond the initial sudden rise remains essentially constant, except that in the case of the wet bed condition a significant hump occurs at 2.59 seconds. Furthermore, the residual difference in the runup between the two bed conditions is noticeable in the case of the vertical wall structure (Figure 7.29-b).

Considering the evolution of the bores as they interact with the two structures, remarkable differences are observed. Firstly, for the column structure, the velocity vectors in the center plane of the channel shown in Figure 7.30 demonstrate the formation of a very well defined surface roller on the upstream face of the column as it is being impacted by the bore propagating on the dry bed. As in shown in Figure 7.8 in the case of simulation D3-L1, over time, this surface roller was observed to increase in elevation and to expand laterally. Subsequently, at the moment when this surface roller reaches the flume walls, it will reach its maximum elevation, which corresponds to the maximum runup at 3.87 seconds on Figure 7.29-a. Following that instant, a wave forms and, due to the obstruction caused by the presence of the column, is reflected back upstream, eventually causing the runup to decrease. The latter can be observed in Figure 7.14 for simulation D3-L1. However, in the case of propagation over the wet bed (Figure 7.31), the flow is also initially forced upwards due to the presence of the column and, instead of being redirected upstream in a rolling motion (as observed in the case of the dry-bed condition), the bulk of the flow is redirected downwards and around the column. Similar

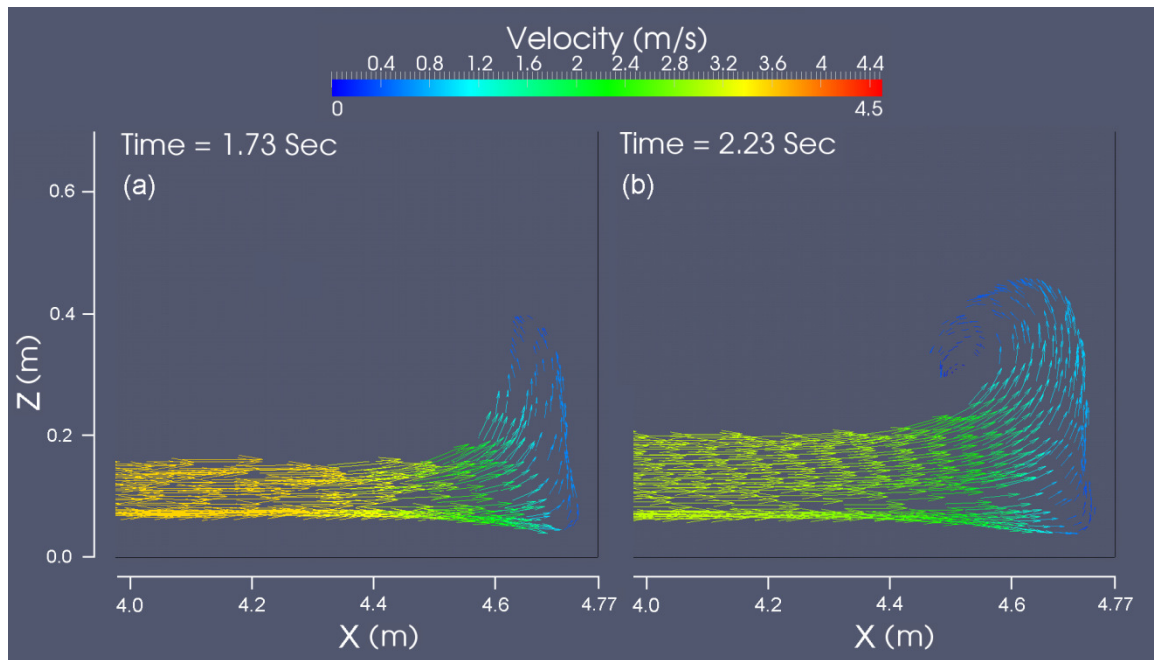


Figure 7.30. Side view of velocity field upstream of column's center for the dry-bed condition (a) 0.50 and (b) 1.00 seconds after initial impact (simulation D3-DRY).

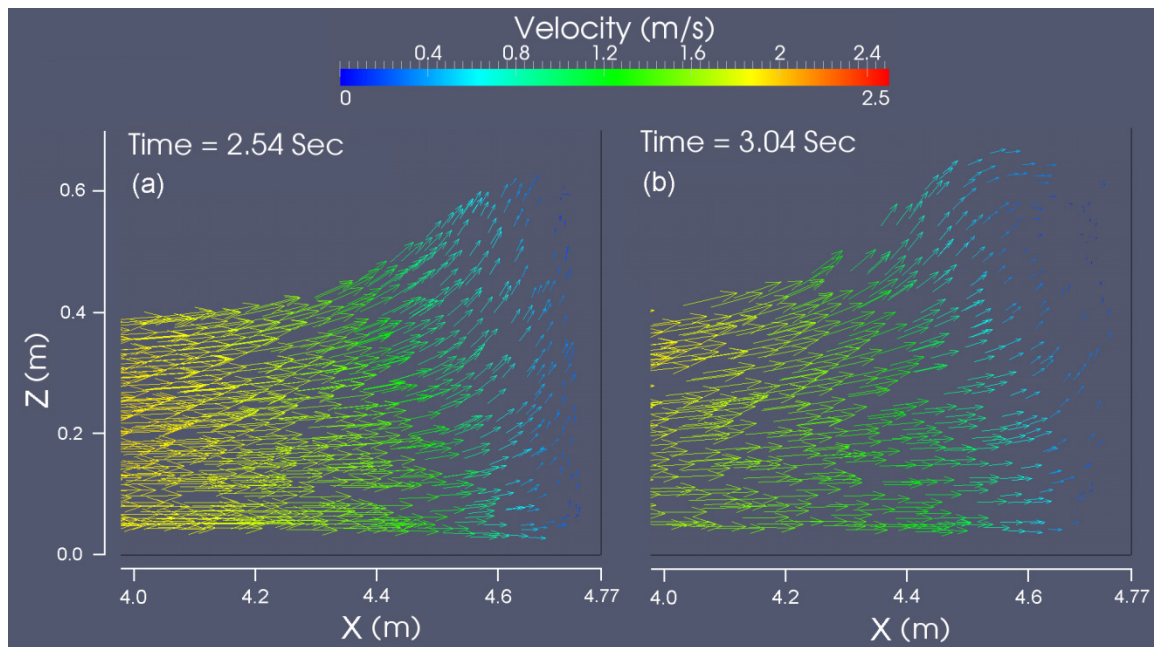


Figure 7.31. Side view of velocity field upstream of column's center for the wet-bed condition at (a) 0.50 and (b) 1.00 seconds after initial impact (simulation D3-WET).

to the case of the dry bed, the flow obstruction engendered by the column causes the formation of a wave that propagates back upstream, followed by a reduction in the runup. Moreover, as shown in Figure 7.32, the complete formation of the wake just downstream of the column occurs much sooner in the case of propagation over the wet bed. This figure displays the velocity vectors in a plane close to the channel's bottom at times corresponding to 0.75 seconds after initial impact. This supports the previous observation that the wake in the physical experiments, generally performed with a wet-bed condition, forms faster than in the corresponding numerical simulations performed with a dry-bed condition (see Figure 6.3-c and Figure 6.3-d in Section 6.1.1). The latter also explains the time lag in the time-history of the water surface elevation at the back of the column at wave gauge W13 (e.g. Figure 6.6). However, as seen in Figure 6.8 for the larger impoundment depth of 1.15m, the reflection effect of the lateral walls discussed in section 7.2.3 seems more predominant and such delay is not occurring.

Secondly, considering the vertical wall structure, both bores advancing on the dry and wet beds are fully reflected back in the upstream direction. However, this process exhibits important differences in flow patterns. Figure 7.33 shows a time sequence of the interaction of the bore propagating on the dry bed with the vertical wall. Moments after initial impact, a fluid jet is reflected relatively close to the wall and plunges backwards on top of the very shallow incoming flow in a motion similar to the breaking of an ocean wave (Figure 7.33-d and Figure 7.33-e). From this point onwards, two very distinct currents with opposite directions can be observed, as the reflected water literally flows upstream over the incoming water (e.g. Figure 7.33-g). Significant shear stresses occur between the two flow layers as velocity reduces to zero at their interface. Furthermore, a stagnant region located directly in front of the wall develops slowly and remains relatively narrow, as the incoming flow is redirected upwards just upstream. For the bore propagating on a wet bed, a reflected fluid jet also plunges backwards on the incoming flow (Figure 7.34-f). However, unlike the dry-bed condition, this occurs at a location further away from the wall and much later after initial impact (Figure 7.33-d). Also, the stagnant region in front of the wall forms more rapidly for the wet-bed condition and its length is also much larger. [Note that the times considered in Figure 7.33 and Figure 7.34 are the same with respect to initial impact of the bores with the vertical wall].

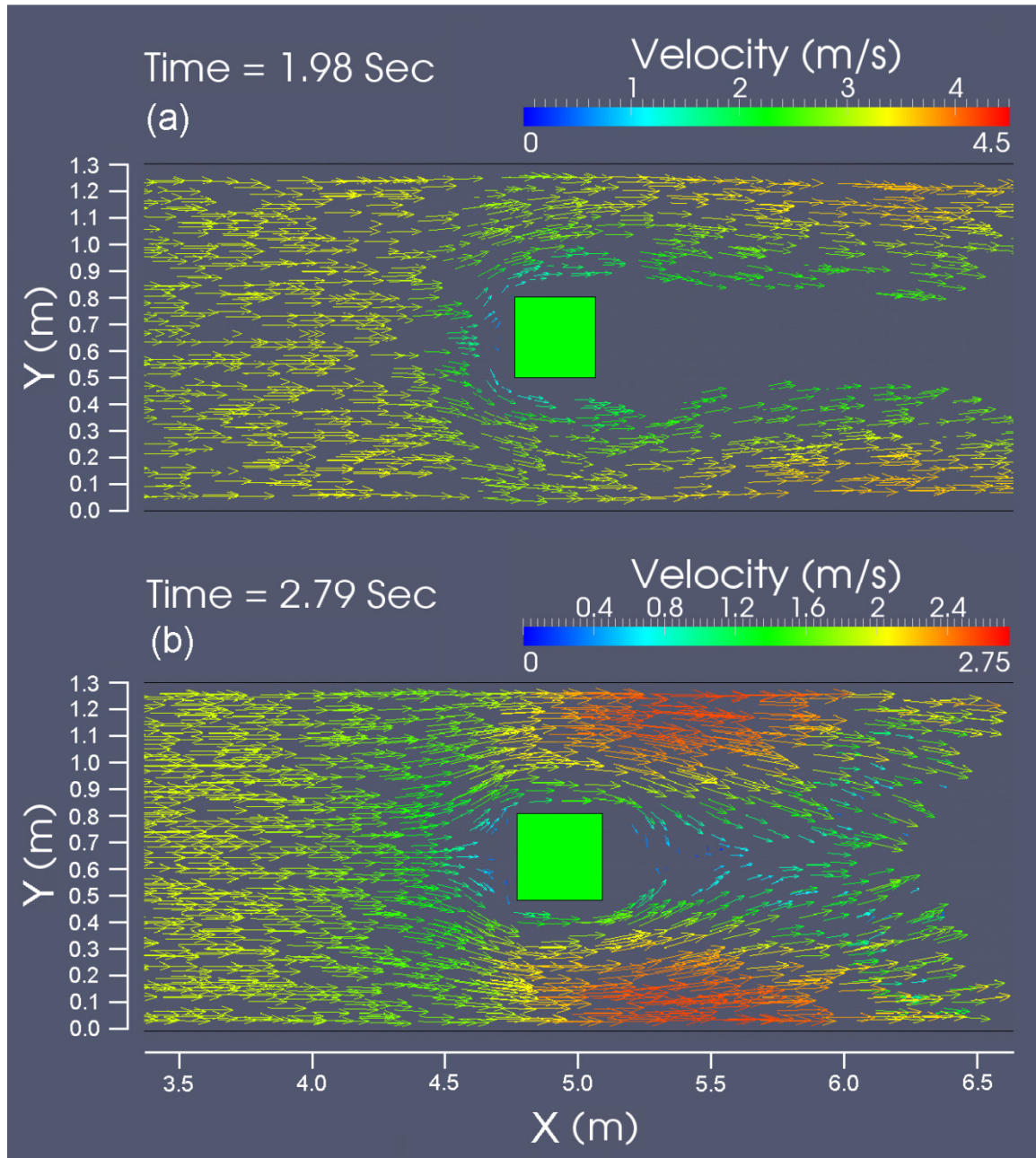


Figure 7.32. Plan view of velocity field around the column at times corresponding to 0.75 seconds after initial impact for (a) dry- and (b) wet-bed conditions corresponding to simulations D3-DRY and D3-WET, respectively.

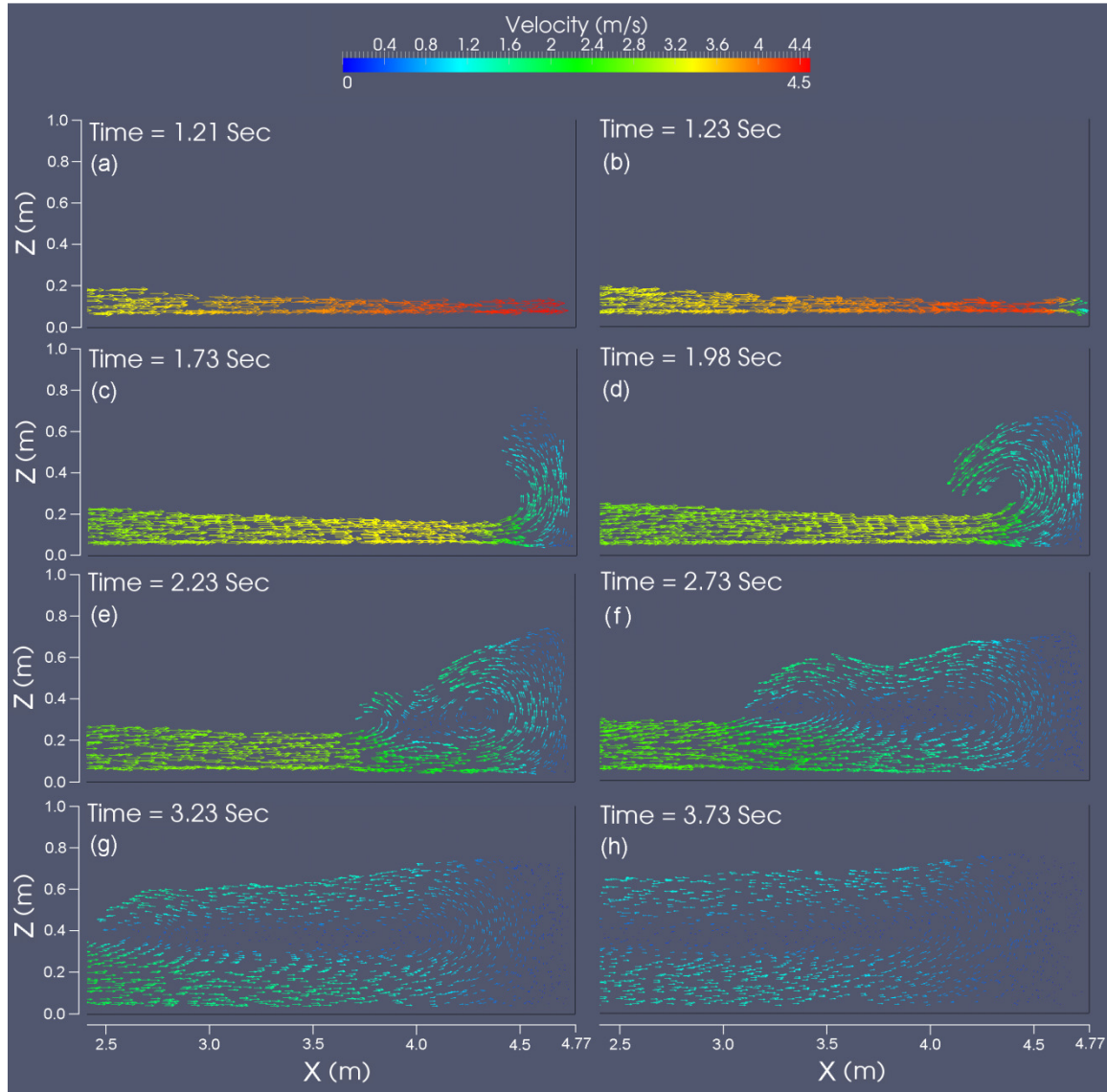


Figure 7.33. Side view of velocity field upstream of wall's center for the dry-bed condition (a) moments before impact, (b) at initial impact and at times corresponding to (c) 0.50, (d) 0.75, (e) 1.00, (f) 1.50 (g) 2.00, and (h) 2.50 seconds after initial impact (simulation D4-DRY).

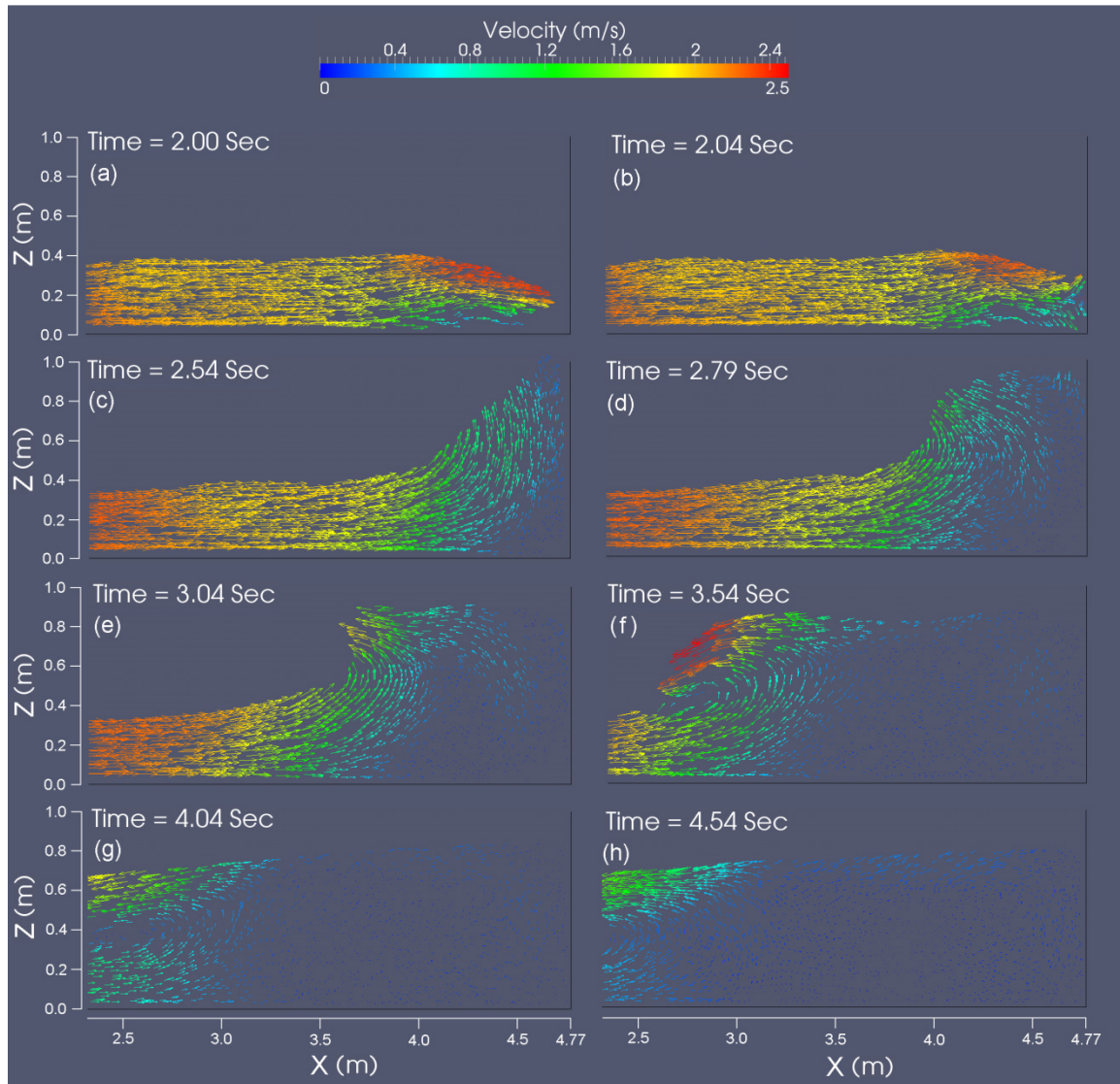


Figure 7.34. Side view of velocity field upstream of wall's center for the wet-bed condition (a) moments before impact, (b) at initial impact and at times corresponding to (c) 0.50, (d) 0.75, (e) 1.00, (f) 1.50 (g) 2.00, and (h) 2.50 seconds after initial impact (simulation D4-WET).

7.4.3 Pressure

Figure 7.35 provides the time-history of the pressure at various elevations. The pressure is obtained at the center of the upstream vertical face of the column structure, as a result of impact of bores propagating on both dry and wet beds. As observed in the physical tests (Figure 3.11-a), in the case of the dry-bed condition, the time-history of the pressure at a low elevation (Figure 7.35-a) has a large spike at initial impact. However, such a spike does not occur in the case of a wet bed. For the dry-bed case, at higher elevations on the column, the magnitude of this spike at initial impact decreases with elevation (e.g. Figure 7.35-c) and practically vanishes at elevations above 0.15m (e.g. Figure 7.35-e). At these elevations, the increase in pressure following initial impact is much more gradual. This is due to the steepness of the bore-front which, as shown in Figure 7.28, is very mild above an elevation of 0.15m. A similar behaviour was also observed by Kirkgöz [1991]. In his experiments of wave breaking on vertical and inclined walls, the highest impact or shock pressures were measured when the wave front was approximately parallel to the wall, or in other words, when it was very steep with respect to the inclination of the wall. For the wet-bed case however, the bore-front is also very steep for elevations of up to approximately 0.20m and there is still no significant spike at initial impact. This is due to not only the lower velocity of the bore propagating on the wet bed, but, more importantly, to increased void in the fluid particles in the lower region of the bore-front (Figure 7.27-b). Although air entrainment is not accounted for in the numerical model, one might surmise that the high turbulence resulting from the transfer of momentum to the still water layer would entrain more air in this lower region of the bore-front. Since air is much more compressible than water [Peregrine, 2003], its presence would undoubtedly provide some attenuation and hence affect the impact pressure. This effect of the entrapped air bubbles in the bore-front was also suspected based on experimental observations discussed in Section 6.1.3. Since the bore-front for the wet-bed condition remains relatively steep at elevations higher than 0.20m, unlike the observed gradual rise for the dry-bed condition, a sudden rise in pressure is generated at higher locations on the column (e.g. Figure 7.35-e). Surprisingly, for these higher elevations on the column, the opposite was observed in the experimental results shown in Figure 6.12 and Figure 6.13. Unfortunately, as wave gauges W2 and W3 upstream of the column were not utilised in

the corresponding experimental runs, it was not possible to examine the steepness of the bore-front in order to further investigate this disagreement. Beyond initial impact, quasi-steady hydrodynamic pressures are observed for both bed conditions. However, for the wet-bed case, a second rise is observed between around 2 and 3 seconds at all elevations. This second rise seems to be extended over a longer time period at higher elevations.

For the case of the vertical wall structure, Figure 7.36 provides, for both dry and wet beds, the time-history of the pressure exerted at various elevations at the center of the wall resulting from bore impact. For the dry-bed condition, very similar observations can be made for the case of the column structure (Figure 7.35). However, in the case of a wet bed, the pressure development is different. Firstly, moderate initial peaks can be observed even at lower elevations on the wall (Figure 7.36-a and Figure 7.36-b). Figure 7.37 shows the plan view of the pressure field at initial impact in a region close to the channel's bottom. In the case of the bore propagating on a wet bed impacting the vertical wall (Figure 7.37-d), the fluid particles from this aforementioned lower void region converge towards the center of the wall. In the case of the column (Figure 7.37-c), fluid particles are ultimately diverted towards and around the sides of the structure. It seems that this convergence of fluid particles, which occurs in the case of the vertical wall, generates the initial peaks, or spikes, in the time-history of the pressure although the bore propagates on a wet bed. It can also be seen that, for all the elevations shown in Figure 7.36, the pressure beyond initial impact for the wet-bed condition is considerably higher when compared to the dry-bed condition. This seems to remain valid until the end of the simulations. This could partly be attributed to the runup reaching higher elevations in the case of the wet-bed condition (Figure 7.29-b). However, as shown in Figure 7.34, it could be mostly due to the large hydrodynamic pressures generated as flow velocities within the bulk of the flow suddenly go to zero moments after initial impact. To support this, the side views of the pressure field in a plane located at the center of the wall structure shown in Figure 7.38 show that, for a dry-bed condition (Figure 7.38-a), the pressures next to the wall are closer to the hydrostatic condition, when compared to the wet-bed case (Figure 7.38-b).

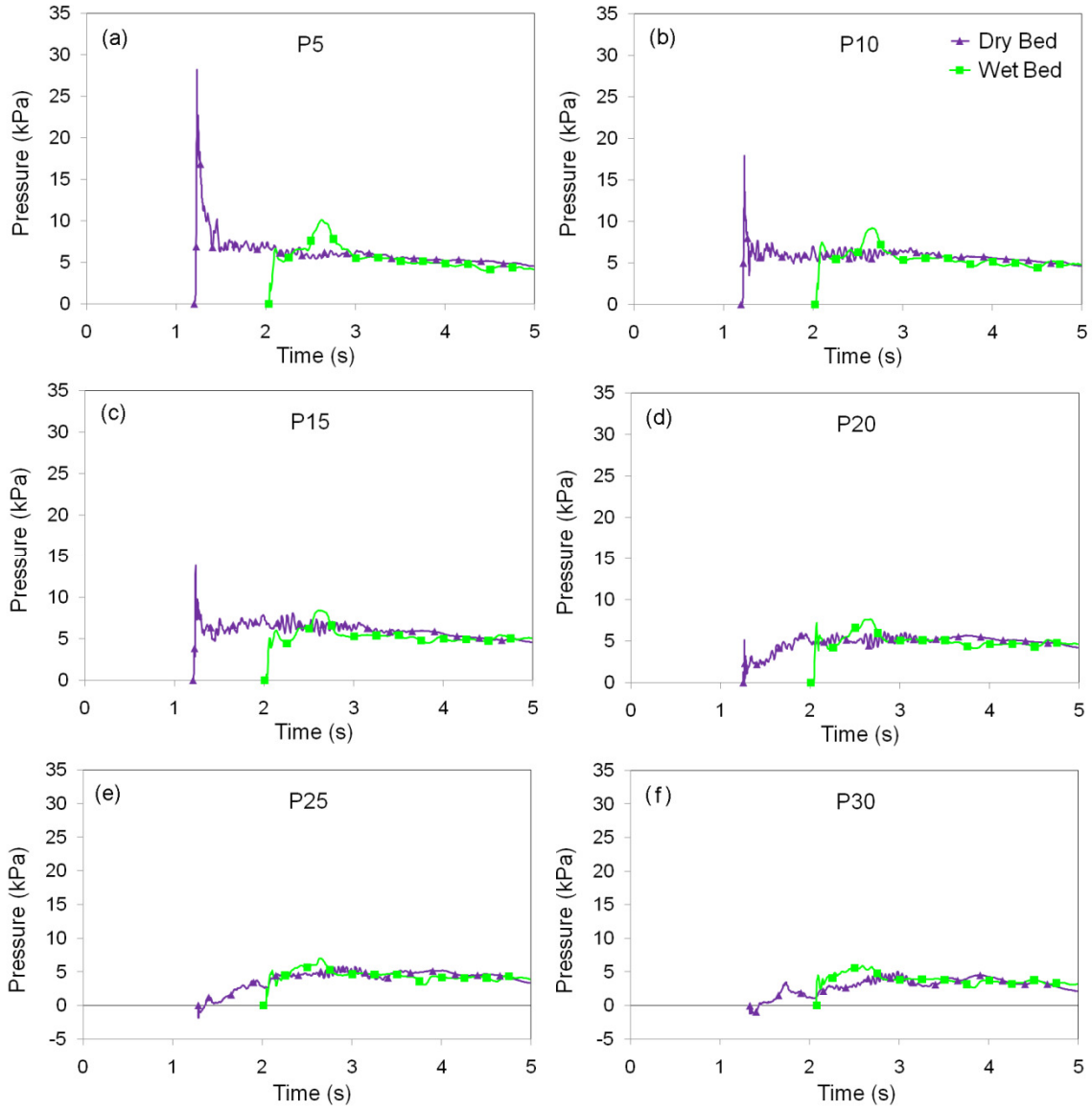


Figure 7.35. Time-history of pressures acting at the center of the upstream face of the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, and (f) 0.30m above the channel's bottom for the dry- (simulation D3-DRY) and wet-bed (simulation D3-WET) conditions.

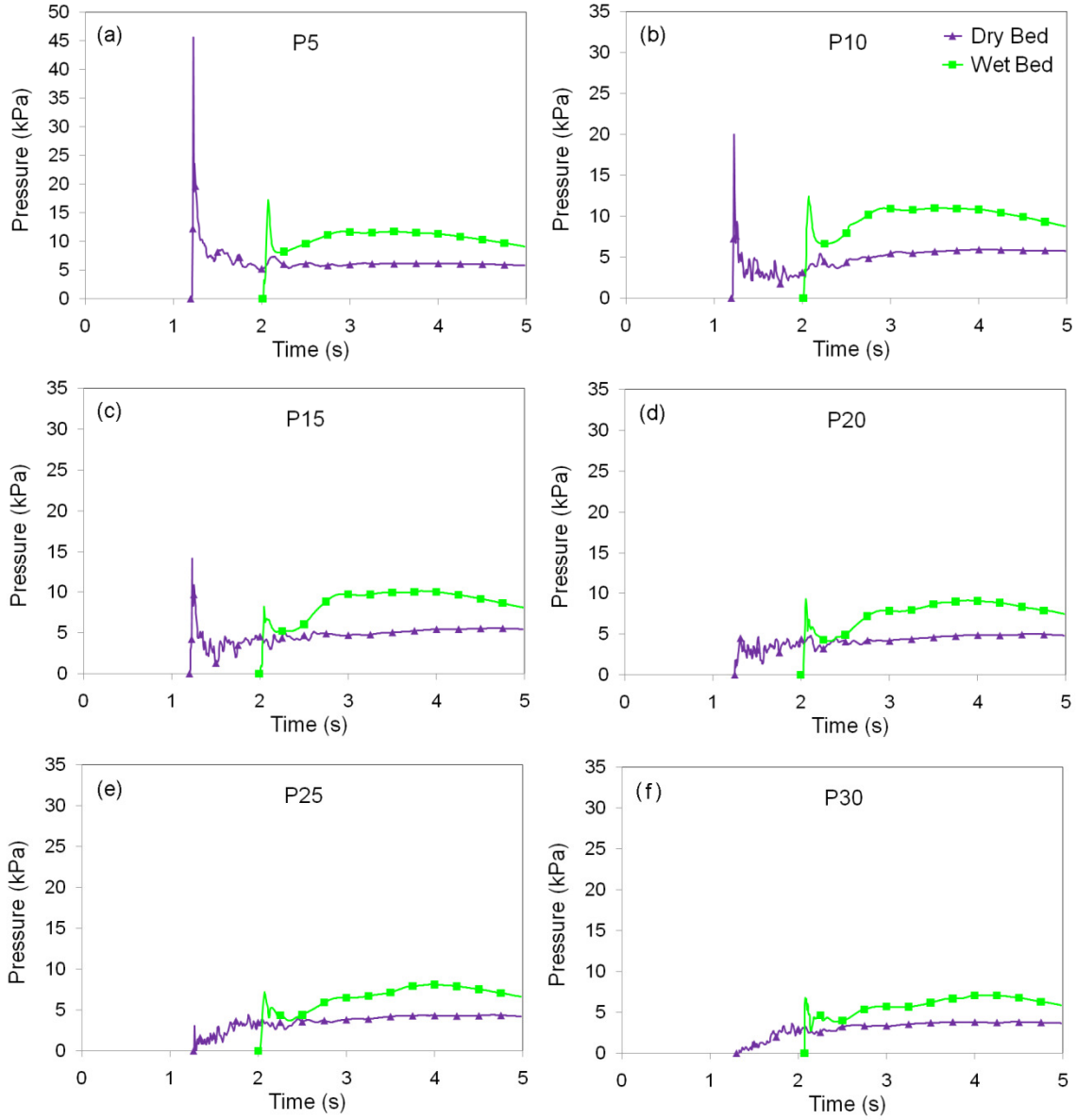


Figure 7.36. Time-history of pressures acting at the center of the wall at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, and (f) 0.30m above the channel's bottom for the dry- (simulation D4-DRY) and wet-bed (simulation D4-WET) conditions.

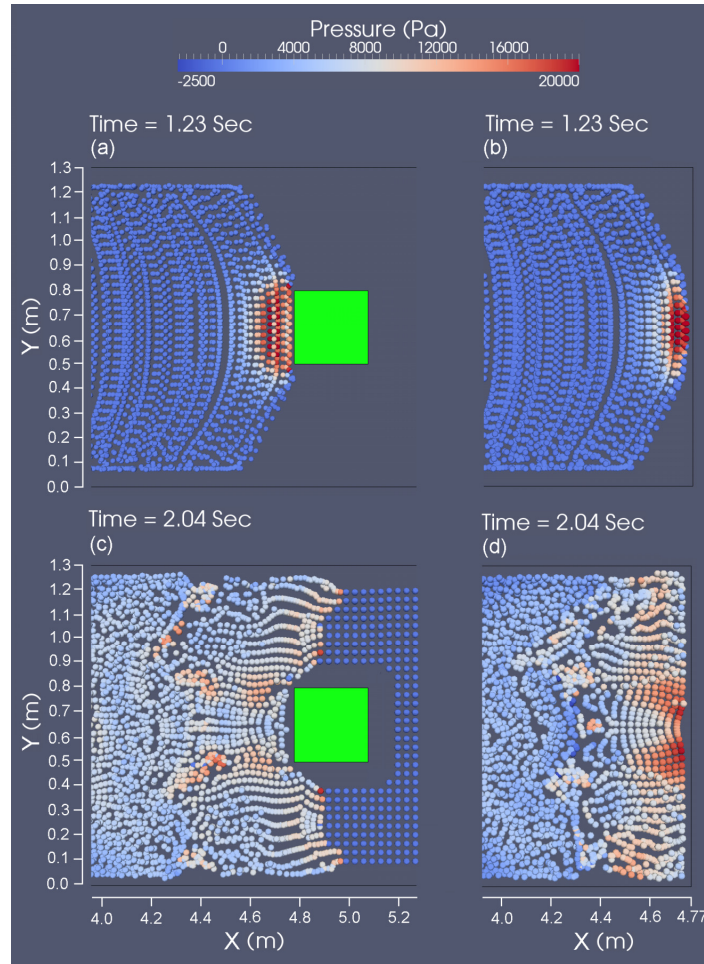


Figure 7.37. Plan views of the pressure field at initial impact with the column (left, simulations D3-DRY and D3-WET) and wall (right, simulations D4-DRY and D4-WET) for the dry- (top) and wet-bed (bottom) conditions.

Referring back to the results of Ramsden [1993] (see Figure 3.10), one can deduce why he observed higher impact pressures on his physical wall model impacted by a bore that propagated over a wet bed. The lowest sampling location he considered corresponded to an elevation for which the bore-front profile for the wet-bed condition was steeper than that for the dry-bed condition. As shown in Figure 7.35-d and Figure 7.36-d, respectively, this is similar to both the column and wall structures cases.

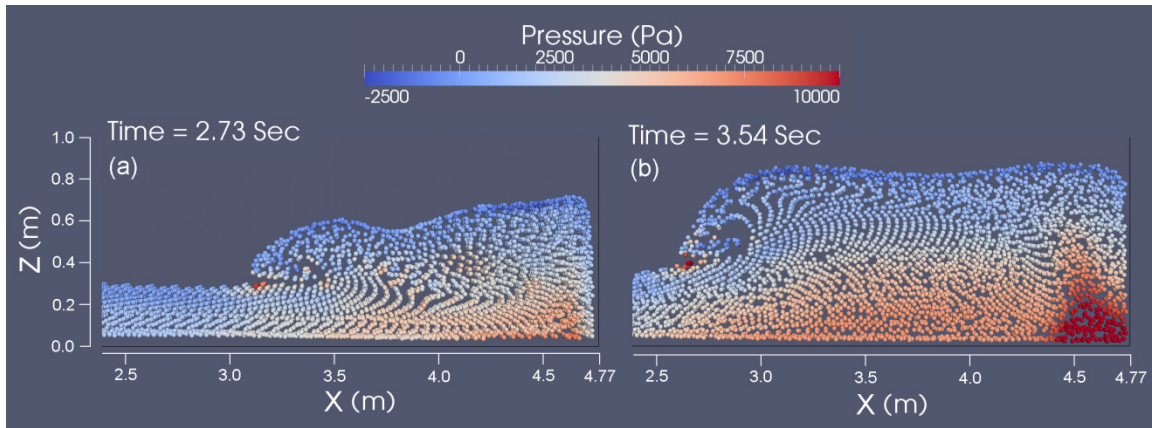


Figure 7.38. Side views of the pressure field at times corresponding to 1.50 seconds after initial impact with the wall for the (a) dry- and (b) wet-bed conditions (simulations D4-DRY and D4-WET).

7.4.4 Force

For both dry and wet bed conditions, Figure 7.39 shows the time-history of the total net base horizontal force acting on the column, along with the total net force acting on the wall per meter width. For both structures, the spike in the force at initial impact is more defined and generally of higher magnitude in the case of the dry-bed condition. This can be anticipated based on the large initial peaks observed in the time-histories of the pressure acting at low elevations on the structures (e.g. Figure 7.35-a and Figure 7.36-a). Nevertheless, for the wet-bed case, the time-histories of the force acting on both types of structure still show a sudden rise at initial impact. This is due to the overall contribution of the small, but sudden spikes in the pressure at higher elevations on the structures. This is in contrast to the dry-bed condition, for which sudden spikes occurred at lower elevations only.

Considering the column structure (Figure 7.39-a), the second peak in the force, or the runup force, for the dry-bed condition is lower in magnitude when compared to the initial impact force. This force occurs at the same instant at which runup is maximum (Figure 7.29-a). On the other hand, for the wet-bed condition, the runup force is larger and occurs sooner, as the water runup on the column rises suddenly to a higher elevation just after initial impact. However, due to the propagation of the bore around the column, this is not sustained over time.

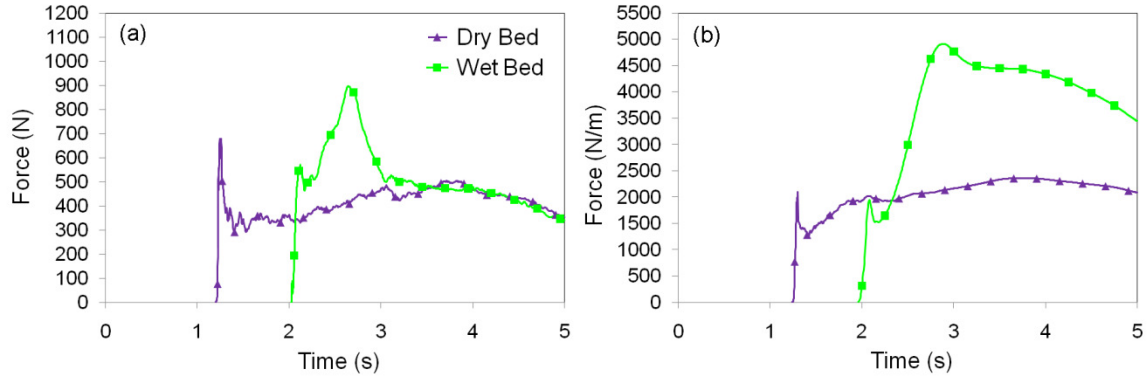


Figure 7.39. Time-history of (a) the net base horizontal force acting on the column (simulations D3-DRY and D3-WET), and of (b) the net force per meter width of wall (simulations D4-DRY and D4-WET) for dry- and wet-bed conditions.

Towards the end of the simulations, the time-histories of the force for both dry- and wet-bed conditions converge to a more quasi-steady hydrodynamic state. A steady hydrodynamic state is not eventually reached, partly due to the short duration of the simulations and implicitly to the limited size of the generated bores. Árnason [2005], who studied the interaction of bores propagating on wet beds with a square column (Figure 3.12), did not observe such a distinctive runup force. Although his experiments were conducted with a wet bed (which explains the lack of the initial impact force), and with a h_d/h_u ratio up to 0.067, Árnason also observed the formation of a surface roller on the upstream face of the column, as was the case in this study for the dry-bed condition. This assumes that, based on the h_d/h_u ratio, the runup process and hence the runup force will either be similar to that recorded for the dry-bed condition if the ratio is low, or similar to the one observed for the wet bed condition considered in this study, for which $h_d/h_u = 0.094$.

For the case of the vertical wall structure (Figure 7.39-b), it was also observed that the runup force, which occurred at 3.80 seconds for a dry-bed condition, is close to the hydrostatic force based on the corresponding runup. However, for the wet-bed condition, the runup force is significantly larger and its maximum value is reached sooner (2.86 seconds). For the wet-bed condition, the development rate after initial impact of the runup force on the wall is similar to the one observed for the column. However, in the case of the column, due to the bore passing around the latter, the development of the

runup force is interrupted at 2.62 seconds (Figure 7.39-a). Lastly, as observed for the time-histories of pressures shown in Figure 7.36, for the wet-bed condition the residual force on the vertical wall is much greater than the hydrostatic one. However, this force seems to return to a hydrostatic value later on in the numerical simulations.

7.4.5 Discussion

The numerical model was used in this section to study the interaction of tsunami-like bores propagating on both dry and wet beds with two types of structures: (a) a slender column with a square cross-section, and (b) a vertical wall with a width corresponding to the width of the test flume.

Based on the literature review (Sections 3.2 and 3.3), which demonstrated that although the propagation of bores over dry and wet beds is relatively well understood, the development of hydrodynamic forces resulting from their impact with structures has not been compared, nor investigated thoroughly. Numerical simulations have been performed to provide detailed insights into the complex flow patterns occurring in the vicinity of the structures analyzed and in the resulting pressure and forces acting on them. The main observations of this qualitative comparison are:

1. At low elevations on the column's front face, large spikes in the time-history of the pressure occur at initial impact of a bore propagating over a dry bed. However, these are not observed in the case of a bore propagating over a wet bed (Figure 7.35-a). This is in agreement with experimental results (Figure 3.11). Numerically, this was determined to occur due to the low particle density in the lower region of the bore-front propagating over the wet bed (Figure 7.27-b). Experimentally, this could be due to a higher concentration of entrained air in this lower part of the bore-front as momentum transfer occurs at this location. However in the case of the vertical wall, fluid particles are concentrated towards the center of the structure (Figure 7.37-d) and such spikes can occur for the wet bed condition. However, they are of lesser magnitudes than the spikes corresponding to the dry bed condition (Figure 7.36-a).

2. For both structures (wall and column), the sudden rises in the time-history of the pressure at initial impact will depend upon the slope of the bore-front at the corresponding elevation. A steep slope will induce a sudden rise or possibly a spike, whereas a mild slope will induce a more gradual increase (e.g. Figure 7.36-d). This observation was first noted by Ramsden [1993].
3. For the time-history of the net force acting on both structures (wall and column), the magnitude of the initial impact force will generally be higher for the case of a bore propagating on a dry bed (Figure 7.39), due mainly to (1) above. However, as the front of a bore propagating on a wet bed is deeper and steeper (Figure 7.28), as mentioned in (2) above sudden rises in the time-history of the pressure at initial impact can occur at elevations further up the structure. These contributions can result in large impact forces.
4. In the case of bores propagating on wet beds, the maximum runup on both structure types is not only more pronounced, but also more sudden (Figure 7.29). Consequently, the second peak in the time-history of net force (runup force) is also higher and more sudden (Figure 7.39). In the case of the column (Figure 7.39-a), the time-history of the net force rapidly converges beyond the runup force to the quasi-steady hydrodynamic state and becomes equal to the time-history of the total net force observed in the case of the dry-bed condition. This abrupt change is due to the bulk of the bore passing around the column, which is in contrast to the complete blockage caused by the vertical wall (Figure 7.39-b).
5. In the case of the column impacted by the bore propagating on a dry bed, a very well defined surface roller forms on the upstream face of the column (Figure 7.30), as the runup on the structure gradually increases beyond initial impact (Figure 7.29). On the other hand, when impacted by the bore propagating on a wet bed, this surface roller does not form (Figure 7.31). Although Árnason [2005] performed similar physical experiments with a wet-bed condition and noted the formation of similar surface rollers, he did not observe the peaks in the runup

force (Figure 3.12) that were observed in this study (Figure 7.39-a). This implies that a surface roller (and hence a less significant runup force), can also form for a wet-bed condition, but for lower h_d/h_u ratios.

Chapter 8. Conclusions and Recommendations

8.1. Conclusions

Based on the quantitative comparison of numerical with physical results presented in Chapter 6 and the extensive analysis of numerical results in Chapter 7, the following conclusions can be drawn:

- Physical experiments, which investigate the impact mechanisms of rapidly-advancing tsunami-like hydraulic bores with structures, were simulated successfully using a single-phase three-dimensional WCSPH numerical model. This study demonstrates the suitability of this relatively new numerical method for reproducing such violent and nonlinear water impacts.
- To prevent the large fluctuations in the pressure field associated with WCSPH (see Section 3.4), a Riemann solver was employed. Although this approach proved to be quite successful, in regards to estimating impact forces on structures, it introduced excessive “numerical viscosity” in the computations. This numerical viscosity made the simulations of complex flow features more difficult. However, as demonstrated in Section 6.2, this detrimental side effect can be mitigated by increasing the resolution of the simulation via a decrease in the initial inter-particle spacing.
- The numerical experiments described in Section 7.1 clearly demonstrated the influence that reservoir geometry had on hydrodynamic forces generated by the bores on the model structures. Upon gate opening, the presence of a sharp 180° bend in the reservoir of our laboratory flume was found to cause brief flow acceleration, accompanied by premature reduction in the flow intensity. [These

flow features are not observed in the classical dam-break scenario case (in which a “straight” reservoir is assumed)]. The reduction in the flow intensity resulted in a higher but less sustained runup force.

- As a bore that is generated by the typical dam-break scenario is generally two-dimensional, the force at initial impact with a slender structure is theoretically not affected by the width of the test channel. The numerical results presented in Section 7.2 indicated that, when the test channel is ‘narrow’ with respect to the width of the column structure, significant flow obstruction results. This causes the flow depth directly upstream of the column to increase significantly compared to that downstream. Consequently, the runup of water on the upstream face of the column reaches higher elevations and the associated runup force is therefore greater. Furthermore, because of ‘wall’ effects, the wake downstream of the column develops more rapidly in a narrower test channel.
- To investigate the influence of reservoir size on both wave dynamics and resulting hydrodynamic forces, the length of a straight reservoir was varied numerically (Section 7.3). In conformity with analytical and empirical relationships, the model simulations showed that, as time advances, the depth and velocity of the flow at any location downstream of the gate tends to $(4/9)h_u$ and $(2/3)\sqrt{gh_u}$, respectively for a ‘long’ reservoir. If the test channel is wide, such that the column structure does not substantially increase the water depth upstream by obstructing the incoming flow, the resulting quasi-steady force acting on the column could be estimated using the standard form drag formula. However, even though the force can reach steady-state, it cannot be estimated using the classical approach if the depths, immediately before and after the column, are not essentially the same.
- The results of laboratory studies described in the literature review (Section 3.2) and the numerical simulations performed in this research (Section 7.4) clearly show that the bed condition (i.e. wet vs. dry) has a significant influence on the

propagation characteristics of a hydraulic bore. Generally, when a bore propagates over a wet bed, its front is both higher and steeper and it also has a lower velocity compared to when it propagates over a dry bed. These differences significantly affect the pressure distributions and resulting hydrodynamic forces acting on impacted structures. Furthermore, when the bed is slightly wet (i.e. a downstream depth of just a few millimeters), the changes in the bore-front profile might not be significant. However, it is suspected that, due to turbulence associated with the momentum transfer from the advancing bore-front to the downstream still water layer, considerable air entrainment occurs. As is well known from many past studies related to breaking wave impacts, entrapped air bubbles can significantly inhibit impulsive shock pressures at initial contact of the water with the structure.

Our research findings demonstrate that the hydrodynamic forces on structures caused by the impact of tsunami bores will depend on certain highly variable factors. As a tsunami is often composed of a series of waves, the bore-front associated with the first wave will generally advance onshore over dry land (i.e. a ‘dry-bed’ condition). However, the subsequent waves, which can sometimes be larger than the first one, will propagate on land previously flooded by the first wave (i.e. a ‘wet-bed’ condition). Clearly, the bed condition at a particular site will strongly depend on topographical land features, such as local depressions. It is also clear that any constrictions influencing the flow width, either natural or manmade, will influence the inundation depths and velocities around impacted structures, and consequently, the resulting hydrodynamic forces acting on them. The effects of other field-related factors such as terrain roughness and fluid density variations, due to mixing with earth materials, remain essentially unknown. Furthermore, it is interesting to note that the salinity of the water will also influence the concentration of entrapped air bubbles. This is because bubbles tend to persist longer in saltwater due to their relatively smaller size [Peregrine, 2003]. Based on the above observations, it is clear that any guidelines for the design of buildings to withstand tsunami loading should account for site-specific flow characteristics.

8.2. Recommendations for Future Work

The research findings described in this thesis helped to identify the following areas for recommended future numerical and experimental work:

- Although it guaranteed reasonable accuracy in the estimation of hydrodynamic forces generated by tsunami bores on structures, the Riemann solver scheme used in this research introduced excessive numerical diffusion. Given the earlier success of Crespo *et al.* [2011], in their GPU-based implementation (see Section 3.4), further research should be undertaken with a view to reducing the inaccuracies in the pressure field associated with WCSPH. Employing a density filter, as an alternative to a Riemann solver, might prove to be successful in this regard.
- The researches of Nouri *et al.* [2010] and Al-Faesly *et al.* [2012] also dealt with the impact of waterborne debris on structures. As the impact mechanics of floating objects are already included in the open source code of SPHysics, future work could concentrate on the numerical reproduction of such impacts. However, as demonstrated in Section 5.3.1, this would require improving the repulsive boundary condition used to simulate solid boundaries.
- The effectiveness of mitigation walls was investigated experimentally by Al-Faesly *et al.* [2012]. Based on the successful results of this thesis, it is believed that these relatively similar experiments could also be reproduced numerically with good agreement using the SPH method.
- Comparisons of SPH with more established grid-based numerical methods are still scarce in the literature and existing ones mainly consider simple benchmark scenarios.
- Regardless of the numerical method employed, the results of this research show that a two-phase model must be used if the impulsive pressures at initial impact

are to be simulated accurately. In the case of the SPH method, open source and even commercial implementations that model both air and liquid phases are rare.

- Additional experimental work is required to explore further the influence of the bed condition on air entrainment in the bore-front. Laboratory tests, requiring the measurement of entrapped air concentrations in the fluid, might possibly employ the special instrumentation used by Bullock *et al.* [2006] in their study of the impact of breaking waves on walls.

References

- [1] Al-Faesly, T., Palermo, D., Nistor, I. and Cornett, A. (2012). *Experimental modeling of extreme hydrodynamic forces on structural models*, International Journal of Protective Structures (IJPS), 3(4), 477-505.
- [2] Arikawa, T. [2011]. Experimental study on tsunami disaster, *7th International Workshop on Coastal Disaster Prevention*, January 26-27, Tokyo, Japan, 2011, CD-ROM.
- [3] Árnason, H. [2005]. “Interactions between an incident bore and a free-standing coastal structure”, PhD Thesis, University of Washington, Seattle, USA.
- [4] Atwater, B. F., Satoko, M.-R., Kenji, S., Yoshinobu, T., Kazue, U. and Yamaguchi, D. K. [2005]. *The orphan tsunami of 1700 – Japanese clues to a parent earthquake in North America*, United States Geological Survey (USGS), Professional Paper No. 1707, 144 pp.
- [5] Bagnold, R. A. [1939]. *Interim report on wave pressure research*, Journal of the Institution of Civil Engineers (ICE), London, 12, 202-226.
- [6] Batchelor, G. K. [1967]. *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge, United Kingdom.
- [7] Benz, W. [1990]. *Smoothed particles hydrodynamics: a review*, in de Jager, C., “Numerical Modeling of Non-linear Stellar Pulsation: Problems and Prospects”, Kluwer Academic, Boston.
- [8] Bonet, J. and Lok, T.-S. L. [1999]. *Variational and momentum preservation aspects of Smoothed Particle Hydrodynamic formulations*, Computer methods in Applied Mechanics and Engineering, 180, 97-115.
- [9] Briggs, M. J., Synolakis, C. E., Harkins, G. S. and Green, D. R. [1995]. *Laboratory experiments of tsunami runup on a circular island*, Pure Applied Geophysics, 144, 569-593.
- [10] Bullock, G. N., Obhrai, C., Peregrine, D. H. and Bredmose, H. [2006]. *Violent breaking wave impacts. Part 1: Results from large-scale regular wave tests on vertical and sloping walls*, Coastal Engineering, Elsevier, 54, 602-617.

- [11] Capone, T., Panizzo, A., Cecioni, C. And Dalrymple, R. A. [2007]. Accuracy and stability of numerical schemes in SPH, *2nd International SPHERIC Workshop*, May 23-25, 2007, Madrid, Spain, pp. 156-160.
- [12] Canadian Association of Earthquake Engineering (CAEE) [2005]. Reconnaissance report on the December 26, 2004 Sumatra earthquake and tsunami, 21 pp.
- [13] Chanson, H. [2006]. *Tsunami surges on dry coastal plains: application of dam break equations*, Coastal Engineering Journal, Japanese Society of Civil Engineers (JSCE), 48(4), 355-370.
- [14] Chow, V. T. [1973]. *Open channel hydraulics*, McGraw-Hill International, New York, 680 pp.
- [15] Cleary, P. W. [1998]. *Modelling confined multi-material heat and mass flows using SPH*, Applied Mathematical Modelling, 22, 981-993.
- [16] Cleary, P. W. and Monaghan, J. J. [1999]. *Conduction modelling using smoothed particle hydrodynamics*, Journal of Computational Physics, 148, 227-264.
- [17] Colagrossi, A. and Landrini, M. [2003]. *Numerical simulation of interfacial flows by smoothed particles hydrodynamics*, Journal of Computational Physics, 191(2), 448-475.
- [18] Crespo, A. J. C., Gómez-Gesteira, B. D. and Dalrymple, R. A. [2008]. *Modeling dam break behaviour over a wet bed using the SPH technique*, Journal of Waterway, Port, Coastal and Ocean Engineering, 134(6), 313-320.
- [19] Crespo, A. J. C., Dominguez, J. M., Barreiro, A., Gómez-Gesteira, M. and Rogers, B. D. [2011]. GPUs, a new tool of acceleration in CFD: efficiency and reliability on smoothed particle hydrodynamics methods, PLoS ONE, 6(6), e20685.
- [20] Crowe, C. T., Elger, D. F. and Roberson, J. A. [2005]. *Engineering Fluid Mechanics*, 8th edition, John Wiley & Sons, 656 pp.
- [21] Cuomo, G., Panizzo, A. and Dalrymple, R. [2006]. SPH-LES two-phase simulation of wave breaking and wave-structure interaction, *30th International Conference on Coastal Engineering*, September 3-8, 2006, San Diego, California, USA, World Scientific Press, pp. 274-286.
- [22] Dalrymple, R. A. and Knio, O. [2001]. SPH modelling of water waves, *4th Conference on Coastal Dynamics*, June 11-15, 2001, Lund, Sweden, American Society of Civil Engineers (ASCE), pp. 749-778.
- [23] Dalrymple, R. A. and Rogers, B. D. [2006]. *Numerical modeling of water waves with the SPH method*, Coastal Engineering, 53, 141-147.

- [24] Dalrymple, R. A. [2007]. Particle Methods and Waves, with Emphasis on SPH, Department of Civil Engineering, Baltimore, 32 pp.
- [25] Dalrymple, R. A., Gómez-Gesteira, M., Rogers, B. D., Panizzo, A., Zou, S., Crespo, A. J. C., Cuomo, G. and Narayanaswamy, M. [2009]. *Smoothed particles hydrodynamics of water waves*, in Ma, Q., “Advances in numerical simulation of nonlinear water waves”, World Scientific, pp. 465.
- [26] Dalrymple, R. A., Hérault, A., Bilotta, G. and Farahani, R. J. [2010]. GPU-Accelerated SPH model for water waves and other free surface flows, 32nd *International Conference on Coastal Engineering*, Shanghai, China, CD-ROM.
- [27] Didier, E. and Neves, M. G. [2010]. A Lagrangian Smoothed Particles Hydrodynamics – SPH – method for modelling waves-coastal structure interaction, 5th *European Conference on Computational Fluid Dynamics*, June 14-17, 2010, Lisbon, Portugal, CD-ROM.
- [28] Drønen, N., Karunarathna, H., Fredsøe, J., Sumer, B. M. and Deigaard, R. [2002]. *An experimental study of rip channel flow*, Coastal Engineering, 45(3), 223-238.
- [29] Earthquake Engineering Research Institute (EERI) [2011]. The Tohoku, Japan, tsunami of March 11, 2011: effects on structures, Special Earthquake Report.
- [30] Ghobarah, A., Saatcioglu, M. and Nistor, I. [2006]. *The impact of the 26 December 2004 earthquake and tsunami on structures and infrastructure*, Engineering Structures, 28 (2), 312-326.
- [31] Gingold, R.A. and Monaghan, J.J. [1977]. *Smoothed particle hydrodynamics: theory and application to non-spherical stars*, Monthly Notices of the Royal Astronomical Society, 181, 375-389.
- [32] Gómez-Gesteira, M., and Dalrymple, R. A. [2004]. *Using a three-dimensional smoothed particle hydrodynamic method for wave impact on a tall structure*, Journal of Waterway, Port, Coastal and Ocean Engineering, 130(2), 63-69.
- [33] Gómez-Gesteira, M., Rogers, B. D., Dalrymple, R. A., Crespo, A. J. C. and Narayanaswamy, M. [2010a]. User guide for the SPHysics code v2.0, Available at: <http://www.sphysics.org>.
- [34] Gómez-Gesteira, M., Rogers, B.D., Dalrymple, R. A. and Crespo, A. J. C. [2010b]. *State-of-the-art of classical SPH for free-surface flows*, Journal of Hydraulic Research, 48 Extra Issue, 6-27.
- [35] Gong, K., Wang, B. and Liu, H. [2010]. Modelling water entry of a wedge by multiphase SPH method, 32nd *International Conference on Coastal Engineering*, Shanghai, China, CD-ROM.

- [36] Gotoh, H., Shao, S. and Memita, T. [2004]. *SPH-LES model for numerical investigation of wave interaction with partially immersed breakwater*, Coastal Engineering Journal, Japanese Society of Civil Engineers (JSCE), 46(1), 39-63.
- [37] Guilcher, P.M., Ducorzet, G., Alessandrini, B. and Ferrant, P. [2007]. Water wave propagation using SPH models, *2nd International SPHERIC Workshop*, May 23-25, 2007, Madrid, Spain, pp. 119-124.
- [38] Graham, D. I. and Hughes, J. P. [2007]. *Accuracy of SPH viscous flow models*, International journal for Numerical Methods in fluids, 56, 1261-1269.
- [39] Hattori, M., Arami, A. and Yui, T. [1994]. Wave impact pressure on vertical walls under breaking waves of various types, Coastal Engineering, 22, 79-114.
- [40] Hirsch, C. [1998]. *Numerical computation of internal and external flows*, John Wiley & Sons, New York, 656 pp.
- [41] Hu, X. Y. and Adams, N. A. [2009]. *A constant-density approach for incompressible multi-phase SPH*, Journal of Computational Physics, 228(6), 2082-2091.
- [42] Hughes, J. P. and Graham, D. I. [2010]. *Comparison of incompressible and weakly-compressible SPH for free-surface flows*, Journal of Hydraulic Research, 48 Extra Issue, 105-117.
- [43] Iizuka, H. and Matsutomi, H. [2010]. Damage due to the flooding flow of tsunami, *Proceedings of Coastal Engineering*, Japan Society of Civil Engineers (JSCE), 47, 381-385 (in Japanese).
- [44] Issa, R. [2004]. "Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flow and its extension to turbulent flows", PhD Thesis, University of Manchester Institute of Science and Technology, United-Kingdom, 199 pp.
- [45] Jánosi, I. M., Jan, D., Szabó, K. G. and Tél, T. [2004]. *Turbulent drag reduction in dam-break flows*, Experiments in Fluids, 37(2), 219-229.
- [46] Jiang, F., Oliveira, M. S. A. and Sousa, A. C. M. [2007], *Mesoscale SPH modeling of fluid flow in isotropic porous media*, Computer Physics Communications, 176, 471-480.
- [47] Johnson, G. R., Stryk, R. A. and Beissel, S. R. [1996]. *SPH for high velocity impact computations*, Computer Methods in Applied Mechanics and Engineering, 139, 347-373.
- [48] Jonsson, P., Jonsén, P., Andreasson, P., Lundström, T. S., Gunnar, J and Hellström, I. [2011], Smoothed particle hydrodynamics modeling of hydraulic

- jump, *II International Conference on Particle-based Methods (PARTICLES II)*, October 26-28, 2011, Barcelona, Spain, 12 pp.
- [49] Khayyer, A. and Gotoh, H. [2010]. *On particle-based simulation of a dam break over a wet bed*, Journal of Hydraulic Research, 48(2), 238-249.
 - [50] Kirkgöz, M. S. [1991]. Impact pressure of breaking waves on vertical and sloping walls, Ocean Engineering, 18, 45-59.
 - [51] Kleefsman, K. M. T., Fekken, G., Veldman, A. E. P., Iwanowski, B. and Buchner, B. [2005]. *A volume-of-fluid based simulation method for wave impact problems*, Journal of Computational Physics, 206, 363–393.
 - [52] Kyodo News [2012]. <http://english.kyodonews.jp/photos/2012/03/145839.html>, referred to on 2012-03-29.
 - [53] Landrini, M., Colagrossi, A., Greco, M. and Tulin, M. P. [2007]. *Gridless simulations of splashing process and near-shore bore propagation*, Journal of Fluid Mechanics, 591, 183-213.
 - [54] Lauber, G. and Hager, W. H. [1998]. *Experiments to dam-break wave: horizontal channel*, Journal of Hydraulic Research, 36(3), 291-307.
 - [55] Leal, J. G. A. B., Ferreira, R. M. L. and Cardoso, A. H. [2006]. *Dam-break wave-front celerity*, Journal of Hydraulic Engineering, 132(1), 69-76.
 - [56] Lee, E.-S., Violeau, D., Issa, R. and Ploix, S. [2010]. *Application of weakly compressible and truly incompressible SPH to 3-D water collapse in waterworks*, Journal of Hydraulic research, 48 Extra Issue, 50–60.
 - [57] Leimkuhler, B.J., Reich, S. and Skeel, R.D. [1996]. *Integration Methods for Molecular Dynamic*, in Mesirov, J.P., Schulten, K. and Sumners, D.W., “Mathematical Approaches to Biomolecular Structure and Dynamics”, IMA Volume in Mathematics and its application, Springer, Vol. 82, pp. 161-185.
 - [58] Liu, G. and Liu, M. [2003]. *Smoothed Particles Hydrodynamics: a meshfree method*, World Scientific, Singapore, 449 pp.
 - [59] Lo, E. and Shao, S. [2002]. Simulation of near-shore solitary wave mechanics by an incompressible SPH method, Applied Ocean Research, 24(5), 275-286.
 - [60] López, D., Marivela, R. and Garrote, L. [2010]. *Smoothed particle hydrodynamics model applied to hydraulic structures: a hydraulic jump test case*, Journal of Hydraulic Research, 48 Extra Issue, 142-158.
 - [61] Lucy, L.B. [1977]. *A numerical approach to the testing of the fission hypothesis*, The Astronomical Journal, 82, 1013-1024.

- [62] Lukkunaprasit, P., Ruangrassamee, A. and Thanasisathit, N. [2009]. *Tsunami loading on buildings with openings*, Science of Tsunami Hazard, 28(5), 303-310.
- [63] Maneti, S., Sibilla, S., Gallati, M., Agate, G. and Guandalini, R. [2012]. *SPH simulation of sediment flushing induced by a rapid water flow*, Journal of Hydraulic Engineering, 138(3), 272.
- [64] Maruzewski, P., Le Touzé, D., Oger, G. and Avellan, F. [2009]. *SPH high-performance computing simulations of rigid solids impacting the free-surface of water*, Journal of Hydraulic Research, 47 Extra Issue, 126-134.
- [65] Mikami, T., Shibayama, T. and Esteban, M. [2012]. *Field survey of the 2011 Tohoku Earthquake and Tsunami in Miyagi and Fukushima prefectures*, Coastal Engineering Journal, Japanese Society of Civil Engineers (JSCE), 54(1), 1250011.
- [66] Monaghan, J. J. and Lattanzio, J.C. [1985]. *A refined particle method for astrophysical problems*, Astronomy and Astrophysics, 149, 135-143.
- [67] Monaghan, J. J. [1989]. *On the problem of penetration in particle methods*, Journal of Computational Physics, 82, 1-15.
- [68] Monaghan, J. J. [1992]. *Smoothed particle hydrodynamics*, Annual Review of Astronomy and Astrophysics, 30, 543-574.
- [69] Monaghan, J. J. [1994]. *Simulating free surface flows with SPH*, Journal of Computational Physics, 110, 399-406.
- [70] Monaghan, J. J. and Kos, A. [1999]. *Solitary waves on a Cretan beach*, Journal of Waterway, Port, Coastal and Ocean Engineering, 125(3), 145-154.
- [71] Monaghan, J. J. [2000]. *SPH without a Tensile Instability*, Journal of Computational Physics, 159, 230-311.
- [72] Monaghan, J. J. [2005]. *Smoothed Particle Hydrodynamics*, Reports on Progress in Physics, 68(8), 1703-1759.
- [73] Morris, J. P., Fox, P. J. and Zhu, Y. [1997]. *Modeling low Reynolds number incompressible flow using SPH*, Journal of Computational Physics, 136, 214-226.
- [74] Morris, J.P., Zhu, Y. and Fox, P.J. [1999]. *Parallel simulation of pore-scale flow through porous media*, Computers and Geotechnics, 25, 227-246.
- [75] Nistor, I., Saatcioglu, M. and Ghobarah, A. [2005]. *The December 2004 earthquake and tsunami – hydrodynamic forces on physical infrastructure in Thailand and Indonesia, 2005 Canadian Coastal Engineering Conference*, Halifax, Canada, CD-ROM, 15 pp.

- [76] Nistor, I., Palermo, D., Nouri, Y., Murty, T. and Saatcioglu, M. [2009a]. *Tsunami forces on structures*, in Kim, Y. C., “Handbook of Coastal and Ocean Engineering”, World Scientific, pp. 261-286.
- [77] Nistor, I., Palermo, D., Al-Faesly, T. and Cornett, A. [2009b]. Modelling of tsunami-induced hydrodynamic forces on buildings, *33rd IAHR Biennial Congress*, Vancouver, BC, August 10-14, 2009, CD-ROM.
- [78] Nistor, I., Palermo, D., Al-Faesly, T. and Cornett, A. [2011]. Field investigation and experimental modelling of the tsunami-induced extreme hydrodynamic forces on structures, *34th IAHR World Congress*, 26 June – 1 July, 2011, Brisbane, Australia, CD-ROM.
- [79] Nouri, Y., Nistor, I. and Palermo, D. (2010). *Experimental investigation of tsunami impact on free standing structures*, Coastal Engineering Journal, JSCE, 52(1), 43-70.
- [80] Omidvar, P., Stansby, P. K., and Rogers, B. D. [2012]. *Wave body interaction in 2D using smoothed particle hydrodynamics (SPH) with variable particle mass*, International Journal for Numerical Methods in Fluids, 68(6), 686-705.
- [81] Palermo, D. and Nistor, I. [2008]. *Understanding tsunami risk to structures: a Canadian perspective*, Science of Tsunami Hazard: The International Journal of the Tsunami Society, 27(4), 1-11.
- [82] Parshikov, A. N. [1999]. *Application of a solution of the Riemann problem to the SPH method*, Computational Mathematics and Mathematical Physics, 39(7), 1173-1182.
- [83] Peregrine, D. H. [2003]. *Water-wave impact on walls*, Annual Review of Fluid Mechanics, 35, 23-43.
- [84] Pope, S. B. [2000]. *Turbulent Flows*, Cambridge University Press, 771 pp.
- [85] Ramsden, T. D. [1993]. “Tsunami forces on a vertical wall caused by long waves, bores, and surges on a dry bed”, PhD Thesis, California Institute of Technology, Pasadena, USA.
- [86] Ritter, A. [1892]. *Die fortpflanzung der wasserwellen*, Vereine Deutscher Ingenieure Zeitschrift, 36(2), 947-954 (in German).
- [87] Rogers, B. D., Dalrymple, R. A. and Stansby, P. K. [2008]. SPH Modeling of Floating Bodies in the Surf Zone, *31st International Coastal Engineering Conference*, August 31 to September 5, 2008, Hamburg, Germany, pp 205-215.
- [88] Rogers, B.D. and Dalrymple, R.A. [2008]. *SPH modeling of tsunami waves*, in Liu, P.L., Yeh, H. and Synolakis, C., “Advanced Numerical Models for tsunami

- waves and runup”, *Advances in Coastal and Ocean Engineering*, World Scientific, Vol. 10, pp. 75-100.
- [89] Rogers, B. D., Dalrymple, R. A. and Stansby, P. K. [2010]. *Simulation of caisson breakwater movement using 2-D SPH*, *Journal of Hydraulic Research*, 48 Extra Issue, 135-141.
 - [90] Ruffman, A. and Hann, V. [2006]. The Newfoundland tsunami of November 18, 1929: An examination of the twenty-eight deaths of the “South Coast Disaster”, *Newfoundland and Labrador Studies*, 21(1), 1719-1726.
 - [91] Saatcioglu, M., Ghobarah, A. and Nistor, I. [2006]. *Performance of structures in Indonesia during the 2004 Sumatra earthquake and tsunami*, *Engineering Spectra*, Earthquake Engineering Research Institute (EERI), 22, 295-319.
 - [92] Shao, S. [2005]. SPH simulation of solitary wave interaction with a curtain-type breakwater, *Journal of Hydraulic Research*, 43(4), 366-375.
 - [93] Shao, S. [2010]. Incompressible SPH flow model for wave interactions with porous media, *Coastal Engineering*, 57(3), 304-316.
 - [94] Shadloo, M. S., Zainali, A., Yildiz, M. and Suleman, A. [2011]. *A robust weakly compressible SPH method and its comparison with an incompressible SPH*, *International Journal for Numerical Methods in Engineering*, 89(8), 939-956.
 - [95] Silvester, T. B. and Cleary, P. W. [2006]. Wave-structure interaction using smoothed particle hydrodynamics, *5th CSIRO International Conference on CFD in the Process Industries*, December 13-15, 2006, Melbourne, Australia, CD-ROM.
 - [96] Stansby, P. K., Chegini, A. and Barnes, T. C. D. [1998]. *The initial stages of dam-break flow*, *Journal of Fluid Mechanics*, 374, 407-242.
 - [97] Stoker, J. J. [1957]. *Water waves: the mathematical theory with its applications*, John Wiley & Sons, 600 pp.
 - [98] Swegle, J. W., Hicks, D. L. and Attaway, J. W. [1995]. *Smoothed particle hydrodynamics sensibility analysis*, *Journal of Computational Physics*, 116, 123-134.
 - [99] Takahashi, S., Kuriyama, Y., Tomita, T., Kawai, Y., Arikawa, T., Tatsumi, D., and Negi, T. [2011]. *Urgent survey for 2011 Great East Japan Earthquake and Tsunami disaster in ports and coasts – part I (tsunami)*, English abstract of the technical note of the Port and Airport Research Institute (PARI), No. 1231, 9 pp.
 - [100] Thusyanthan, N. I. and Madabhushi, S. P. G. [2008]. *Tsunami wave loading on coastal houses: a model approach*, *Proceedings of the Institution of Civil Engineering (ICE)*, 161(2), 77-86.

- [101] Tomita, T., Imamura, F., Arikawa, T., Yasuda, T. and Kawata, Y. [2006]. *Damage caused by the 2004 Indian Ocean Tsunami on the south-western coast of Sri Lanka*, Coastal Engineering Journal, Japanese Society of Civil Engineers (JSCE), 48(2), 99-116.
- [102] Toro, E. F., Spruce, M. and Speares, W. [1994]. *Restoration of the contact surface in the HLL-Riemann solver*, Shock Waves, 4, 25-34.
- [103] Toro, E. F. [2001]. *Shock capturing methods for free surface shallow flows*, John Wiley & Sons, New York, 309 pp.
- [104] Ulrich, C. and Rung, T. [2010]. Multiphysics SPH for harbour and ocean engineering hydrodynamics, *5th European Conference on Computational Fluid Dynamics*, June 14-17, 2010, Lisbon, Portugal, CD-ROM.
- [105] United States Geological Survey (USGS) [2010]. http://earthquake.usgs.gov/earthquakes/world/events/1960_05_22.php, referred on 2012-03-29.
- [106] United States Geological Survey (USGS) [2011]. http://earthquake.usgs.gov/earthquakes/states/events/1964_03_28.php, referred on 2012-03-29.
- [107] United States Geological Survey (USGS) [2012a]. <http://earthquake.usgs.gov/earthquakes/eqinthenews/2004/us2004slav/#details>, referred on 2012-03-29.
- [108] United States Geological Survey (USGS) [2012b]. <http://earthquake.usgs.gov/earthquakes/eqinthenews/2011/usc0001xgp/#details>, referred on 2012-03-29.
- [109] Vacondio, R., Rogers, B. D., Stansby, P. K., Mignosa, P. and Feldman, J. [2011]. A dynamic particle coalescing and splitting scheme for SPH, *6th International SPHERIC Workshop*, June 8-10, 2011, Hamburg, Germany, pp. 93-100.
- [110] Vandamme, J. Zou, Q. and Reeve, D. E. [2011]. *Modeling floating object entry and exit using smoothed particle hydrodynamics*, Journal of Waterway, Port, Coastal and Ocean Engineering, 137(5), 213-225.
- [111] Vaughan, G. L., Healy, T. R., Bryan, K. R., Sneyd, A. D. and Gorman, R. M. [2007]. *Completeness, conservation and error in SPH for fluids*, International Journal for Numerical Methods in Fluids, 56, 37-62.
- [112] Vila, J.-P. [1999]. *On particle weighted methods and Smooth Particle Hydrodynamic*, Mathematical Models and Methods in Applied Sciences, 9(2), 161-209.

- [113] White, W. R. H. [1966]. *The Alaska Earthquake – Its effects in Canada*, Canadian Geographical Journal, 72(6), 210-219.
- [114] Yamamoto, Y., Takanashi, S., Hettiarachchi, S. and Samarawickrama, Y. [2006]. *Verification of the destruction mechanism of structures in Sri Lanka and Thailand due to the Indian Ocean tsunami*, Coastal Engineering Journal, Japanese Society of Civil Engineers (JSCE), 48(2), 117-146.
- [115] Yeh, H. and Petroff, C. [n.d.]. Mitigation of local tsunami effects, University of Washington, Seattle, USA.
- [116] Zhu, Y., Fox, P. J. and Morris, J. P. [1999]. *A pore-scale numerical model for flow through porous media*, International Journal for Numerical and Analytical methods in Geomechanics, 23, 881-904.
- [117] Zou, S. and Dalrymple, R. [2006]. Sediment suspension simulation under oscillatory flow with SPH-SPS method, *30th International Conference on Coastal Engineering*, September 3-8, 2006, San Diego, California, USA, World Scientific Press, pp. 2477-2484.
- [118] Zou, S. [2007]. “Coastal Sediment Transport Simulation by Smoothed Particle Hydrodynamics”, PhD Thesis, Department of Civil Engineering, Johns Hopkins University, USA.

Publications

The following are publications that were prepared during the completion of this thesis based on its content.

Journal Papers

- [1] St-Germain, P., Nistor, I., Shibayama, T. and Townsend, R. [2012]. *Numerical modeling of structures impacted by tsunami bores using SPH*, Journal of Waterways, Port, Coastal and Ocean Engineering, ASCE (submitted by invitation).
- [2] St-Germain, P., Nistor, I. and Townsend, R. [2012]. *Numerical modeling of the impact with structures of tsunami bores propagating on dry and wet beds using the SPH method*, International Journal of Protective Structures, 3(2), 221-255.

Refereed Conference Proceedings

- [1] St-Germain, P., Nistor, I. and Townsend, R. [2012]. Numerical modeling of tsunami-induced hydrodynamic forces on onshore structures using SPH, *33rd International Conference on Coastal Engineering*, ASCE, July 1-6, 2012, Santander, Spain.¹
- [2] St-Germain, P., Nistor, I., Palermo, D. and Townsend, R. [2011]. SPH modeling of extreme hydrodynamic forces on slender structures, *34th IAHR Biennial Congress*, June 26-July 1, 2011, Brisbane, Australia.²
- [3] St-Germain, P., Nistor, I. and Townsend, R. [2011]. SPH modeling of tsunami-induced hydrodynamic forces on columns, *20th Canadian Hydrotechnical Conference*, CSCE, June 14-17, 2011, Ottawa, Canada.

¹ The *Best Student Abstract* award was received for the paper submission to the conference.

² A *Best Paper Presentation* award was received for the presentation of the paper at the conference.

Appendix A Riemann Solver Description

The following is a general introduction to the Riemann problem along with a brief description of the Harten – Lax – van Leer – Contact (HLLC) approximate Riemann solver [Toro *et al.*, 1994] using a Monotone Upwind-centred Scheme for Conservation Laws (MUSCL) [Toro, 2001] with a general β -limiter [Hirsch, 1998], as mentioned in Section 2.3.5. Note that this description considers the tree-dimensional case and it is according to the SPHysics source code which implements a WCSPH model. For a full description of Riemann solvers in general, the reader is referred to Toro [2001].

The Riemann problem is defined as a discontinuity in space of a field variable, and as example here, a discontinuity in the density field (ρ) between a particle i and a particle j (Figure A.1-a). Theoretically, when such condition arises, a *shock wave* propagates in one direction (right) while a *rarefaction wave* propagates in the opposite direction (left) as shown in Figure A.1-b. The contact area of these two waves is referred here as the *middle wave*.

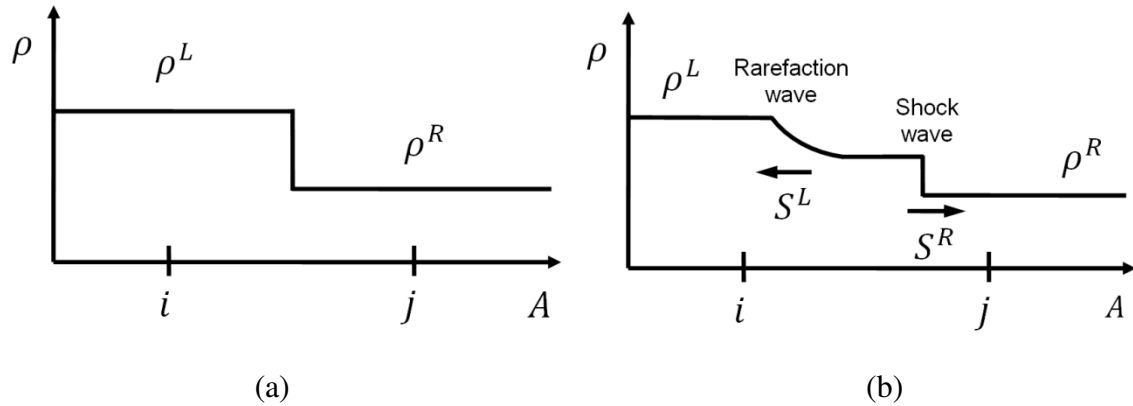


Figure A.1. Definition sketch of (a) initial Riemann problem between interacting particles i and j , and (b) subsequent propagating waves (based on Gómez-Gesteira *et al.* [2010a]).

Including the left (superscript L) and right (superscript R) states, Figure A.2 displays the two additional regions within which the field variables are assumed constant. The slope of each straight line in Figure A.2 represents to the speed (S) of its corresponding wave.

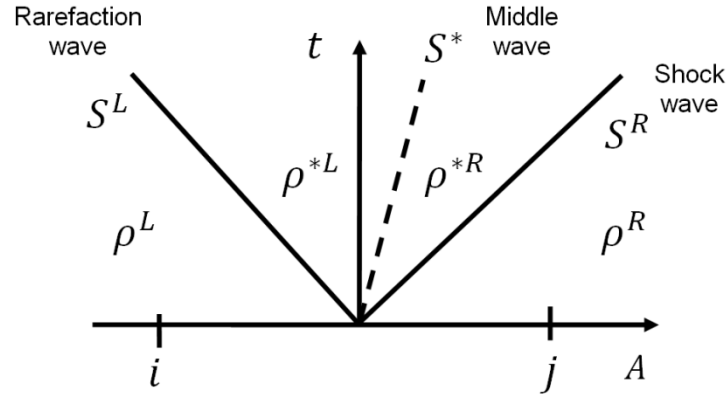


Figure A.2. Possible states of the Riemann problem shown in the A -time plane (based on Toro *et al.* [1994]).

In a nut shell, the main idea of the Riemann solver hereby is to determine from which state the field variables should be evaluated in order to be used in the reformulated governing equations (Eq. 2.104 and Eq. 2.105). To do so, for each field variable φ (e.g. ρ , u , v , and w), the first step consist in constructing the left and right state of the Riemann problem using a MUSCL scheme along with a specified slope limiter (β) as follows (u , v , and w are the components in the X, Y and Z direction of a velocity vector $\mathbf{u}$, respectively):

$$[A.1] \quad \varphi_i^L = \varphi_i + \overline{\Delta\varphi_i}$$

$$[A.2] \quad \varphi_j^R = \varphi_j - \overline{\Delta\varphi_j}$$

with

$$[A.3] \quad \overline{\Delta\varphi_i} \text{ and } \overline{\Delta\varphi_j} = \max[0, \min(\beta\Delta\varphi_i, \Delta\varphi_j), \min(\Delta\varphi_i, \beta\Delta\varphi_j)] \quad \text{if } \Delta\varphi_j \geq 0$$

or

$$[A.4] \quad \overline{\Delta\varphi_i} \text{ and } \overline{\Delta\varphi_j} = \min[0, \max(\beta\Delta\varphi_i, \Delta\varphi_j), \max(\Delta\varphi_i, \beta\Delta\varphi_j)] \quad \text{if } \Delta\varphi_j < 0$$

and

$$[A.5] \quad \text{if } \overline{\Delta\varphi_i} \text{ or } \overline{\Delta\varphi_j} > \frac{1}{2}|\varphi_i - \varphi_j|, \quad \text{then } \overline{\Delta\varphi_i} \text{ or } \overline{\Delta\varphi_j} = \frac{1}{2}|\varphi_i - \varphi_j|$$

where

$$[A.6] \quad \Delta\varphi_i = \nabla\varphi_i \cdot \frac{1}{2}(\mathbf{x}_j - \mathbf{x}_i)$$

$$[A.7] \quad \Delta\varphi_j = \nabla\varphi_j \cdot \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)$$

In the case where field variables constitute a vector quantity (i.e. velocity), the following step multiplies a transformation matrix (T) such that this vector is projected on the axis (A) aligning particle i and j (Figure A.3).

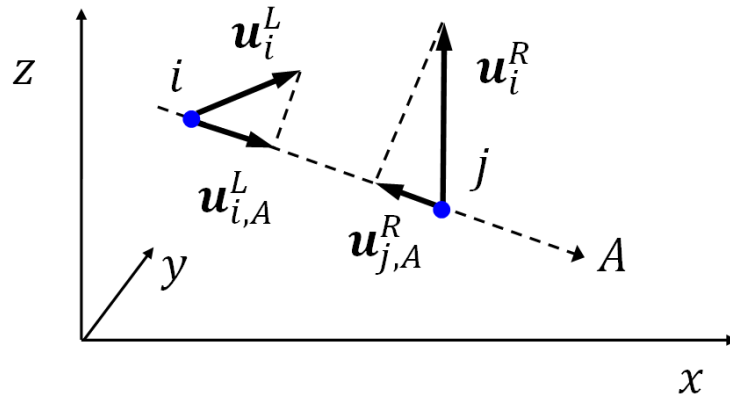


Figure A.3. Projections of the left (L) and right (R) states of vector field variables on the axis (A) aligning particle i and j (based on Parshikov [1999]).

$$[\text{A.8}] \quad \mathbf{u}_{i,A}^L = [T] \begin{bmatrix} u_i^L \\ v_i^L \\ w_i^L \end{bmatrix}$$

$$[\text{A.9}] \quad \mathbf{u}_{j,A}^R = [T] \begin{bmatrix} u_j^R \\ v_j^R \\ w_j^R \end{bmatrix}$$

with

$$\begin{aligned}
T_{11} &= -(x_i - x_j)/r_{ij}^2 \\
T_{12} &= -(y_i - y_j)/r_{ij}^2 \\
T_{13} &= \sin \phi \\
T_{21} &= -(\sin \psi)(\sin \phi)(\cos \theta) - (\cos \psi)(\sin \theta) \\
[\text{A.10}] \quad T_{22} &= -(\sin \psi)(\sin \phi)(\sin \theta) + (\cos \psi)(\cos \theta) \\
T_{23} &= (\sin \psi)(\cos \phi) \\
T_{31} &= -(\cos \psi)(\sin \phi)(\cos \theta) + (\sin \psi)(\sin \theta) \\
T_{32} &= -(\cos \psi)(\sin \phi)(\sin \theta) - (\sin \psi)(\cos \theta) \\
T_{33} &= (\cos \psi)(\cos \phi)
\end{aligned}$$

and

$$\begin{aligned}
\sin \phi &= -(z_i - z_j)/r_{ij}^2 \\
\cos \phi &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} / r_{ij}^2 \\
\sin \psi &= -(z_i - z_j) / \sqrt{(y_i - y_j)^2 + (z_i - z_j)^2} \\
[A. 11] \quad \cos \psi &= -(y_i - y_j) / \sqrt{(y_i - y_j)^2 + (z_i - z_j)^2} \\
\sin \theta &= -(y_i - y_j) / \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\
\cos \theta &= -(x_i - x_j) / \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}
\end{aligned}$$

where x , y and, z are the components in the X, Y and Z direction of a position vector $\mathbf{x}$. The next step consists in calculating the resulting values of the field variables along the axis A using an approximate HLLC Riemann solver. Considering the following column vector of the field variables

$$[A. 12] \quad \mathbf{Q} = \begin{bmatrix} \rho \\ u \\ v \\ w \end{bmatrix}$$

the solution of the Riemann problem on the axis A is defined as

$$[A. 13] \quad \mathbf{Q}_{\text{Solution},A} = \begin{cases} \mathbf{Q}_{L,A} & \text{if } S_{i,j} \leq S_L \\ \mathbf{Q}_{L,A}^* & \text{if } S_L < S_{i,j} \leq S^* \\ \mathbf{Q}_{R,A}^* & \text{if } S^* < S_{i,j} \leq S_R \\ \mathbf{Q}_{R,A} & \text{if } S_R < S_{i,j} \end{cases}$$

with

$$[A.14] \quad \mathbf{Q}_A^L = \begin{bmatrix} \rho_i^L \\ u_{i,A}^L \\ v_{i,A}^L \\ w_{i,A}^L \end{bmatrix}$$

$$[A.15] \quad \mathbf{Q}_A^{*R} = \rho_{i,A}^L \left(\frac{S^L - u_{i,A}^L}{S^L - S^*} \right) \begin{bmatrix} 1 \\ S^* \\ v_{i,A}^L \\ w_{i,A}^L \end{bmatrix}$$

$$[A.16] \quad \mathbf{Q}_A^{*R} = \rho_{j,A}^R \left(\frac{S^R - u_{j,A}^R}{S^R - S^*} \right) \begin{bmatrix} 1 \\ S^* \\ v_{j,A}^R \\ w_{j,A}^R \end{bmatrix}$$

$$[A.17] \quad \mathbf{Q}_A^R = \begin{bmatrix} \rho_j^R \\ u_{j,A}^R \\ v_{j,A}^R \\ w_{j,A}^R \end{bmatrix}$$

where $S_{i,j}$ is the average velocity of both particles i and j along the axis A calculated as

$$[A.18] \quad S_{i,j} = \left[\frac{u_i^L + u_j^R}{2} \quad \frac{v_i^L + v_j^R}{2} \quad \frac{w_i^L + w_j^R}{2} \right] \cdot [T_{11} \quad T_{12} \quad T_{13}]$$

and

$$[A.19] \quad S^L = u_{i,A}^L - c_i^L q_i^L$$

$$[A.20] \quad S^* = u^*$$

$$[A.21] \quad S^R = u_{j,A}^R + c_j^R q_j^R$$

for which

$$[A.22] \quad q_i^L = \begin{cases} 1 & \text{for } P^* \leq P_i^L \\ \left[\frac{\left(\frac{\rho_i^L}{\rho^*}\right) (P_i^L - P^*)}{(P_i^L - P^*)(c_i^L)^2} \right]^{1/2} & \text{for } P^* > P_i^L \end{cases}$$

$$[A.23] \quad q_j^R = \begin{cases} 1 & \text{for } P^* \leq P_j^R \\ \left[\frac{\left(\frac{\rho_j^R}{\rho^*}\right) (P_j^R - P^*)}{(P_j^R - P^*)(c_j^R)^2} \right]^{1/2} & \text{for } P^* > P_j^R \end{cases}$$

$$[A.24] \quad u^* = \frac{u_{i,A}^L + u_{j,A}^R}{2} + \frac{c_i^L + c_j^R}{\gamma - 1}$$

$$[A.25] \quad c^* = \frac{c_i^L + c_j^R}{2} + 0.25(\gamma - 1)(u_{i,A}^L - u_{j,A}^R)$$

$$[A.26] \quad \rho^* = \left(\frac{c^*}{c_0} \right)^{1/3}$$

where $\gamma = 7$ and P^* is calculated by the equation of state (Eq. 2.66) using ρ^* ; c_i^L and c_j^R are the speed of sounds of the left and right states, respectively, calculated using Eq. 2.69 with densities ρ_i^L and ρ_j^R , respectively. Lastly, the final step consist in converting the solution of the Riemann problem from the A axis back to the original coordinate system by using the transpose of matrix T , and to calculate the pressure by using the equation of state (Eq. 2.66) with the solution of the density.

$$[\text{A.27}] \quad \mathbf{u}_{ij}^* = [T]^T \begin{bmatrix} (\mathbf{Q}_{\text{solution},A})_{12} \\ (\mathbf{Q}_{\text{solution},A})_{13} \\ (\mathbf{Q}_{\text{solution},A})_{14} \end{bmatrix}$$

$$[\text{A.28}] \quad P_{ij}^* = \frac{c_0^2 \rho_0}{\gamma} \left[\left(\frac{\rho_{ij}^*}{\rho_0} \right)^\gamma - 1 \right]$$

with

$$[\text{A.29}] \quad \rho_{ij}^* = (\mathbf{Q}_{\text{solution},A})_{11}$$

Appendix B Additional Results of Sensitivity Analysis

This appendix contains additional graphical results of the sensitivity analysis performed to calibrate SPH-related parameters. In the case of Figure B.1 and Figure B.3, the values of the parameters considered for the base case are provided in Table B.1 below. Note that for all other graphical results presented hereby, the values considered for the base case are provided in Table 4.2.

Table B.1. SPH-related parameters considered for the base case of the sensitivity analysis results presented in Figure B.1 and Figure B.3.

Parameter	Value
Initial inter-particle spacing (Δ)	0.0300m
Slope limiter (β)	1.3
Smoothing length (h)	0.0368m
Speed of sound (c_0)	46.2m/s
Time Stepping algorithm	Symplectic
Time stepping coefficient (α_{time})	0.2
Smoothing kernel function	Quadratic
Kinematic viscosity (ν)	$1 \times 10^{-6} \text{ m}^2/\text{s}$
Wall viscosity (ν_{wall})	$8 \times 10^{-5} \text{ m}^2/\text{s}$
XSPH coefficient (ϵ)	0.5

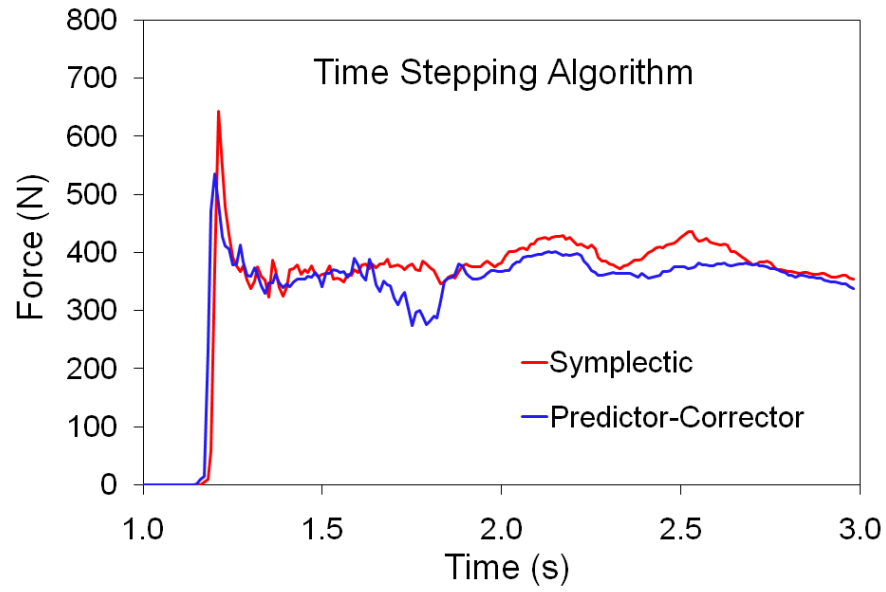


Figure B.1. Time-history of the net base horizontal force acting on the column using the symplectic (red) and with the predictor-corrector (blue) time stepping algorithms.

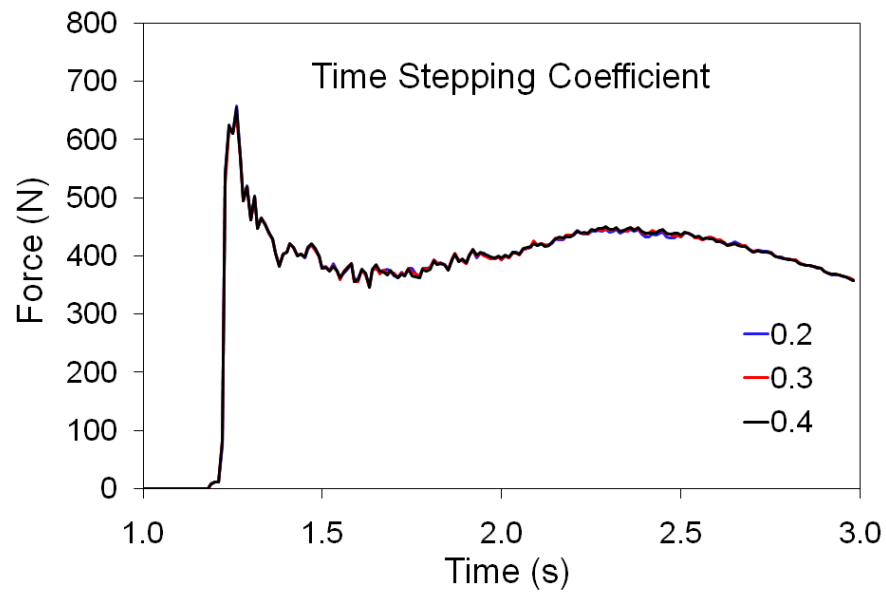


Figure B.2. Time-history of the net base horizontal force acting on the column for various time stepping coefficients (α_{time}).

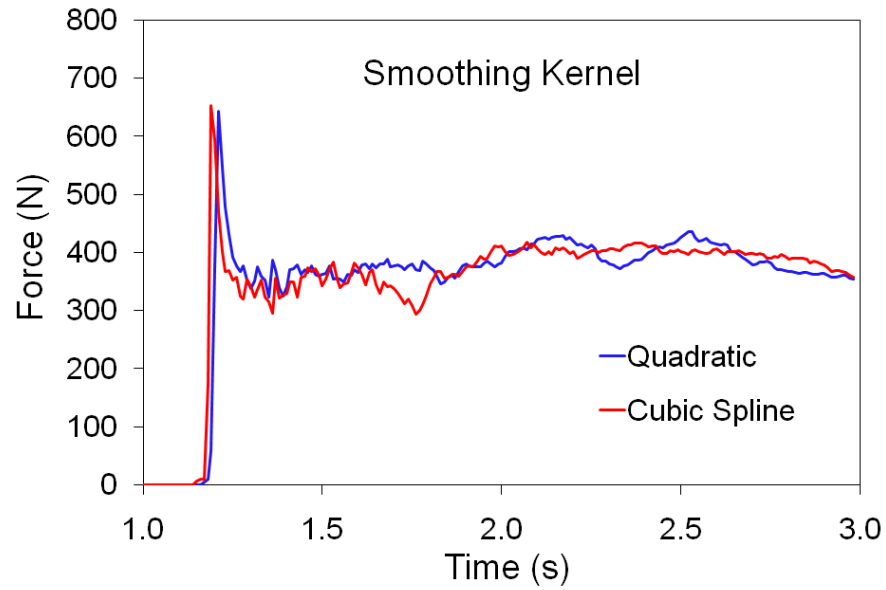


Figure B.3. Time-history of the net base horizontal force acting on the column using the quadratic smoothing kernel (blue) and the cubic spline smoothing kernel (red).

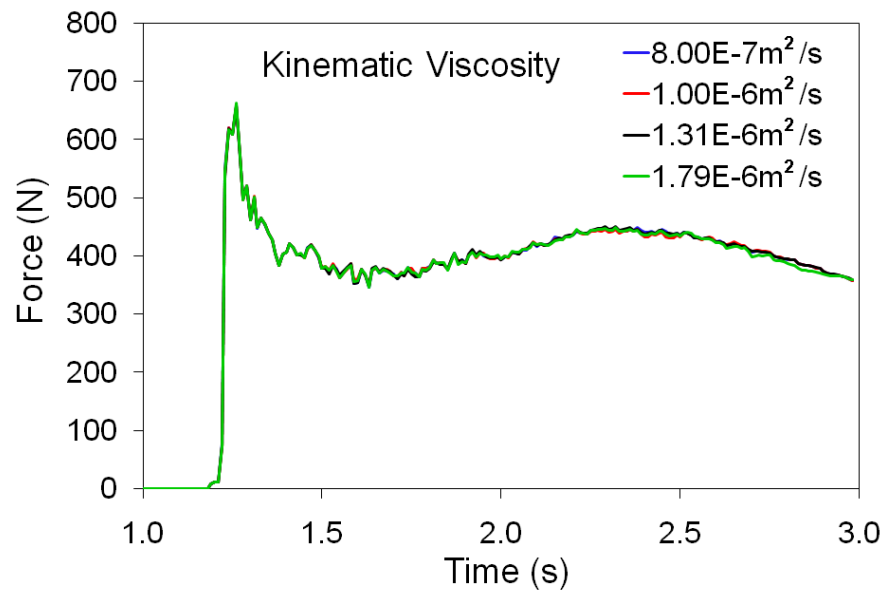


Figure B.4. Time-history of the net base horizontal force acting on the column for various kinematic viscosities of laminar flow (ν).

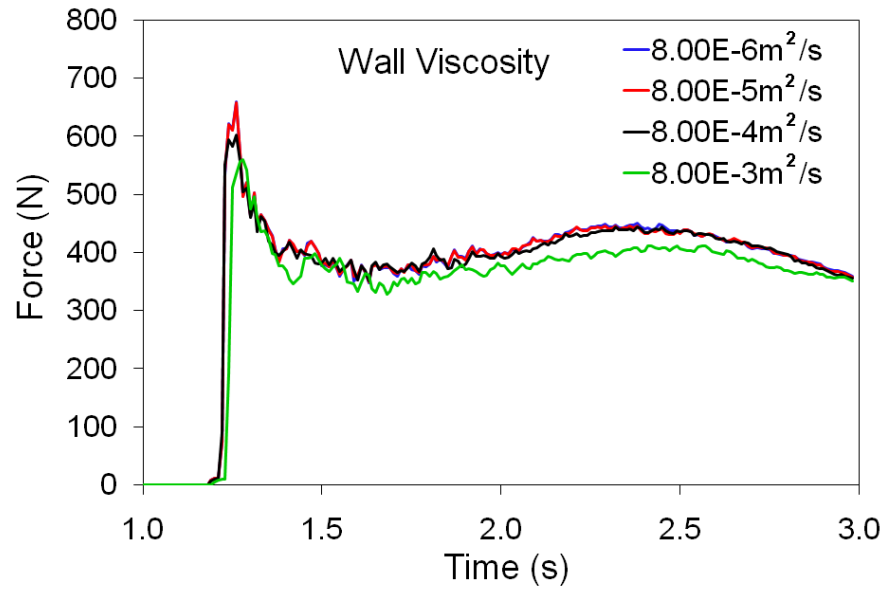
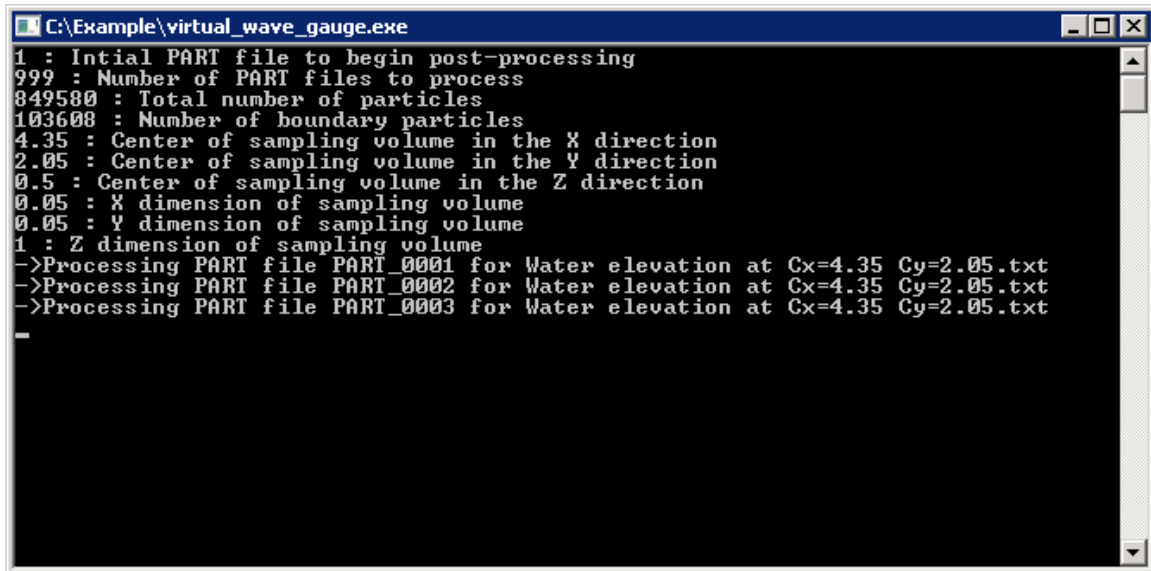


Figure B.5. Time-history of the net base horizontal force acting on the column for various wall viscosities (v_{wall}).

Appendix C Post-Processing Code

This appendix provides the source code of the post-processing programs that were developed in order to obtain specific results from raw numerical outputs by the approach of particle sampling. From a conceptual perspective, their implementation was discussed in Chapter 5. In total, four programs were developed in the C++ language using the object-oriented programming (OOP) paradigm. Only the three-dimensional case is considered hereby although the code can easily be simplified to the two-dimensional case where applicable. These programs were developed using the *Microsoft Visual Studio 2008* integrated development environment (IDE) on the Microsoft Server 2008 operating system (OS).

In each of the following sub-sections, guidance to the utilization of these programs is provided in addition to their source code. Note that the raw outputs of the numerical model SPHysics (or ParallelSPHysics) are called *PART* files. These are ASCII files are created at a specified frequency in which each indexed line contains the properties of a single particle in the simulation. They are labelled from *PART_0001* and so on, and the simulation time corresponding to each of these output files are found in a single output file called *DT*. Generally, other implementations of the SPH method generate outputs in a similar format. Hence, in order to use to following post-processing programs with other implementations, it is only required to modify the code section responsible for opening and reading the files containing the particles' properties. In addition, for each of the programs, their specified input parameters (discussed below) are presented in the command prompt (MS-DOS) window for verification once stated, and the index of the current *PART* file being analysed is displayed to provide feedback on the progress of the execution (Figure C.1). Lastly, note that all length and coordinate inputs mentioned in the following sections are in meters.



```
C:\Example\virtual_wave_gauge.exe
1 : Initial PART file to begin post-processing
999 : Number of PART files to process
849580 : Total number of particles
103608 : Number of boundary particles
4.35 : Center of sampling volume in the X direction
2.05 : Center of sampling volume in the Y direction
0.5 : Center of sampling volume in the Z direction
0.05 : X dimension of sampling volume
0.05 : Y dimension of sampling volume
1 : Z dimension of sampling volume
->Processing PART file PART_0001 for Water elevation at Cx=4.35 Cy=2.05.txt
->Processing PART file PART_0002 for Water elevation at Cx=4.35 Cy=2.05.txt
->Processing PART file PART_0003 for Water elevation at Cx=4.35 Cy=2.05.txt
```

Figure C.1. Sample screenshot of the feedback provided during the execution of the post-processing programs.

Virtual Wave Gauge

The program *virtual_wave_gauge.exe* provides the time-history of the water surface elevation at a specified location as explained in Section 5.2.2. Its input file containing the post-processing parameters must be an ASCII file called *wave_gauge_parameters.txt* and Figure C.2 shows an example. The first line (1) corresponds to the index (type integer) of the PART file from which post-processing will begin while the second line (2) corresponds to the index (type integer) at which it will end. With these parameters, only a portion of the simulation can be analysed as doing so for its entire length can require long computational times. The third line (3) corresponds to the total number (type integer) of particles in the simulation (boundary and water particles). The fourth line (4) is the number (type integer) of boundary particles. By providing the latter, this program and all the ones presented subsequently discard all boundary particles for the post-processing. Since all boundary particles are listed first in the PART files, the program only reads data when the lines corresponding to water particles are reached. The three following lines (5-6-7) corresponds the coordinates (type double) of the rectangular sampling volume in the x, y and z directions, respectively. The last three lines (8-9-10) correspond to the dimensions (type double) of the rectangular sampling volume in the x, y and z directions, respectively.

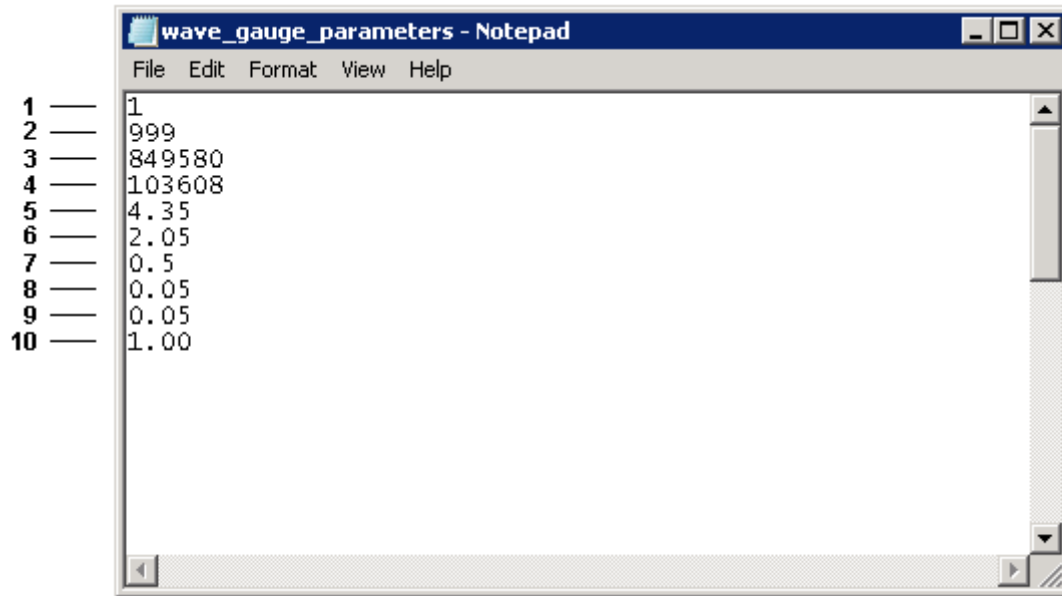


Figure C.2. Example of the content of an input file for the post-processing program *virtual_wave_gauge.exe* considering run D2-85.

Once the program terminates, an ASCII output file named for instance *Water elevation at Cx=4.35 Cy=2.05.txt* will be created. The name of this file provides the center coordinates in the x-y plane of the virtual wave gauge for easier bookkeeping. Its content is a single column of data corresponding the time-history of the water surface elevation over the specified range of PART files. Lastly, if no particle is present in the rectangular sampling volume (i.e. the virtual wave gauge) at a particular time, the resulting value will be indefinite (-1.#IND).

The following are the source files the program utilizes. Note that the main subroutine is located in the file *virtual_wave_gauge.cpp* and that the header files *particle.h* and *sampling_volume.h* are also used by all other programs subsequently presented.

particle.h

```
/******  
Code by Philippe St-Germain, University of Ottawa, 2012  
The author makes no warranty or guarantee, expressed or  
implied, as to the accuracy and reliability of the results  
derived from the use of this code. Its free distribution  
does not constitute any such warranty or guarantee. No  
responsibility or liability is assumed by the author.  
*****/  
#ifndef PARTICLE_H  
#define PARTICLE_H  
  
namespace post_processing {  
  
    class Particle {  
  
        public:  
            //Attributes associated to a water particle  
            double x_pos;  
            double y_pos;  
            double z_pos;  
            double x_vel;  
            double y_vel;  
            double z_vel;  
            double density;  
            double pressure;  
            double mass;  
  
    }; //End of class Particle  
}  
  
#endif
```

sampling_volume.h

```

/*****
Code by Philippe St-Germain, University of Ottawa, 2012
The author makes no warranty or guarantee, expressed or
implied, as to the accuracy and reliability of the results
derived from the use of this code. Its free distribution
does not constitute any such warranty or guarantee. No
responsibility or liability is assumed by the author.
*****/
#ifndef SAMPLING_VOLUME_H
#define SAMPLING_VOLUME_H

#ifndef PI
#define PI 3.141592654
#endif

#include "particle.h"
#include <math.h>
#include <iostream>
#include <vector>

namespace post_processing{

    using namespace std;

    class SamplingVolume{

    public:

        //Array of particles inside the sampling volume
        std::vector <Particle> particle_sample;

        //*****/
        //Variable related to a spherical sampling volume
        //for the kernel approximation of a flow property
        //*****/
        //Center coordinates of sampling volume
        double center_x;
        double center_y;
        double center_z;
        //Smoothing length of sampling volume
        double h;
        //Variable for the equation of state (Eq. 2.66):
        //B= (c0)^2 * rho0 / gamma
        double B;
        //*****/

        //*****/
        //Dimensions of rectangular sampling volume
        //for the determination of the water surface elevation
        //
        //OR
        //Dimensions of rectangular tributary area on a
        //rectangular solid object for the determination of
        //the resulting for by pressure integration
    }
}

```

```

//*****
double x_dim;
double y_dim;
double z_dim;
//*****

//*****
//Variable for sampling volumes involved in the
//determination of the resulting for by pressure
//integration
//*****
//Start angle of radial tributary area
double theta1;
//End angle of radial tributary area
double theta2;
//Distance from the center of the circular column to
//the center of the sampling volume
//      OR
//Distance from a center axis of a square column to t
//the center of the sampling volume
double rv;
//*****

Function for the kernel approximation of the velocity
//in the X direction
double avg_x_vel(){
    //For the correction of Bonet and Lok (Eq. 2.93)
    double sum1=0;
    double sum2=0;
    //Distance separating a fluid particle from the
    //center of the sampling volume
    double d;
    //Looping over all fluid particles inside the
    //sampling volume
    for(int i=0;i<particle_sample.size();i++){

        d=sqrt(pow(particle_sample[i].x_pos-center_x,2)+
        →      pow(particle_sample[i].y_pos-center_y,2)+
        →      pow(particle_sample[i].z_pos-center_z,2));

        sum1+=particle_sample[i].x_vel*
        →      particle_sample[i].mass*
        →      kernel(d)/
        →      particle_sample[i].density;

        sum2+=particle_sample[i].mass*
        →      kernel(d)/
        →      particle_sample[i].density;

        }//End of loop over all fluid particles
    return sum1/sum2;
}

//Function for the kernel approximation of the velocity
//in the Y direction (similar as function avg_x_vel)
double avg_y_vel(){
    double sum1=0;

```

```

double sum2=0;
double d;
for(int i=0;i<particle_sample.size();i++){

    d=sqrt(pow(particle_sample[i].x_pos-center_x,2)+
→      pow(particle_sample[i].y_pos-center_y,2)+
→      pow(particle_sample[i].z_pos-center_z,2));

    sum1+=particle_sample[i].y_vel*
→      particle_sample[i].mass*
→      kernel(d)/
→      particle_sample[i].density;

    sum2+=particle_sample[i].mass *
→      kernel(d)/
→      particle_sample[i].density;

}
return sum1/sum2;
} //End of function avg_y_vel

//Function for the kernel approximation of the velocity
//in the Z direction (similar as function avg_x_vel)
double avg_z_vel(){
    double sum1=0;
    double sum2=0;
    double d;
    for(int i=0;i<particle_sample.size();i++){

        d=sqrt(pow(particle_sample[i].x_pos-center_x,2)+
→      pow(particle_sample[i].y_pos-center_y,2)+
→      pow(particle_sample[i].z_pos-center_z,2));

        sum1+=particle_sample[i].z_vel*
→      particle_sample[i].mass*
→      kernel(d)/
→      particle_sample[i].density;

        sum2+=particle_sample[i].mass*
→      kernel(d)/
→      particle_sample[i].density;

    }
    return sum1/sum2;
} //End of function avg_z_vel

//Function for the calculation of the pressure
//assumed to be acting at the center of the
//sampling volume
double avg_pressure(){
    //Equation of state (Eq. 2.66)
    return B*(pow(avg_density()/1000.0,7)-1);
}

//Function for the kernel approximation of the density
double avg_density(){
    //For the correction of Bonet and Lok (Eq. 2.93)

```

```

double sum1=0;
double sum2=0;
//Distance separating a fluid particle from the
//center of the sampling volume
double d;
//Looping over all fluid particles inside the
//sampling volume
for(int i=0;i<particle_sample.size();i++){

    d=sqrt(pow(particle_sample[i].x_pos-center_x,2)+
→         pow(particle_sample[i].y_pos-center_y,2)+
→         pow(particle_sample[i].z_pos-center_z,2));

    sum1+=particle_sample[i].mass*kernel(d);

    sum2+=particle_sample[i].mass/
→         particle_sample[i].density*
→         kernel(d);

    }//End of loop over all fluid particles
return sum1/sum2;
};//End of function avg_density

//Quadratic kernel function (Eq. 2.65)
//Takes as parameter the distance from of the
//water particle from the center of the sampling volume
double kernel(double d){
    double r;
    r=d/h;
    double alpha_d;
    alpha_d=5/(4*PI*pow(h,3));
    if(0<=r&& r<= 2){
        return alpha_d*
→         (3.0/16.0*pow(r,2)-3.0/4.0*r+3.0/4.0);
    }
    else{
        return 0;
    }
};//End of function kernel

//Function for the determination of the elevation of the
//highest water particle inside the sampling volume
//Note that the function parameters are passed by
//reference
void max_elev(double &elev){
    //Setting the maximum elevation to zero
    //initially
    elev=0;
    //Looping over all water particles inside the
    //sampling volume. If a higher elevation is
    //found, the elev variable is updated.
    for(int i=0;i<particle_sample.size();i++){
        if(particle_sample[i].z_pos>elev){
            elev=particle_sample[i].z_pos;
        }
    }//End of loop over all fluid particles
};//End of Function max_elevation

```



```
}; //End of class SamplingVolume  
}  
#endif
```

virtual_wave_gauge.cpp

```
/*
*****
Code by Philippe St-Germain, University of Ottawa, 2012
The author makes no warranty or guarantee, expressed or
implied, as to the accuracy and reliability of the results
derived from the use of this code. Its free distribution
does not constitute any such warranty or guarantee. No
responsibility or liability is assumed by the author.
*****
*/
#ifndef WAVE_GAUGE
#define WAVE_GAUGE

#include "particle.h"
#include "sampling_volume.h"

#include <math.h>
#include <string.h>
#include <vector>
#include <iostream>
#include <string>
#include <sstream>
#include <fstream>
#include <cstdlib>

namespace post_processing {

    using namespace std;

    void sampleParticle(int np, int nb, char* filename,
→ SamplingVolume &sampling_volume) {

        //Limits of the rectangular sampling volume
        double x_min=sampling_volume.center_x-
→ sampling_volume.x_dim/2;
        double x_max=sampling_volume.center_x+
→ sampling_volume.x_dim/2;
        double y_min=sampling_volume.center_y-
→ sampling_volume.y_dim/2;
        double y_max=sampling_volume.center_y+
→ sampling_volume.y_dim/2;
        double z_min=sampling_volume.center_z-
→ sampling_volume.z_dim/2;
        double z_max=sampling_volume.center_z+
→ sampling_volume.z_dim/2;

        //Current particle being sampled
        Particle currentParticle;
        //Single line of the PART file
        string line;
        char *charLine;
        //Position of the current particle
        double x_pos;
        double y_pos;
        double z_pos;
    }
}
```

```

//Opening the PART file for reading
ifstream infile;
infile.open(filename);

//Looping over all lines of the PART file
int nn=0;
while(nn<np){

    //Making sure that the end of the PART file
    //is not reached
    if(!infile.eof()){

        //Reading a line
        getline(infile,line);

        //Adding a dummy token at the end of the string
        //in order to avoid an error with the atof function
        //when an empty line is read
        line =line + " 1";

        //If not an empty line
        if(!(line==" 1")){

            //Only sampling water particles and not
            //boundary partilces
            if(nn>nb){

                //Decomposing the line into an array of
                //characters
                charLine = new char[line.length()];
                for(unsigned int i=0;i< line.length();i++){
                    charLine[i] = line[i];
                }

                //Determination of the fluid particle position
                x_pos=atof(strtok (charLine," "));
                y_pos=atof(strtok (NULL," "));
                z_pos=atof(strtok (NULL," "));

                //Checking if the particle is inside the
                //retangular sampling volume
                if(((x_min<x_pos)&&(x_pos<x_max))&&
→          ((y_min<y_pos)&&(y_pos<y_max))&&
→          ((z_min<z_pos)&&(z_pos<z_max))) {

                    currentParticle.x_pos=x_pos;
                    currentParticle.y_pos=y_pos;
                    currentParticle.z_pos=z_pos;
                    currentParticle.x_vel=atof(strtok (NULL," "));
                    currentParticle.y_vel=atof(strtok (NULL," "));
                    currentParticle.z_vel=atof(strtok (NULL," "));
                    currentParticle.density=atof(strtok (NULL," "));
                    currentParticle.pressure=atof(strtok (NULL," "));
                    currentParticle.mass=atof(strtok (NULL," "));

```

```

→         sampling_volume.particle_sample.push_back
                                   (currentParticle);

        }

        //Deleting pointer
        delete[] charLine;
        charLine=0;
    } //End of if water particle

    } //end of if not empty line
} //end of if not end of file

nn++;

} //end of loop over all particles

//Closing the PART file
infile.close();

} //End of sampleParticle function

void printMaxElev(SamplingVolume sampling_volume,
→         vector<string> &elev_time_history,
→         int part_index, int num_part,
→         string outFileName){

    //Determination of the maximum elevation
    double elevation;
    sampling_volume.max_elev(elevation);

    //Converting the maximum elevation into a string
    string elevationString;
    ostringstream ee;
    ee<<elevation;
    elevationString = ee.str();

    //Inserting the maximum elevation at the in the
    //array of the time-history of this one
    elev_time_history.push_back(elevationString);

    //If all the PART files have been processed,
    //the final output file is written
    if(part_index==num_part ){

        //Creating the output file containg the time-history
        //of the water surface elevation
        char* result_filename;
        result_filename=0;
        result_filename = new char[outFileName.length() + 1];
        strcpy(result_filename,outFileName.c_str());
        ofstream outfile(result_filename);

        //Writing the entire time-history to the output file
        for(int i=0;i<elev_time_history.size();i++){
            outfile<<elev_time_history[i]<<"\n";
        }
    }
}

```

```

        //Closing the output file
        outfile.close();

        //Deleting pointer
        delete [] result_filename;
        result_filename=0;
    }

} // End of printMaxElev function
}

//*****
//Main program
//*****
using namespace post_processing;

int main(int argc, char** argv){

    //Variables for reading the raw PART files
    char* in_filename;
    string in_filenameTemp;
    string indexString;

    //Array for the time-history of the water surface elevation
    vector<string> elevation_time_history;

    //*****
    //Input parameters
    //*****

    //Opening file containing input parameters
    ifstream infile;
    infile.open("wave_gauge_parameters.txt");

    //Index of the intial PART file to begin post-processing
    int ini_part;
    //Number of PART files to be processed beyond initial
    //PART file
    int num_part;
    //Total number of particles in the simulation
    int np;
    //Number of boundary particles in the simulation
    int nb;
    //Coordinates of the center of rectangular
    //sampling volume
    double center_x;
    double center_y;
    double center_z;
    //Dimansions of the rectangular sampling colume
    double x_dim;
    double y_dim;
    double z_dim;

```

```

//Reading in the inputs
infile>>ini_part;
infile>>num_part;
infile>>np;
infile>>nb;
infile>>center_x;
infile>>center_y;
infile>>center_z;
infile>>x_dim;
infile>>y_dim;
infile>>z_dim;

//Closing the input file
infile.close();

//Writing the inputs to the command prompt window
cout<<ini_part<<" : Intial PART file to begin post-processing"<<"\n";
cout<<num_part<<" : Number of PART files to process"<<"\n";
cout<<np<<" : Total number of particles"<<"\n";
cout<<nb<<" : Number of boundary particles"<<"\n";
cout<<center_x<<" : Center of sampling volume in the X
→          direction"<<"\n";
cout<<center_y<<" : Center of sampling volume in the Y
→          direction"<<"\n";
cout<<center_z<<" : Center of sampling volume in the Z
→          direction"<<"\n";
cout<<x_dim<<" : X dimension of sampling volume"<<"\n";
cout<<y_dim<<" : Y dimension of sampling volume"<<"\n";
cout<<z_dim<<" : Z dimension of sampling volume"<<"\n";
//*****

//Name of the output file containing the post-processing
//results
string outFileName;
ostringstream cx;
cx<<center_x;
ostringstream cy;
cy<<center_y;
outFileName="Water elevation at Cx="
→          + cx.str() + " Cy="+cy.str()+ ".txt";

//SamplingVolume object
SamplingVolume samplingVolume;

//Initializing the properties of the SamplingVolume
//object
samplingVolume.center_x=center_x;
samplingVolume.center_y=center_y;
samplingVolume.center_z=center_z;
samplingVolume.x_dim =x_dim;
samplingVolume.y_dim=y_dim;
samplingVolume.z_dim=z_dim;

//Looping over PART files
for(int i=ini_part;i<=num_part;i++){

```

```

//*****
//Generation of the string corresponding to the current
//PART file to be processed
//*****
ostreamstream ii;
ii<<i;
indexString=ii.str();

if (i <10){
    in_filenameTemp="PART_000"+indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else if (i<100){
    in_filenameTemp="PART_00"+indexString;
    in_filename = new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else if (i<1000){
    in_filenameTemp ="PART_0" +indexString;
    in_filename = new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else{
    in_filenameTemp="PART_" +indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
//*****

//Calling the function that will search the current PART
//file and construct the list of particle located inside
//the sampling volume
sampleParticle(np,nb,in_filename,samplingVolume);

//Calling the function that will write the results to
//the output file
printMaxElev(samplingVolume,elevation_time_history,i,
→          num_part,outFileName);

//Progress feedback on the command prompt window
cout<<"->Processing file "<<in_filenameTemp<<" for
→      "<<outFileName<<"\n";

//Clearing the sampled particles
samplingVolume.particle_sample.clear();

//Deleting pointer
delete [] in_filename;
in_filename=0;

} //End of looping over PART files
return 0;
} //End of main program
//*****
#endif

```

Kernel Approximation at Single Point

The program *kernel_approximation_single_point.exe* provides the time-history of a specified flow property at a single point in the computational domain as explained in Section 5.2. The ASCII input file containing the post-processing parameters for this program must be called *kernel_approx_single_point_parameters.txt* and Figure C.3 shows an example. Up to the seventh line (1-7), the input parameters are the same ones as for the *virtual_wave_gauge.exe* program. The eighth line (8) contains the smoothing length h (type double) of the spherical sampling volume, while the ninth line (9) corresponds to the flow property (type string) to be approximated at the center of this sampling volume. The possible options for the latter are (exactly written) *PRESSURE*, *X_VELOCITY*, *Y_VELOCITY*, and *Z_VELOCITY*. If the approximation of the pressure is selected, a tenth line (10) is required containing the value corresponding to $(c_0^2 \rho_0)/\gamma$ (see equation of state given by Eq. 2.66).

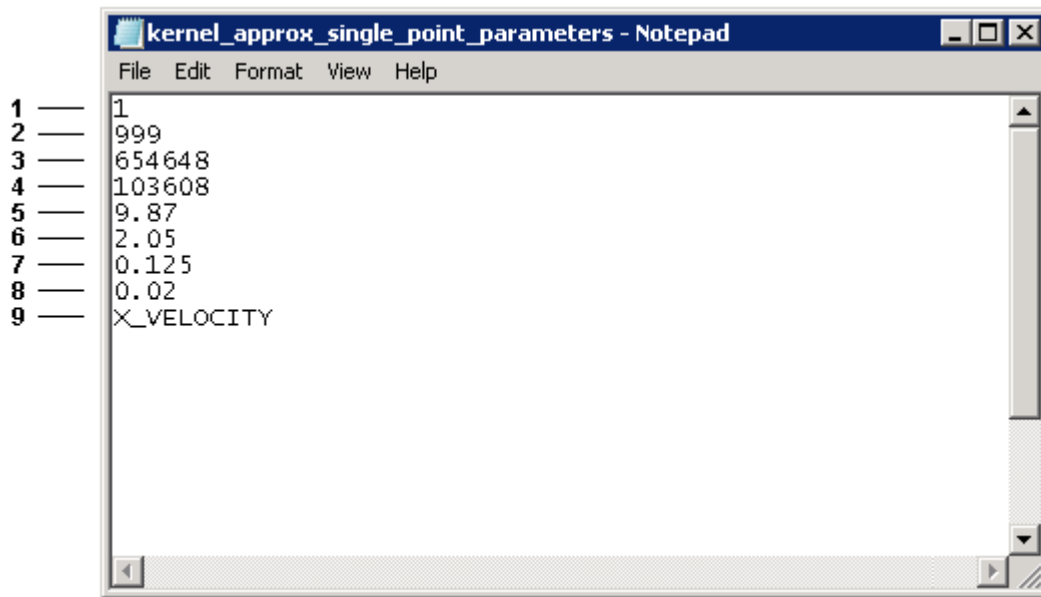


Figure C.3. Example of the content of an input file for the post-processing program *kernel_approximation_single_point.exe* considering run D2-85.

Once the program terminates, an ASCII output file named for instance *X velocity at Cx=9.87 Cy=2.05 Cz=0.125 with h=0.02.txt* will be created. The name of this file

provides the center coordinates of the spherical sampling volume, and its content is a single column of data corresponding the time-history of the flow property approximated over the specified range of PART files. If this property was the pressure, the units of the results are Pascals (Pa or N/m^2) and if the latter was the velocity in either the x, y, or z direction, the units of the results are m/s . Lastly, if no particle is present in the spherical sampling volume at a particular instant, the resulting value will be indefinite ($-1.#IND$).

The following is the source file the program utilizes in addition to the header files *particle.h* and *sampling_volume.h* presented earlier. Note that the main subroutine is located in the source file below.

kernel_approximation_single_point.cpp

```
/*
Code by Philippe St-Germain, University of Ottawa, 2012
The author makes no warranty or guarantee, expressed or
implied, as to the accuracy and reliability of the results
derived from the use of this code. Its free distribution
does not constitute any such warranty or guarantee. No
responsibility or liability is assumed by the author.
*/
#ifndef KERNEL_APPROX
#define KERNEL_APPROX

#include "particle.h"
#include "sampling_volume.h"

#include <math.h>
#include <string.h>
#include <vector>
#include <iostream>
#include <string>
#include <sstream>
#include <fstream>
#include <cstdlib>

namespace post_processing {

    using namespace std;

    void sampleParticle(int np, int nb, char* filename, SamplingVolume
→      &samplingVolume) {

        //Current particle being sampled
        Particle currentParticle;
        //Single line of the PART file
        string line;
        char *charLine;
        //Position of the current particle
        double x_pos;
        double y_pos;
        double z_pos;

        //Distance between the current particle and
        //the center of the sampling volume
        double d;

        //Opening the PART file for reading
        ifstream infile;
        infile.open(filename);

        //Looping over all lines of the PART file
        int nn=0;
        while(nn<np) {
            //Making sure that the end of the PART file
            //is not reached

```

```

if(!infile.eof()){

    //Reading a line
    getline(infile,line);
    //Adding a dummy token at the end of the string
    //in order to avoid an error with the atof function
    //when an empty line is read
    line =line + " 1";
    //If not an empty line
    if(!(line==" 1")){
        //Only sampling water particles and not
        //boundary partilces
        if(nn>nb){

            //Decomposing the line into an array of
            //characters
            charLine = new char[line.length()];
            for(unsigned int i=0;i<line.length();i++){
                charLine[i] = line[i];
            }

            //Determination of the fluid particle position
            x_pos=atof(strtok (charLine, " "));
            y_pos=atof(strtok (NULL, " "));
            z_pos=atof(strtok (NULL, " "));

            //Computing the current particle distance
            d=sqrt(pow(((x_pos)-samplingVolume.center_x),2)+
                pow(((y_pos)-samplingVolume.center_y),2)+
                pow(((z_pos)-samplingVolume.center_z),2));

            //Checking if the particle is inside the
            //retangular sampling volume
            if(d<=(2*samplingVolume.h)){

                Particle currentParticle;

                currentParticle.x_pos=x_pos;
                currentParticle.y_pos=y_pos;
                currentParticle.z_pos=z_pos;
                currentParticle.x_vel=atof(strtok (NULL, " "));
                currentParticle.y_vel=atof(strtok (NULL, " "));
                currentParticle.z_vel=atof(strtok (NULL, " "));
                currentParticle.density=atof(strtok (NULL, " "));
                currentParticle.pressure=atof(strtok (NULL, " "));
                currentParticle.mass=atof(strtok (NULL, " "));

                samplingVolume.particle_sample.push_back
                    (currentParticle);
            }
            //Deleting pointer
            delete[] charLine;
            charLine=0;
        } //End of if water particle
    } //end of if not empty line
} //end of if not end of file

```

```

        nn++;

    } //end of loop over all particles

    //Closing the PART file
    infile.close();

} //End of sampleParticle function

void printTimeHistory(SamplingVolume &samplingVolume,
    vector<string> &time_history,
    int part_index, int num_part,
    string outFileName,
    string propertyToAnalyze){

    //Approximation of the flow property
    double measurement;
    string measurementString;
    ostringstream pp;
    if(propertyToAnalyze=="PRESSURE" ){
        measurement=samplingVolume.avg_pressure();
    }
    if(propertyToAnalyze=="X_VELOCITY" ){
        measurement=samplingVolume.avg_x_vel();
    }
    if(propertyToAnalyze=="Y_VELOCITY" ){
        measurement=samplingVolume.avg_y_vel();
    }
    if(propertyToAnalyze=="Z_VELOCITY" ){
        measurement=samplingVolume.avg_z_vel();
    }
    pp<<measurement;
    measurementString=pp.str();
    time_history.push_back(measurementString);

    //If all the PART files have been processed,
    //the final output file is written
    if(part_index==num_part){

        //Creating the output file containg the time-history
        //of the water surface elevation
        char* result_filename;
        result_filename=0;
        result_filename = new char[outFileName.length() + 1];
        strcpy(result_filename,outFileName.c_str());
        ofstream outfile(result_filename);

        //Writing the entire time-history to the output file
        for (int i=0;i<time_history.size();i++){
            outfile<<time_history[i]<<"\n";
        }

        //Closing the output file
        outfile.close();

        //Deleting pointer
        delete [] result_filename;
    }
}

```

```

        result_filename=0;
    }
} // End of printTimeHistory function

}

//*****
//Main program
//*****
using namespace post_processing;

int main(int argc, char** argv){

    //Variables for reading the raw PART files
    char* in_filename;
    string in_filenameTemp;
    string indexString;

    //Array for the time-history of the measurement
    vector <string> time_history;

    //*****
    //Input parameters
    //*****

    //Opening file containing input parameters
    ifstream infile;
    infile.open("kernel_approx_single_point_parameters.txt");

    //Index of the initial PART file to begin post-processing
    int ini_part;
    //Number of PART files to be processed beyond initial
    //PART file
    int num_part;
    //Total number of particles in the simulation
    int np;
    //Number of boundary particles in the simulation
    int nb;
    //Initial coordinates of the center of spherical sampling volume
    double center_x;
    double center_y;
    double center_z;
    //Smoothing length of sampling volume
    double h;
    //Variable for the equation of state (Eq. 2.66):
    //B = (c0)^2 * rho0 / gamma
    double B;
    //Property to analyze for (PRESSURE, X_VELOCITY,
    //Y_VELOCITY, or Z_VELOCITY)
    string propertyToAnalyze;

    //Reading in the inputs
    infile>>ini_part;
    infile>>num_part;
    infile>>np;
    infile>>nb;
    infile>>center_x;

```

```

infile>>center_y;
infile>>center_z;
infile>>h;
infile>>propertyToAnalyze;
if(propertyToAnalyze=="PRESSURE"){
    infile >> B;
}

//Closing the input file
infile.close();

//Writing the inputs to the command prompt window
cout<<ini_part<<" : Initial PART file to begin post-processing"<<"\n";
cout<<num_part<<" : Number of PART files to process"<<"\n";
cout<<np<<" : Total number of particles"<<"\n";
cout<<nb<<" : Number of boundary particles"<<"\n";
cout<<center_x<<" : Center of sampling volume in the X
→           direction"<<"\n";
cout<<center_y<<" : Center of sampling volume in the Y
→           direction"<<"\n";
cout<<center_z<<" : Center of sampling volume in the Z
→           direction"<<"\n";
cout<<h<<" : Smoothing length"<<"\n";
cout<<propertyToAnalyze<<" : Flow property to approximate"<<"\n";
if (propertyToAnalyze=="PRESSURE"){
    cout<<B<<" : B = (c0)^2 * rho0 / gamma"<<"\n";
}
//*****

//Name of the output file containing the post-processing
//results
string outFileName;
ostringstream cx;
cx << center_x;
ostringstream cy;
cy << center_y;
ostringstream cz;
cz << center_z;
ostringstream h_temp;
h_temp<<h;

if(propertyToAnalyze=="PRESSURE"){
    outFileName="Pressure at Cx="+
→           cx.str()+" Cy="+cy.str()+
→           " Cz="+cz.str()+" with h="+h_temp.str()
→           +".txt";
}
if(propertyToAnalyze=="X_VELOCITY"){
    outFileName="X velocity at Cx="+
→           cx.str()+" Cy="+cy.str()+
→           " Cz="+cz.str()+" with h="+h_temp.str()
→           +".txt";
}
if(propertyToAnalyze=="Y_VELOCITY"){
    outFileName="Y velocity at Cx="+
→           cx.str()+" Cy="+cy.str()+
→           " Cz="+cz.str()+" with h="+h_temp.str()

```

```

→         +".txt";
    }
    if(propertyToAnalyze=="Z_VELOCITY"){
        outFileName="Z velocity at Cx="+
→         cx.str()+" Cy="+cy.str()+
→         " Cz="+cz.str()+" with h="+h_temp.str()
→         +".txt";
    }

    //SamplingVolume object
    SamplingVolume samplingVolume;

    //Initializing the properties of the SamplingVolume
    //object
    samplingVolume.center_x =center_x;
    samplingVolume.center_y = center_y;
    samplingVolume.center_z = center_z;
    samplingVolume.h = h;
    if(propertyToAnalyze=="PRESSURE"){
        samplingVolume.B=B;
    }

    //Looping over PART files
    for (int i=ini_part; i<= num_part; i++){

        //*****
        //Generation of the string corresponding to the current
        //PART file to be processed
        //*****
        ostringstream ii;
        ii<<i;
        indexString=ii.str();

        if (i<10){
            in_filenameTemp="PART_000"+indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else if (i<100){
            in_filenameTemp = "PART_00"+indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else if (i<1000){
            in_filenameTemp="PART_0" +indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else{
            in_filenameTemp="PART_" +indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        //*****

        //Calling the function that will search the current PART
        //file and construct the list of particle located inside

```

```

    //the sampling volume
    sampleParticle(np,nb,in_filename,samplingVolume);

    printTimeHistory(samplingVolume,time_history,i,
→         num_part,outFileName,propertyToAnalyze);

    //Progress feedback
    cout<<"Processing file "<<in_filenameTemp<<" for " <<
→         outFileName<<"\n";

    //Clearing the sampling volume
    samplingVolume.particle_sample.clear();

    //Delete pointer
    delete [] in_filename;
    in_filename=0;

    }//End of loop over PART files

    return 0;
} //End of main program
//*****

#endif

```


Kernel Approximation of Vertical Distribution

The program *kernel_approximation_vertical_distribution.exe* provides the time-history of a specified flow property at numerous points in the computational domain along a vertical axis. This flow property is approximated in the exact same matter as in the *kernel_approximation_single_point.exe* program presented above. In addition, this analysis is iterated and the location at which the approximation is performed varies incrementally based on provided parameters. The latter was implemented for sampling particles in regions of more realistic pressures by increasing the spacing separating the location of the approximation from a solid boundary as discussed in Section 5.3.2. For instance, this movement could also be used to determine the vertical velocity distributions along a cross-section more systematically. The ASCII input file containing the post-processing parameters for this program must be called *kernel_approx_vertical_dist_parameters.txt* and Figure C.4 shows an example.

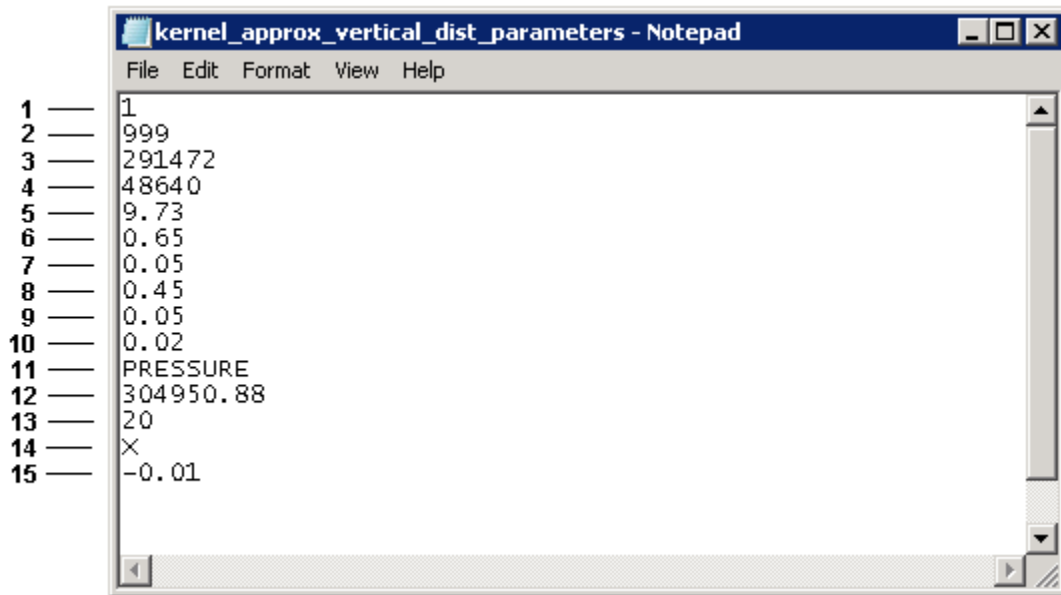


Figure C.4. Example of the content of an input file for the post-processing program *kernel_approximation_vertical_distribution.exe* considering run D3-DRY.

The first four lines (1-4) of the input file correspond to the same parameters as in the input files for the programs presented in the above sections. The fifth and sixth lines (5-6) correspond to the center coordinates (type double) in the x and y direction, respectively, of the vertical axis along which is align the several spherical sampling volumes. The seventh and eight lines (7-8) contain the lowest and highest elevations (type double), respectively, at which sampling volumes will be aligned on the vertical axis, while the ninth line (9) corresponds to the vertical distance (type double) center-to-center between these sampling volumes (meters). Note that if the difference between the maximum and lowest elevations specified in line eight and seventh, respectively, is not a multiple of the vertical distance separating the sampling volumes (ninth line), the elevation of the last sampling volume at the top of the vertical axis will correspond to the highest multiple of this vertical distance below the maximum elevation specified. The tenth line (10) contains the smoothing length h (type double) of the spherical sampling volume, while the eleventh line (11) corresponds to the flow property (type string) to be approximated at the center of the sampling volumes. The possible options for the latter are (exactly written) *PRESSURE*, *X_VELOCITY*, *Y_VELOCITY*, and *Z_VELOCITY*. If the approximation of the pressure is selected, the twelfth line (12) will correspond to $(c_0^2 \rho_0)/\gamma$ (see equation of state given by Eq. 2.66). Otherwise, the twelfth line corresponds to the number of iterations the analysis will be performed varying the location of the vertical axis. The two following lines (13-14) are the direction (either capital X or Y) and the increment (type double), respectively, according to which the location of the vertical axis will vary between iterations.

Every time an iteration is completed, an ASCII output file named for instance *Pressure distribution at Cx=9.73 Cy=0.65 with h=0.02.txt* will be created. The name of this file provides the center coordinates in the x-y plane of the vertical axis along which the sampling volumes are aligned, and the smoothing length h used for the approximations. Its content is numerous space-limited columns of data corresponding to the time-history of the approximated flow property for each of the sampling volumes over the specified range of PART files. The first column (most left) corresponds to the lowest sampling volume while the last column (most right) corresponds to the highest

sampling volume. Lastly, if no particle is present in a particular spherical sampling volume at a given instant, the corresponding result will be indefinite (*-1.#IND*).

The following is the source file the program utilizes in addition to the header files *particle.h* and *sampling_volume.h* presented earlier. Note that the main subroutine is located in the source file below.

kernel_approximation_vertical_distribution.cpp

```
/******
Code by Philippe St-Germain, University of Ottawa, 2012
The author makes no warranty or guarantee, expressed or
implied, as to the accuracy and reliability of the results
derived from the use of this code. Its free distribution
does not constitute any such warranty or guarantee. No
responsibility or liability is assumed by the author.
******/
#ifndef KERNEL_APPROX_DIST
#define KERNEL_APPROX_DIST

#include "particle.h"
#include "sampling_volume.h"

#include <math.h>
#include <string.h>
#include <vector>
#include <iostream>
#include <string>
#include <sstream>
#include <fstream>
#include <cstdlib>

namespace post_processing {

    using namespace std;

    void sampleParticle(int np, int nb, char* filename,
→         vector <SamplingVolume>
→         &sampling_volumes) {

        /******
        //NOTE:
        /******
        //To increase computational efficiency, particles located
        //within the rectangular limits of the sampling volumes
        //in the X-Y plane are searched first before being
        //assigned to the sampling volumes individually
        /******

        //Limits of the sampling volumes in the X-Y plane
        // in the inputs z_max < z_min there will be no sampling
        //volume in the array sampling_volumes and this will
        //generate an execution error
        double x_min=sampling_volumes[0].center_x-
→         2*sampling_volumes[0].h;
        double x_max=sampling_volumes[0].center_x+
→         2*sampling_volumes[0].h;
        double y_min=sampling_volumes[0].center_y-
→         2*sampling_volumes[0].h;
        double y_max=sampling_volumes[0].center_y+
→         2*sampling_volumes[0].h;
```

```

//Array containing the initial sample of particles
vector <Particle> sampled_particles;
//Variable for the current particle
Particle currentParticle;

//Single line of the PART file
string line;
char *charLine;
//Position of the current particle
double x_pos;
double y_pos;

//Opening the PART file for reading
ifstream infile;
infile.open(filename);

//Looping over all lines of the PART file
int nn=0;
while(nn<np){
    //Making sure that the end of the PART file
    //is not reached
    if(!infile.eof()){

        //Reading a line
        getline(infile,line);
        //Adding a dummy token at the end of the string
        //in order to avoid an error with the atof function
        //when an empty line is read
        line =line + " 1";
        //If not an empty line
        if(!(line==" 1")){
            //Only sampling water particles and not
            //boundary particles
            if(nn>nb){

                //Decomposing the line into an array of
                //characters
                charLine = new char[line.length()];
                for(unsigned int i=0;i<line.length();i++){
                    charLine[i] = line[i];
                }

                //Determination of the fluid particle position
                x_pos=atof(strtok(charLine," "));
                y_pos=atof(strtok(NULL," "));

                if(((x_min<=x_pos)&&(x_pos<=x_max))&&
→          ((y_min<=y_pos)&&(y_pos<=y_max))){

                    currentParticle.x_pos=x_pos;
                    currentParticle.y_pos=y_pos;
                    currentParticle.z_pos=atof(strtok(NULL," "));
                    currentParticle.x_vel=atof(strtok(NULL," "));
                    currentParticle.y_vel=atof(strtok(NULL," "));
                    currentParticle.z_vel=atof(strtok(NULL," "));
                    currentParticle.density=atof(strtok(NULL," "));
                    currentParticle.pressure=atof(strtok(NULL," "));
                }
            }
        }
    }
}

```

```

        currentParticle.mass=atof(strtok(NULL, " "));

        sampled_particles.push_back(currentParticle);
    }

    //Deleting pointer
    delete[] charLine;
    charLine=0;
    }//End of if water particle
    }//end of if not empty line
    }//end of if not end of file

    nn++;

} //end of loop over all particles

//Closing the PART file
infile.close();

//Distance between the current particle and
//the center of a particular sampling volume
double d;

//Looping over all sampling volumes
for(int i=0;i<sampling_volumes.size();i++){

    //Looping over all initially sampled particles
    for(int j=0;j<sampled_particles.size();j++){

        currentParticle = sampled_particles[j];

        //Computing the current particle distance
        d=sqrt(pow(((currentParticle.x_pos)-
→ sampling_volumes[i].center_x),2)+
→ pow(((currentParticle.y_pos)-
→ sampling_volumes[i].center_y),2)+
→ pow(((currentParticle.z_pos)-
→ sampling_volumes[i].center_z),2));

        //Checking if the current particle is inside the
        //current sampling
        if(d<=(2*sampling_volumes[i].h)){

            sampling_volumes[i].particle_sample.push_back
→ (currentParticle);

        }
    } //End of loop over all initially sampled particles
} //End of loop over all sampling volumes aligned
//vertically
} //End of sampleParticle function

void printDistribution(vector<SamplingVolume>
→ &sampling_volumes,
→ vector<vector<string>>
→ &time_history,int part_index,
→ int ini_part,int num_part,
```

```

→                                     string outFileName,
→                                     string propertyToAnalyze){

//Approximation of the flow property for each sampling
//volume
for(int i=0;i<sampling_volumes.size();i++){

    double measurement;
    string measurementString;
    ostringstream pp;

    if(propertyToAnalyze=="PRESSURE" ){
        measurement=sampling_volumes[i].avg_pressure();
    }
    if(propertyToAnalyze=="X_VELOCITY" ){
        measurement=sampling_volumes[i].avg_x_vel();
    }
    if(propertyToAnalyze=="Y_VELOCITY" ){
        measurement=sampling_volumes[i].avg_y_vel();
    }
    if(propertyToAnalyze=="Z_VELOCITY" ){
        measurement=sampling_volumes[i].avg_z_vel();
    }

    //Storing the approximated measurement in the
    //time-history array
    pp<<measurement;
    measurementString=pp.str();
    if(part_index==ini_part){
        vector <string> temp;
        temp.push_back(measurementString);
        time_history.push_back(temp);
    }
    else{
        time_history[i].push_back(measurementString);
    }

}

//End of loop over sampling volumes

//If all the PART files have been processed,
//the final output file is written
if(part_index==num_part){
    //Creating the output file containing the time-history
    //of the water surface elevation
    char* result_filename;
    result_filename=0;
    result_filename = new char[outFileName.length() + 1];
    strcpy(result_filename,outFileName.c_str());
    ofstream outfile(result_filename);

    //String variable for the construction of a single line
    //to be written to the output file
    string outputLine;

    //Writing the entire time-history of the measurement
    //for each of the sampling volume
    for(int i=0;i<time_history[0].size();i++){

```

```

        outputLine = "";
        for (int j =0;j<time_history.size();j++){
            outputLine+=(time_history[j])[i]+" ";
        }
        outputLine+="\n";
        outfile<<outputLine;
    }

    //Closing the output file
    outfile.close();

    //Deleting pointer
    delete [] result_filename;
    result_filename=0;
}
} // End of printDistribution function
}
//*****
//Main program
//*****
using namespace post_processing;

int main(int argc, char** argv) {

    //Variables for reading the raw PART files
    char* in_filename;
    string in_filenameTemp;
    string indexString;

    //Array for the time-history of the measurement
    vector<vector<string>> time_history;

    //Array of sampling volume objects
    vector <SamplingVolume> sampling_volumes;

    //*****
    //Input parameters
    //*****

    //Opening file containing input parameters
    ifstream infile;
    infile.open("kernel_approx_vertical_dist_parameters.txt");

    //Index of the intial PART file to begin post-processing
    int ini_part;
    //Number of PART files to be processed beyond initial
    //PART file
    int num_part;
    //Total number of particles in the simulation
    int np;
    //Number of boundary particles in the simulation
    int nb;
    //Initial coordinates of the center of sherical sampling
    //volumes
    double ini_center_x;
    double ini_center_y;

```



```

//Coordinates of the center of spherical sampling
//volumes to be incremented
double center_x;
double center_y;
//Center elevation of lowest sampling volume
double z_min;
//Center elevation of highest sampling volume
double z_max;
//Vertical distance between centers of adjacent sampling
//volumes
double z_dim;
//Smoothing length of sampling volume
double h;
//Variable for the equation of state (Eq. 2.66):
//B = (c0)^2 * rho0 / gamma
double B;
//Property to analyze for (PRESSURE, X_VELOCITY,
//Y_VELOCITY, or Z_VELOCITY)
string propertyToAnalyze;
//Number of iterations for the procedure
int numIterations;
//Direction in which the procedure is repeated
string direction; // "X" or "Y"
//Incremental distance between iterations
double increment;

//Reading in the inputs
infile>>ini_part;
infile>>num_part;
infile>>np;
infile>>nb;
infile>>ini_center_x;
infile>>ini_center_y;
infile>>z_min;
infile>>z_max;
infile>>z_dim;
infile>>h;
infile>>propertyToAnalyze;
if(propertyToAnalyze=="PRESSURE"){
    infile >> B;
}
infile>>numIterations;
infile>>direction;
infile>>increment;

//Closing the input file
infile.close();

//Writing the inputs to the command prompt window
cout<<ini_part<<" : Initial PART file to begin post-processing"<<"\n";
cout<<num_part<<" : Number of PART files to process"<<"\n";
cout<<np<<" : Total number of particles"<<"\n";
cout<<nb<<" : Number of boundary particles"<<"\n";
cout<<ini_center_x<<" : Initial center of sampling volume in the X
→ direction"<<"\n";
cout<<ini_center_y<<" : Initial Center of sampling volume in the Y
→ direction"<<"\n";

```

```

cout<<z_min<<" : Center elevation of lowest sampling volumes"<<"\n";
cout<<z_max<<" : Center elevation of highest sampling volumes"<<"\n";
cout<<z_dim<<" : Vertical distance between centers of sampling
→          volumes"<<"\n";
cout<<h<<" : Smoothing length"<<"\n";
cout<<propertyToAnalyze<<" : Flow property to approximate"<<"\n";
if(propertyToAnalyze=="PRESSURE"){
    cout<<B<<" :  $B = (c_0)^2 * \rho_0 / \gamma$ "<<"\n";
}
cout<<numIterations<<" : # of iterations for the procedure at
→          different locations along the specified
→          direction"<<"\n";
cout<<direction<<" : Direction in which the procedure is repeated (X
→          or Y)"<<"\n";
cout<<increment<<" : Incremental distance between iterations"<<"\n";
//*****

//Looping for each iteration of the procedure
for(int j=0;j<numIterations;j++){

    //Name of the output file containing the post-processing
    //results
    string outFileName;
    ostringstream cx;
    ostringstream cy;
    ostringstream h_temp;

    if(direction=="X"){
        center_x=ini_center_x+j*increment;
        center_y=ini_center_y;
    }
    else if(direction=="Y"){
        center_x=ini_center_x;
        center_y=ini_center_y+j*increment;
    }

    cx<<center_x;
    cy<<center_y;
    h_temp<<h;

    outFileName="Cx="+cx.str()+" Cy="+cy.str()+" with h="+
→          h_temp.str();

    if(propertyToAnalyze=="PRESSURE"){
        outFileName="Pressure distribution at "+
→          outFileName+".txt";
    }
    if(propertyToAnalyze=="X_VELOCITY"){
        outFileName="X velocity distribution at "+
→          outFileName+".txt";
    }
    if(propertyToAnalyze=="Y_VELOCITY"){
        outFileName="Y velocity distribution at "+
→          outFileName+".txt";
    }
    if(propertyToAnalyze=="Z_VELOCITY"){
        outFileName="Z velocity distribution at "+

```

```

→         outFileNames+".txt";
    }

    //Constructing the vertical stack of sampling volumes
    for(double z=z_min;z<z_max;z=z+z_dim){
        //Current sampling volume being constructed
        SamplingVolume samplingVolume;
        //Initializing its variables
        samplingVolume.center_x=center_x;
        samplingVolume.center_y=center_y;
        samplingVolume.center_z=z;
        samplingVolume.h=h;
        samplingVolume.B=B;
        //Inserting it in the array of sampling volume objects
        sampling_volumes.push_back(samplingVolume);
    }

    //Looping over PART files
    for (int i=ini_part;i<=num_part;i++){

        //*****
        //Generation of the string corresponding to the current
        //PART file to be processed
        //*****
        ostringstream ii;
        ii<<i;
        indexString=ii.str();

        if (i<10){
            in_filenameTemp="PART_000"+indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else if (i<100){
            in_filenameTemp = "PART_00"+indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else if (i<1000){
            in_filenameTemp="PART_0" +indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        else{
            in_filenameTemp="PART_" +indexString;
            in_filename=new char[in_filenameTemp.length()+1];
            strcpy(in_filename,in_filenameTemp.c_str());
        }
        //*****

        //Calling the function that will search the current PART
        //file and construct the list of particle located inside
        //the sampling volume
        sampleParticle(np,nb,in_filename,sampling_volumes);

        printDistribution(sampling_volumes,time_history,i,
→         ini_part,num_part,

```

```

→                                     outFileNames,propertyToAnalyze);

    //Progress feedback
    cout<<"->Processing file "<<in_filenameTemp<<" for
→      "<<outFileName<<"\n";

    //Clearing the sampling volume
    for(int k=0;k<sampling_volumes.size();k++){
        sampling_volumes[k].particle_sample.clear();
    }

    //Delete pointer
    delete [] in_filename;
    in_filename=0;
} //End of loop over PART files

//Emptying the time-history vector for the next iteration
time_history.clear();

//Clearing the sampling volumes for the next iteration
sampling_volumes.clear();

} //End of iterations

return 0;

} //End of main program
//*****

#endif

```

Pressure Integration

The program *pressure_integration.exe* provides the time-history of the resulting force acting in the x direction on either a circular or square column. This force is determined by the integration of the pressure acting on the surface of the column as detailed in Section 5.2.1. The implementation of this program resembles to some extent the one of the *kernel_approximation_vertical_distribution.exe* program as multiple vertical alignments of spherical sampling volumes are programmatically placed along the perimeter of the column. Due to the presence of a layer of unrealistic pressures as described in Section 5.3.2, the spacing between the sampling volumes and the surface of the column is incrementally increased through iterations within the execution of the program. Note that in order to obtain the resulting force in another direction than the x direction and if the cross-section of the column is not either circular or square, modifications to the code are required. The ASCII input file containing the post-processing parameters for this program must be called *pressure_integration_parameters.txt* and Figure C.5 shows an example.

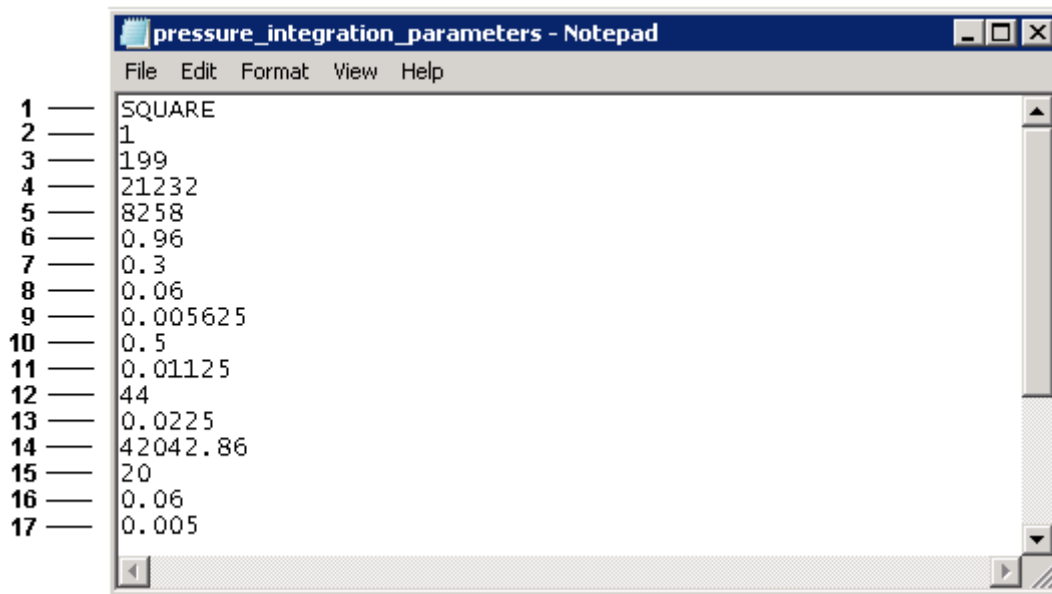


Figure C.5. Example of the content of an input file for the post-processing program *pressure_integration.exe* considering run D1.

The first line (1) of this input file specifies the cross-section (type string) of the column which is either (exactly written) *CIRCULAR* or *SQUARE*. The second line (2) corresponds to the index (type integer) of the PART file from which post-processing will begin while the third line (3) corresponds to the index (type integer) at which it will end. The fourth line (4) corresponds to the total number (type integer) of particles in the simulation (boundary and water particles) and the fifth line (5) is the number (type integer) of boundary particles alone. The sixth (6) and seventh (7) lines correspond to the center coordinates (type double) of the column in the x and y direction, respectively, and the eight line (8) represents the column's radius (circular column) or half of its side dimension (square column). The ninth (9) and tenth (10) lines contain the lowest and highest elevations (type double), respectively, at which sampling volumes will be aligned vertically around the perimeter of the column. The eleventh line (11) contains the vertical distance (type double) center-to-center between these sampling volumes. The twelfth line (12) corresponds to the number of vertical alignments of sampling volumes (type integer) that will be placed around the perimeter of the column or in other words, the number of sampling volumes around the column per elevation. Note that the values entered on the eleventh and twelfth lines, referred as *z_dim* and *numVol* in the source code, respectively, determine indirectly the size of the tributary areas of the sampling volumes based on the equations below.

$$[C.1] \quad \textit{Tributary Area} = z_dim \frac{2\pi}{numVol} r_c \quad \text{if circular column}$$

$$[C.2] \quad \textit{Tributary Area} = z_dim \frac{side\ dimension}{numVol/4} \quad \text{if square column}$$

where r_c is the radius of the column is circular. In the case of a square column *numVol* should be selected as a multiple of four. The thirteenth line (13) contains the smoothing length h (type double) of each spherical sampling volume and the fourteenth line (14) will correspond to $(c_0^2 \rho_0)/\gamma$ (see equation of state given by Eq. 2.66). The fifteenth line (15) corresponds to the number of iterations the analysis will be performed increasing the

spacing between the sampling volumes and the surface of the column. The last two lines (16-17) are the initial distance (type double) separating the center of the sampling volumes from the center of the column and its increment (type double), respectively.

Every time an iteration is completed, an ASCII output file named for instance *Force time-history for rv=0.06 h=0.0225 z_dim=0.01125 numVol=44.txt* will be created. The name of this file provides the center-to-center distance separating the sampling volumes from the column during the iteration, the smoothing length used, the center-to-center vertical distance separating the sampling volumes, and the number of sampling volumes around the column per elevation, respectively. Its content is a single column of data corresponding to the time-history of the resulting force acting in the x direction on the column.

The following is the source file the program utilizes in addition to the header files *particle.h* and *sampling_volume.h* presented earlier. Note that the main subroutine is located in the file source file below.

pressure_integration.cpp

```
/******
Code by Philippe St-Germain, University of Ottawa, 2012
The author makes no warranty or guarantee, expressed or
implied, as to the accuracy and reliability of the results
derived from the use of this code. Its free distribution
does not constitute any such warranty or guarantee. No
responsibility or liability is assumed by the author.
******/
#ifndef PRESSURE_INTEGRATION
#define PRESSURE_INTEGRATION

#ifndef PI
#define PI 3.141592654
#endif

#include "particle.h"
#include "sampling_volume.h"

#include <math.h>
#include <string.h>
#include <vector>
#include <iostream>
#include <string>
#include <sstream>
#include <fstream>
#include <cstdlib>

namespace post_processing {

    using namespace std;

    void sampleParticles(int np, int nb, char* filename,
→         vector<vector<SamplingVolume>>
→         &sampling_volumes, double cx, double cy){

        /******
        //NOTE:
        /******
        //To increase computational efficiency, particles located
        //within the rectangular limits of sampling volumes
        //in the X-Y plane are searched first before being
        //assigned to the sampling volumes individually
        /******

        //Limits of the sampling volumes in the X-Y plane
        // in the inputs z_max < z_min there will be no sampling
        //volume in the array sampling_volumes and this will
        //generate an execution error
        double h=(sampling_volumes[0])[0].h;
        double rv=(sampling_volumes[0])[0].rv;
        double x_min=cx-rv-2*h;
        double x_max=cx+rv+2*h;
        double y_min=cy-rv-2*h;
```



```

double y_max=cy+rv+2*h;

//Array containing the initial sample of particle
vector <Particle> sampled_particles;

//Current particle
Particle currentParticle;

//Single line of the PART file
string line;
char *charLine;
//Position of the current particle
double x_pos;
double y_pos;

//Opening the PART file for reading
ifstream infile;
infile.open(filename);

//Looping over all lines of the PART file
int nn=0;
while(nn<np){
    //Making sure that the end of the PART file
    //is not reached
    if(!infile.eof()){

        //Reading a line
        getline(infile,line);
        //Adding a dummy token at the end of the string
        //in order to avoid an error with the atof function
        //when an empty line is read
        line =line + " 1";
        //If not an empty line
        if(!(line==" 1")){
            //Only sampling water particles and not
            //boundary particles
            if(nn>nb){

                //Decomposing the line into an array of
                //characters
                charLine = new char[line.length()];
                for(unsigned int i=0;i<line.length();i++){
                    charLine[i] = line[i];
                }

                //Determination of the fluid particle position
                x_pos=atof(strtok (charLine," "));
                y_pos=atof(strtok (NULL," "));

                //Checking if the particle is within the
                //outer limit
                if (( (x_min<=x_pos)&&(x_pos<=x_max))&&
                    ((y_min<=y_pos)&&(y_pos<=y_max))) {
→
                    currentParticle.x_pos=x_pos;
                    currentParticle.y_pos=y_pos;
                    currentParticle.z_pos=atof(strtok(NULL," "));

```

```

        currentParticle.x_vel=atof(strtok(NULL, " "));
        currentParticle.y_vel=atof(strtok(NULL, " "));
        currentParticle.z_vel=atof(strtok(NULL, " "));
        currentParticle.density=atof(strtok(NULL, " "));
        currentParticle.pressure=atof(strtok(NULL, " "));
        currentParticle.mass=atof(strtok(NULL, " "));

        sampled_particles.push_back(currentParticle);

    }

    //Deleting pointer
    delete[] charLine;
    charLine=0;
} //End of if water particle
} //end of if not empty line
} //end of if not end of file

nn++;

} //end of loop over all particles

//Closing the PART file
infile.close();

//Distance between the current particle and
//the center of a particular sampling volume
double d;

//Looping over all the "vertical alignments" of
//sampling volumes
for(int i=0;i<sampling_volumes.size();i++){

    //Looping over each sampling volume of a
    //"vertical alignments"
    for(int j=0;j<sampling_volumes[i].size();j++){

        //Looping over all initially sampled particles
        for(int k=0;k<sampled_particles.size();k++){
            currentParticle = sampled_particles[k];
            //Computing the current sampled particle's distance
            d = sqrt(pow(((sampled_particles[k].x_pos)-
→          (sampling_volumes[i])[j].center_x),2)+
→          pow(((sampled_particles[k].y_pos)-
→          (sampling_volumes[i])[j].center_y),2)+
→          pow(((sampled_particles[k].z_pos)-
→          (sampling_volumes[i])[j].center_z),2));

            //Checking if the fluid particle is inside
            //the current sampling volume
            if (d <=(2*h)){

→          (sampling_volumes[i])[j].particle_sample.push_back
→          (currentParticle);

            }

        }

    } //End of looping over all initially sampled

```

```

        //particles
    } //End of looping over all sampling volumes of a
    // "vertical alignments"
} //End of looping over all "vertical alignments"
//of sampling volumes
} //End of RectangularSampleParticles function

void printForceInX(string type, vector<vector<SamplingVolume>>
→      &sampling_volumes, vector<double>
→      &force_time_history, int part_index,
→      int num_part, double col_dim,
→      int numVol, string outFileName){

    //Average pressure in the current sampling volume
    double pressure;
    //Tributary area of the current sampling volume
    double area;
    //Force contribution of the current sampling
    double forceInX;
    //Orientation angle of force contribution
    //(only used with circular column)
    double theta_force;
    //Width of tributary area of sampling volumes
    //(only used with square column
    int numVolPerFace=numVol/4;
    double volumeDim=2*col_dim/numVolPerFace;
    //Total resultant force in x acting on the column
    double totalForceInX=0.0;

    //Looping over all "vertical alignments" of sampling volumes
    for (int i=0; i< sampling_volumes.size(); i++){
        //Looping over all sampling volumes of "vertical alignment" i
        for(int j=0; j< sampling_volumes[i].size(); j++){

            //Getting the pressure
            pressure=(sampling_volumes[i])[j].avg_pressure();

            if(type=="CIRCULAR"){
                //Computing the orientation of the force
                //contribution
                theta_force=((sampling_volumes[i])[j].theta1+
→                (sampling_volumes[i])[j].theta2)/2-PI;

                //*****
                //Computing the tributary area
                //*****
                //For the radial sampling volume at 0 degrees

                if((sampling_volumes[i])[j].theta2<
→                (sampling_volumes[i])[j].theta1){
                    area=abs((sampling_volumes[i])[j].theta2*2)*
→                    col_dim*(sampling_volumes[i])[j].z_dim;
                }
                else{
→                area=abs((sampling_volumes[i])[j].theta2-
→                (sampling_volumes[i])[j].theta1)*col_dim*
→                (sampling_volumes[i])[j].z_dim;

```

```

    }
    /******

    //Computing the force contribution in the x
    //direction
    forceInX=pressure*area*cos(theta_force);

    //To check that the force contribution is not
    //-1.#IND before incrementing the total resultant
    //force
    if (forceInX==forceInX){
        totalForceInX+=forceInX;
    }
}
//SQUARE column
else{
    //Computing the tributary area
    area=volumeDim*(sampling_volumes[i])[j].z_dim;

    //Computing the force contribution in the
    //x direction
    forceInX=pressure*area;

    //To check that the force contribution is not
    //-1.#IND before incrementing the total resultant
    //force
    if(forceInX==forceInX){
        //Only considering sampling volumes at the front
        //or at the back of the column
        if(i<numVolPerFace){
            totalForceInX+=forceInX;
        }
        if((2*numVolPerFace<=i)&&(i<3*numVolPerFace)){
            totalForceInX-=forceInX;
        }
    }
}
}
} //End of Loop over all sampling volumes of "vertical alignment"
//i
} //End of Loop over all "vertical alignments" of sampling volumes

//Adding the total resultant force to the force
//time-history array
force_time_history.push_back(totalForceInX);

//If all the PART files have been processed,
//the final output file is written
if(part_index==num_part){
    //Creating the output file containg the time-history
    //of the water surface elevation
    char* result_filename;
    result_filename=0;
    result_filename = new char[outFileName.length() + 1];
    strcpy(result_filename,outFileName.c_str());
    ofstream outfile(result_filename);

    for(int i=0;i<force_time_history.size();i++){

```

```

        outfile<<force_time_history[i]<<"\n";
    }

    //Closing the output file
    outfile.close();

    //Deleting pointer
    delete [] result_filename;
    result_filename=0;
}
} // End of printForceInX method
}

//*****
//Main program
//*****
using namespace post_processing;

int main(int argc, char** argv){

    //Variables for reading the raw PART files
    char* in_filename;
    string in_filenameTemp;
    string indexString;

    //Array used to store the resultant force time-history
    vector <double> force_time_history;

    //*****
    //Input parameters
    //*****

    //Opening file containing input parameters
    ifstream infile;
    infile.open("pressure_integration_parameters.txt");

    //Type of column considered (CIRCULAR or SQUARE)
    string type;
    //Index of the intial PART file to begin post-processing
    int ini_part;
    //Number of PART files to be processed beyond initial
    //PART file
    int num_part;
    //Total number of particles in the simulation
    int np;
    //Number of boundary particles in the simulation
    int nb;
    //Coordinates of the column's center in the X-Y plane
    double cx;
    double cy;
    //Half dimension of column:
    //radius for circular column
    //side dimension/2 for square column
    double col_dim;
    //Cente elevation of lowest sampling volume
    double z_min;

```

```

//Center elevation of highest sampling volume
double z_max;
//Vertical distance between centers of adjacent sampling
//volumes
double z_dim;
//Initial distance between center of sampling volumes
//with center of column
double ini_rv;
//Number of sampling volumes per elevation
int numVol;
//Smoothing length of sampling volume
double h;
//Variable for the equation of state (Eq. 2.66):
//B = (c0)^2 * rho0 / gamma
double B;
//Number of iterations for the procedure
int numIterations;
//Incremental distance between iterations
double increment;

//Reading in the inputs
infile >> type;
infile>>ini_part;
infile>>num_part;
infile>>np;
infile>>nb;
infile>>cx;
infile>>cy;
infile>>col_dim;
infile>>z_min;
infile>>z_max;
infile>>z_dim;
infile>>numVol;
infile>>h;
infile >> B;
infile>>numIterations;
infile>>ini_rv;
infile>>increment;

//Closing the input file
infile.close();

//Writing the inputs to the command prompt window
cout<<type<<" : Column type (CIRCULAR or SQUARE)"<<"\n";
cout<<ini_part<<" : Intial PART file to begin post-processing"<<"\n";
cout<<num_part<<" : Number of PART files to process"<<"\n";
cout<<np<<" : Total number of particles"<<"\n";
cout<<nb<<" : Number of boundary particles"<<"\n";
cout<<cx<<" : X coordinate of the column's center"<<"\n";
cout<<cy<<" : Y coordinate of the column's center"<<"\n";
cout<<col_dim<<" : Half dimension of column"<<"\n";
cout<<z_min<<" : Center elevation of lowest sampling volumes"<<"\n";
cout<<z_max<<" : Center elevation of highest sampling volumes"<<"\n";
cout<<z_dim<<" : Vertical distance between centers of sampling
→ volumes"<<"\n";
cout<<numVol<<" : # of sampling volumes around the column per
→ elevation (multiple of 4)"<<"\n";

```

```

cout<<h<<" : Smoothing length"<<"\n";
cout<<B<<" : B = (c0)^2 * rho0 / gamma" <<"\n";
cout<<numIterations<<" : Number of iterations for the procedure with
→          different spacings"<<"\n";
cout<<ini_rv<<" : Initial dist. between center of sampling volumes
→          with center of column"<<"\n";
cout<<increment<<" : Incremental distance (spacing) between
→          iterations"<<"\n";
//*****

//Looping for each iteration of the procedure
for(int k = 0; k < numIterations; k++){

    //Name of the output file containing the post-processing
    //results
    string outFileName;
    ostringstream ini_rv_temp ;
    ostringstream h_temp ;
    ostringstream z_dim_temp ;
    ostringstream numVol_temp ;
    ini_rv_temp << ini_rv + k * increment;
    h_temp << h;
    z_dim_temp << z_dim;
    numVol_temp << numVol;

    outFileName="Force time-history for rv="+
→          ini_rv_temp.str()+" h= "+h_temp.str()+
→          " z_dim= "+z_dim_temp.str()+" numVol= "+
→          numVol_temp.str()+".txt";

    //Array containing "columns" of sampling volumes
    //Each of these "columns" are themselves
    //arrays of sampling volumes
    vector<vector<SamplingVolume>> sampling_volumes;

    //Layout of sampling volumes around a
    //circular column
    if(type=="CIRCULAR"){
        //Arc angle of the tributary area associated to
        //each sampling volume
        double delta_theta;
        delta_theta=6.283185307/numVol;

        //Constructing numVol times "vertical alignments"
        //of sampling volumes
        for(int j=0;j<numVol;j++){
            for(double z=z_min;z<=z_max;z=z+z_dim){
                //Current sampling volume
                SamplingVolume samplingVolume;

                //Initializing the start and end angles
                if(j==0){
                    double temp=j*delta_theta-delta_theta/2;
                    samplingVolume.theta2=temp+delta_theta ;
                    samplingVolume.theta1=6.283185307-delta_theta/2;
                }
                else{

```

```

→      samplingVolume.theta1=j*delta_theta-
→          delta_theta/2;
→      samplingVolume.theta2=samplingVolume.theta1+
→          delta_theta;
→  }

  //Initializing tributary heigth of sampling volume
  samplingVolume.z_dim=z_dim;

  //Initializing center radius of sampling volume
  samplingVolume.rv=ini_rv+k*increment;

  //Volume's center coordinates
  if (j==0){
→      samplingVolume.center_x=cx+samplingVolume.rv*
→          cos(0.0);
→      samplingVolume.center_y=cy+samplingVolume.rv*
→          sin(0.0);
→      samplingVolume.center_z=z;
  }
  else{
→      samplingVolume.center_x=cx+samplingVolume.rv*
→          cos((samplingVolume.theta1+
→              samplingVolume.theta2)/2);
→      samplingVolume.center_y=cy+samplingVolume.rv*
→          sin((samplingVolume.theta1+
→              samplingVolume.theta2)/2);
→      samplingVolume.center_z=z;
  }

  //Initializing smoothing length of sampling volume
  samplingVolume.h=h;

  //Initializing Coefficient for EoS of sampling volume
  samplingVolume.B=B;

  //If very first sampling volume at the bottom
  if (z==z_min){
      vector <SamplingVolume> temp;
      temp.push_back(samplingVolume);
      sampling_volumes.push_back(temp);
  }
  //Else insert the current sampling volume in
  //the sampling volume array directly
  else{
      sampling_volumes[j].push_back(samplingVolume);
  }
  }
}

//Layout of sampling volumes around a
//square column
else{

  //Number of sampling volume per face
  int numVolPerFace=numVol/4;

```



```

//x or y dimension of tributary area of the
//sampling volumes
double volumeDim=2*col_dim/numVolPerFace;

//Front face of square column
for(int j=0;j<numVolPerFace;j++){
    for(double z=z_min;z<=z_max;z=z+z_dim){
        //Current sampling volume
        SamplingVolume samplingVolume;

        //Height of tributary area
        samplingVolume.z_dim=z_dim;

        //Volume's center coordinates
        samplingVolume.center_x=cx-ini_rv-k*increment;
        samplingVolume.center_y=(cy+col_dim)-
            ((j+1)*volumeDim)+(volumeDim/2);
        samplingVolume.center_z=z;

        //Distance from center of column
        samplingVolume.rv=ini_rv+k*increment;

        //Initializing smoothing length sampling volume
        samplingVolume.h=h;

        //Initializing Coefficient for EoS of sampling volume
        samplingVolume.B=B;

        //If very first sampling volume at the bottom
        if(z==z_min){
            vector<SamplingVolume> temp;
            temp.push_back(samplingVolume);
            sampling_volumes.push_back(temp);
        }
        //Else insert the current sampling volume in
        //the sampling volume array directly
        else{
            sampling_volumes[j].push_back(samplingVolume);
        }
    }
}

//Skier's right face of square column
for(int j=0;j<numVolPerFace;j++){
    for(double z=z_min;z<=z_max;z=z+z_dim){
        //Current sampling volume
        SamplingVolume samplingVolume;

        //Height of tributary area
        samplingVolume.z_dim=z_dim;

        //Volume's center coordinates
        samplingVolume.center_x=(cx-col_dim)+
            ((j+1)*volumeDim)-(volumeDim/2);
        samplingVolume.center_y=cy-ini_rv-k*increment;
        samplingVolume.center_z=z;
    }
}

```

```

//Distance from center of column
samplingVolume.rv=ini_rv+k*increment;

//Initializing smoothing length of sampling volume
samplingVolume.h = h;

//initializing Coefficient for EoS of sampling volume
samplingVolume.B = B;

//If very first sampling volume at the bottom
if (z==z_min){
    vector<SamplingVolume> temp;
    temp.push_back(samplingVolume);
    sampling_volumes.push_back(temp);
}
//Else insert the current sampling volume in
//the sampling volume array directly
else{
    sampling_volumes[j+numVolPerFace].push_back
        (samplingVolume);
}
}
}

//Back face of square column
for(int j=0;j<numVolPerFace;j++){
    for (double z=z_min; z<=z_max; z=z+z_dim){
        //Current sampling volume
        SamplingVolume samplingVolume;

        //Height of tributary area
        samplingVolume.z_dim=z_dim;

        //Volume's center coordinates
        samplingVolume.center_x=cx+ini_rv+k*increment;
        samplingVolume.center_y =(cy-col_dim)+((j+1)*volumeDim)-
            (volumeDim/2);
        samplingVolume.center_z=z;

        //Distance from center of column
        samplingVolume.rv=ini_rv+k*increment;

        //Initializing smoothing length of sampling volume
        samplingVolume.h=h;

        //Initializing Coefficient for EoS of sampling volume
        samplingVolume.B=B;

        //If very first sampling volume at the bottom
        if(z==z_min){
            vector<SamplingVolume> temp;
            temp.push_back(samplingVolume);
            sampling_volumes.push_back(temp);
        }
        //Else insert the current sampling volume in
        //the sampling volume array directly
        else{

```

```

→         sampling_volumes[j+2*numVolPerFace].push_back
                                                    (samplingVolume);

    }
}

//Skier's left face of square column
for(int j=0;j<numVolPerFace;j++){
    for(double z=z_min;z<=z_max;z=z+z_dim){
        //Current sampling volume
        SamplingVolume samplingVolume;

        //Height of tributary area
        samplingVolume.z_dim=z_dim;

        //Volume's center coordinates
        samplingVolume.center_x=(cx+col_dim)-
→         ((j+1)*volumeDim)+(volumeDim/2);
        samplingVolume.center_y=cy+ini_rv+k*increment;
        samplingVolume.center_z=z;

        //Distance from center of column
        samplingVolume.rv=ini_rv+k*increment;

        //Initializing smoothing length of sampling volume
        samplingVolume.h=h;

        //Initializing Coefficient for EoS of radial sampling volume
        samplingVolume.B=B;

        //If very first sampling volume at the bottom
        if(z==z_min){
            vector <SamplingVolume> temp;
            temp.push_back(samplingVolume);
            sampling_volumes.push_back(temp);
        }
        //Else insert the current sampling volume in
        //the sampling volume array directly
        else{
            sampling_volumes[j+3*numVolPerFace].push_back
→         (samplingVolume);
        }
    }
}

//Main loop over the PART files
for(int i=ini_part;i<=num_part;i++){

    //*****
    //Generation of the string corresponding to the current
    //PART file to be processed
    //*****
    ostringstream ii;

```

```

ii<<i;
indexString=ii.str();

if (i<10){
    in_filenameTemp="PART_000"+indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else if (i<100){
    in_filenameTemp = "PART_00"+indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else if (i<1000){
    in_filenameTemp="PART_0" +indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
else{
    in_filenameTemp="PART_" +indexString;
    in_filename=new char[in_filenameTemp.length()+1];
    strcpy(in_filename,in_filenameTemp.c_str());
}
//*****

sampleParticles(np,nb,in_filename,
→          sampling_volumes,cx,cy);

printForceInX(type,sampling_volumes,force_time_history,
→          i,num_part,col_dim,numVol,outFileName);

//Progress feedback
cout<<"->Processing file "<<in_filenameTemp<<" for
→          "<<outFileName<<"\n";

//Emptying the sampled particles
//Loopin over all the "columns" of sampling volumes
for(int m=0 ;m<sampling_volumes.size();m++){
    //Looping over each radial sampling volume
    for (int n=0;n<sampling_volumes[m].size();n++){
        (sampling_volumes[m])[n].particle_sample.clear();
    }
}

//Delete pointer
delete [] in_filename;
in_filename=0;

} //End of loop over PART files

//Emptying the force time-history array
force_time_history.clear();

//Emptying the sampling volume arrays
for(int i=0;i<sampling_volumes.size();i++){
    (sampling_volumes[i]).clear();
}

```

```
        sampling_volumes.clear();

    } //End of Iterations

    return 0;

} //End of main program
//*****
#endif
```

Appendix D Additional Comparison of Numerical with Physical Results

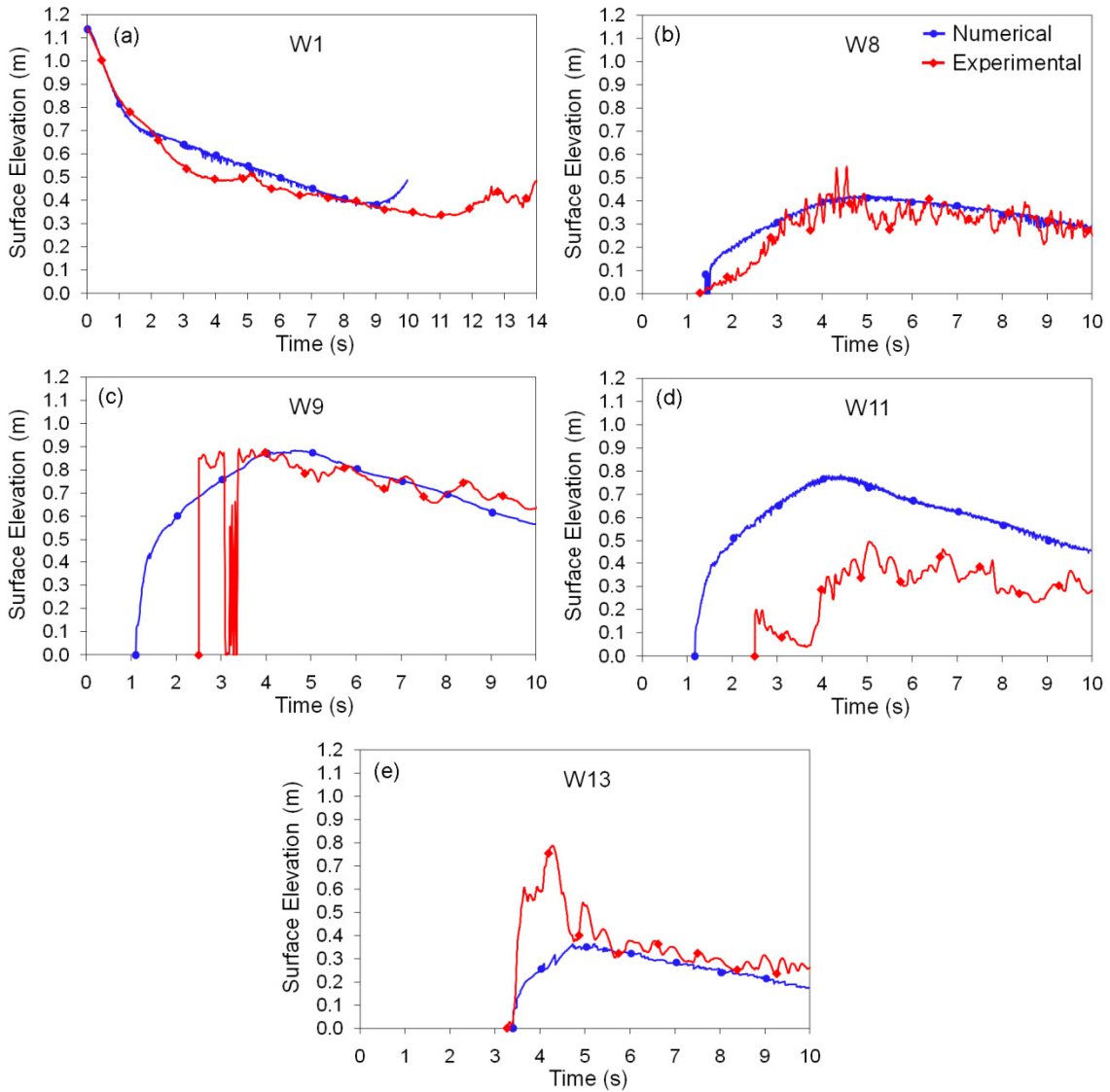


Figure D.1. Second comparison of numerical and experimental time-history of the water surface elevation at wave gages (a) W1, (b) W8, (c) W9, (d) W11, and (e) W13 for $h_u = 1.15\text{m}$ (runs E-115 and D2-115).

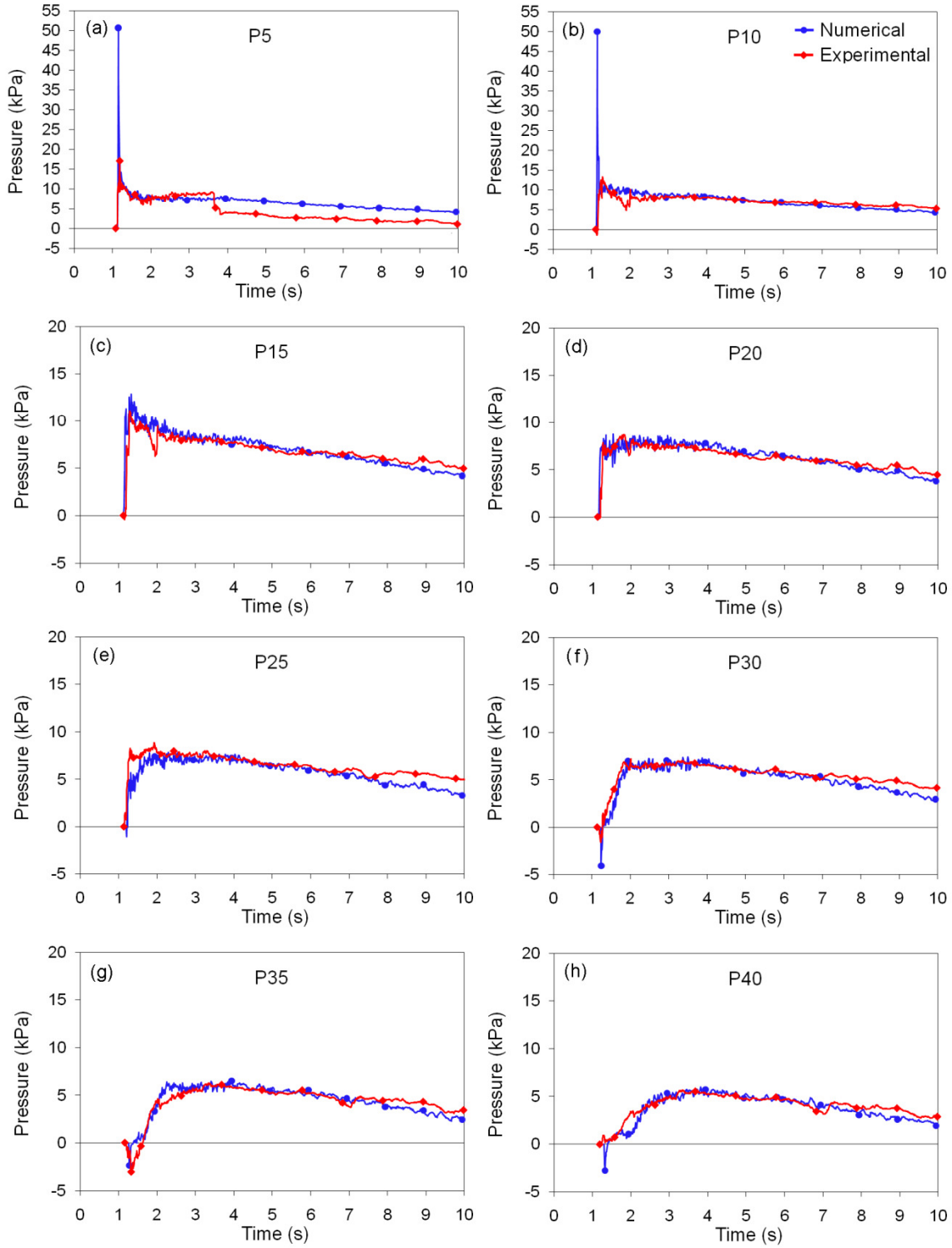


Figure D.2. Second comparison of numerical and experimental time-history of pressures acting on the column at elevations of (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, and (h) 0.40m above the flume's bottom for $h_u = 1.15\text{m}$ (runs E-115 and D2-115).

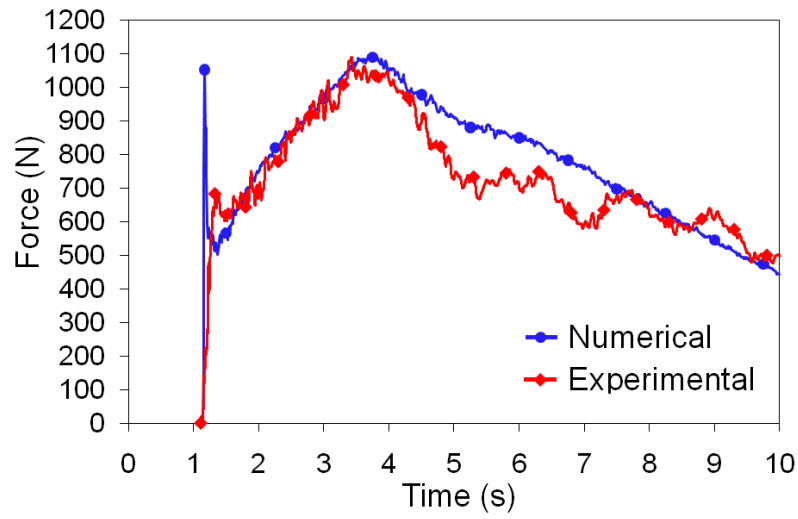


Figure D.3. Second comparison of numerical and experimental time-history of the net base horizontal force acting on the column for $h_u = 1.15\text{m}$ (runs E-115 and D2-115).