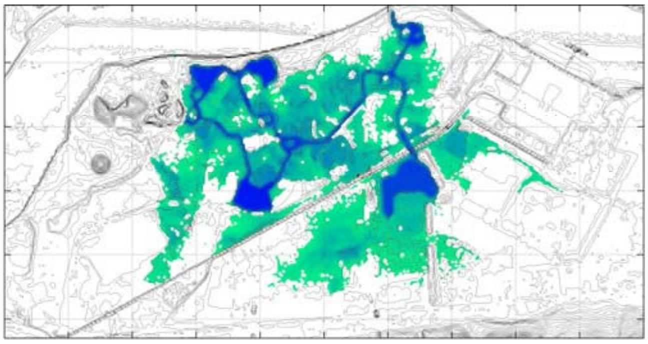
Università degli Studi di Parma
Facoltà di Ingegneria

Dottorato di Ricerca in Ingegneria Civile (XXII Ciclo)
Curriculum: protezione idraulica del territorio, ICAR\02

Renato Vacondio

Shallow Water and Navier-Stokes SPH-like numerical modelling of rapidly varying free-surface flows



Dissertazione per il conseguimento del titolo di Dottore di Ricerca

Tutore: Prof. Paolo Mignosa
Co-tutore: Dr. Benedict D. Rogers
Coordinatore del Dottorato: Prof. Paolo Mignosa

Parma, Gennaio 2010

a Fede

Acknowledgment

Il primo ringraziamento va al prof. Mignosa per la fiducia che mi ha dimostrato fin dal primo momento; la chiarezza e la curiosità che lo contraddistinguono hanno rappresentato per me un esempio prezioso. Con lui ringrazio la prof.ssa Tanda e il prof. Longo per i consigli e la gentilezza, Andrea M., Andrea Z. e Francesca per i suggerimenti e l'aiuto che hanno saputo darmi. Con Marco Luca e Chiara ho condiviso buona parte delle difficoltà di questo percorso; le loro risate e gli incoraggiamenti che mi hanno dato sono stati davvero indispensabili.
"Tutti noi siamo quello che ci hanno insegnato ad essere" perciò grazie ai miei genitori, hai quali devo tutto.

A special thank to Dr. Benedict D. Rogers, whom I shared a good part of this work with, for his generously support and valuable guidance. He introduced the secrets of SPH to me and gave me generous technical helps. I would like to express my sincere gratitude also to Prof Peter K. Stansby for giving me the opportunity of visiting the University of Manchester and for his useful scientific hints during our brainstorming meetings in Bens office.
A big thank you to Rui for all the 5-a-side matches, thanks to Maurice for the teas and beers. Last but not least, I would like to express my appreciation to Pourya since he made me feel at home from the very first time in Manchester.

Abstract

In coastal engineering, Lagrangian meshless numerical methods have reached a good popularity and they have been applied with success to describe wave breaking, impact of wave on structures and other rapid phenomena. This is due to the fact that they have a number of advantages in comparison with classical Eulerian schemes: no explicit treatment of the free surface and no computational grid mean that sophisticated meshing is not needed for complex geometries and therefore a number of problems that were considered largely intractable using classical Eulerian numerical methods such as finite volume or finite elements can now be simulated. As a relatively new method in Computational Fluid Dynamics, this kind of methods may be considered immature and many fundamental aspects and key characteristics remain to be fully investigated. The solid boundary condition is such an example: imposing closed boundary conditions in meshless methods in general, and in Smoothed Particle Hydrodynamic (SPH) in particular, is still an open problem. In the first chapter of this thesis an approximate Virtual Boundary Particle Method (VBP) for solid boundary conditions in two-dimensional (2-D) SPH models is presented; this is a development of the original VBP method recently proposed by Ferrari et al. (A new 3-D parallel SPH scheme for free-surface flows, Computers & Fluids, 38(6), 1203-1217, 2009). The aim is to maintain the zeroth moment of the kernel function as closely as possible to unity, (a property referred to as zero-consistency), for particles close to solid boundaries. The main advantage of the MVBP in comparison with other methods such as Mirrored Particles is that curved boundaries or boundaries with angles can be easily reproduced.

Some authors applied the Smoothed Particle Hydrodynamics (SPH) method to integrate the Shallow Water Equations (SWE) obtaining promising results for simple test cases where no open boundaries are present and the analytical formulation of source terms are applied: with SPH the wet-dry fronts do not need any special treatment, the equations are solved just where the fluid is present and

this can potentially speed up the calculations if there are large dry areas in the domain. A 2D Shallow Water code based on the SPH interpolation is developed in the chapters 2 - 4 of this work, with the aim of further improving the capability of these numerical schemes of simulating real flooding events. The SPH-SWEs code is developed following the variational formulation, thanks to this approach the numerical scheme is robust and both the total mass and the momentum are conserved.

Some major improvements has been introduced in the SPH-SWEs model in order to make the simulation of real floodings feasible. The Modified Virtual Boundary Particles (MVBP) is used to describe the closed boundaries, the bottom and the friction source term are described by a set of bottom particles. This discretization is effective not just for simple test case but also in for real bathymetries. Moreover, a particle splitting procedure has been inserted: it has the purpose to avoid the lack of resolution due to the variable kernel size being inversely proportional to water depth. This splitting procedure conserves mass and momentum by varying the smoothing length, velocity and acceleration of each refined particle. This improves predictions but does not necessarily provide good shock capturing. This is improved by treating particle interactions as a Riemann problem with MUSCL reconstruction providing stability.

The last limitation that inhibits the use of the SPH-SWEs for real flooding simulation is the absence of any method to impose open boundary conditions. These are introduced in chapter 4 by adopting a simplified version of the Characteristic boundary method. Both supercritical and subcritical inflow and outflow boundary conditions can be simulated.

Thanks to all the improvements described above, the simulation of two real events by a SPH-SWEs is presented in chapter 4, for the first time. The first case is the Okushiri tsunami occurred in Japan in 1993, whereas the second one is a flooding flood inundation at Thamesmead (UK).

In Chapter 5 the simulation of rapidly varying flows is analysed removing the hypothesis of Shallow Water flows: a meshless Lagrangian numerical model called Finite Pointset Method (FPM) for the integration of Navier-Stokes equations in presence of free-surface flow is presented. The Finite Pointset Method (FPM) is a Lagrangian meshless method for numerical integration of pure incompressible Navier-Stokes equations, applied to date just to internal flows. It belongs to SPH like family because each particle carries a vector of field quantities such as pressure, density, velocity etc. and information and physical quantities are approximated using particles in a circular neighbourhood. FPM holds also some remarkable advantages in comparison with classical SPH methods: it is based on a moving least squares approach, where particles are just interpolation points without any associated mass and this means that any order of accuracy can be reached regardless to the particles position. In FPM the fluid is described as purely incompressible and the Navier-Stokes equation are solved numerically by means of the projection method therefore no spurious oscillations in the pressure field are present. Moreover in FPM boundary conditions can be analytically enforced using boundary particles and fluid particles can be added and removed in order to preserve the stability of the solution. This fact represents another fundamental advantage in comparison with classical SPH. Originally the FPM has been confined to single or two phase flow, but in chapter 5 it has extended also to free-surface flows by introducing a novel algorithm for free surface detection. In addition to that, a novel formulation of the Projection Method, called Incremental Pressure Projection Method, has been applied in order to preserve the hydrostatic condition.

Contents

1 Zeroth-order Consistent Boundary Conditions 3

1.1 Literature review of closed boundary methods 3
1.2 Zeroth-order consistency in SPH formalism 4
1.3 Virtual Boundary Particle method 6
1.4 Zeroth-order consistent evaluation for a range of possible wall configurations 8
1.5 Correction for variable smoothing length 11
1.6 Still shallow water test cases 16
  1.6.1 1-D 17
  1.6.2 2-D 27
1.7 Concluding Remarks 37

2 SPH numerical model for Shallow Water equations 41

2.1 Introduction 41
2.2 SPH for Shallow water: Variational formulation 42
  2.2.1 Density evaluation 43
  2.2.2 Momentum equation 45
  2.2.3 Internal force calculations 47
2.3 Bed gradient source term 49
2.4 Friction source term 51
2.5 Time integration scheme 52
2.6 Postprocessing output 52
2.7 Stabilization term 53
2.8 Wetting - drying test case: Thacker basin 58
2.9 Concluding Remarks 63

3 Particle Splitting 67

3.1 The need for particle splitting 67
3.2 Splitting procedure 68
3.3 SWE-SPH code with splitting procedure 73
3.3.1 Theoretical test case 73
3.4 Test Cases 75
3.4.1 Circular Dam Break over dry bed 75
3.4.2 1D-2D dam break over dry bed 79
3.4.3 CADAM test case with a 45° bend 79
3.5 Concluding Remarks 91

4 Development of open boundaries for SPH-SWEs and Application to real problems 93

4.1 Characteristic boundary method 93
4.2 Open Boundaries in SPH 95
4.2.1 SPH interpolation at Open Boundary particles 96
4.3 1-D Steady flow over a bump 98
4.4 2-D Uniform flow in a sloping rectangular channel 99
4.5 1993 Okushiri tsunami 104
4.6 Flood inundation in Thamesmead 106
4.7 Concluding Remarks 114

5 Finite Pointset Method for free-surface Flow 121

5.1 Finite pointset method and SPH 121
5.2 Incremental Pressure Projection Method 122
5.3 FPM implementation 124
5.3.1 Moving Least Squares approximation 124
5.3.2 Elliptic equations 125
5.3.3 Particle management 127
5.3.4 Variable smoothing length 128
5.3.5 Free-surface boundary condition 129
5.3.6 Time step 129
5.4 Test Cases 130
5.4.1 Stationary fluid in a box 130
5.4.2 Taylor-Green flow 131
5.4.3 Evolution of intially circular water bubble 133
5.4.4 Initial stages of a dam break flow 136
5.5 Concluding Remarks 137

6 Conclusions **139**

List of Figures

1.1 stencil of a internal particle 7

1.2 particle close to a boundary straight line: stencil generated using (a) the original VBP and (b) the MVBP 8

1.3 particle close to 90° internal angle: stencil generated using (a) the original VBP and (b) the MVBP 8

1.4 particle close to a curved boundary: stencil generated using (a) the original VBP and (b) the MVBP 9

1.5 internal particle close to 270° internal angle: stencil generated using the original VBP and the MVBP (there is no difference) . . 9

1.6 m_0 calculated for a set of disordered particles in a square bounded domain: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP 11

1.7 m_0 calculated in a bounded domain with an 270° internal angle with an L-shape geometry: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP 12

1.8 m_0 calculated in a bounded domain with 270° internal angle: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP 13

1.9 m_0 calculated in a bounded domain with a toroid shape: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP 14

1.10 comparison of m_0 calculated in a bounded domain with a toroid shape in the section y=0 using no boundary condition, mirrored particle method, VBP and MVBP 14

1.11 stencil generated for a particle close to a boundary: (a)-straight wall and (b)-internal angle of 90° 15

1.12 zero-th moment m_0 calculated with and without the variable smoothing length correction of Equation (1.9) 16
1.13 still water over a hump: (a) water depth, (b) velocity 18
1.14 still water over a hump, L_2 norm of error: (a) water depth, (b) velocity 19
1.15 still water over a hump and wetting and drying interface: (a) water depth, (b) velocity 20
1.16 still water over a hump and wetting and drying interface, L_2 norm of error: (a) water depth, (b) velocity 21
1.17 still water over a step: (a) water depth, (b) velocity 23
1.18 still water over a step, L_2 norm of error: (a) water depth, (b) velocity 24
1.19 still water over a parabolic topography with wetting and drying interface: (a) water depth, (b) velocity 25
1.20 still water over a parabolic topography with wetting and drying interface, L_2 norm of error: (a) water depth, (b) velocity 26
1.21 still water over a parabolic topography: (a) water depth, (b) velocity 27
1.22 still water over a parabolic topography, L_2 norm of error: (a) water depth, (b) velocity 28
1.23 still water over a parabolic topography 2-D at time 1000 s: velocity field and particle positions 29
1.24 still water over a parabolic topography 2-D with wetting and drying interface at time 1000 s: (a) free surface elevation, (b) velocity magnitude 30
1.25 still water over a parabolic topography 2-D with wetting and drying interface:, section y=0: (a) free surface elevation, (b) velocity magnitude 31
1.26 still water over a parabolic topography 2-D with MVBP boundary condition at time 1000 s: velocity field and particle positions . . . 31
1.27 still water over a parabolic topography 2-D with MVBP method at time 1000 s: (a) free surface elevation, (b) velocity magnitude . 32
1.28 still water over a parabolic topography 2-D with mirrored particle boundary condition at time 1000 s: velocity field and particle positions 32

1.29 still water over a parabolic topography 2-D with mirrored particle boundary condition at time 1000 s: (a) free surface elevation, (b) velocity magnitude 33
1.30 still water in a box with an internal angle of 300° with MVBP method at time 1 s: velocity field and particle positions, (a) whole domain (b) zoom 34
1.31 still water in a box with an internal angle of 300° with MVBP method at time 1 s: (a) free surface elevation, (b) velocity magnitude 35
1.32 still water in a box with an internal angle of 300° with mirrored particle boundary condition: velocity field and particles position, (a) whole domain (b) zoom 35
1.33 still water in a box with an internal angle of 300° with mirrored particle boundary condition: velocity field and particles position: (a) free surface elevation, (b) velocity magnitude 36
1.34 still water in a box with an internal angle of 300° with VBP: velocity field and particles position after 0.1 s of simulation, (a) whole domain (b) zoom 37
1.35 still water in a box with an internal angle of 345° with MVBP method at time 1 s: velocity field and particle positions 37
1.36 still water in a box with an internal angle of 345° with MVBP method at time 1 s: (a) free surface elevation, (b) velocity magnitude 38

2.1 flow with a free surface under the effect of gravity 42
2.2 Water depth for 1-D Dam Break with wet bed at time 50 s: (a) no viscosity, (b) artificial viscosity (c) Lax-Friedrichs flux (d) Lax-Friedrichs flux with MUSCL reconstruction (e) two-shock Riemann solver with MUSCL reconstruction 57
2.3 Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Lax-Friedrichs flux 59
2.4 Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Lax-Friedrichs flux and MUSCL reconstruction 60
2.5 Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Riemann solver and MUSCL reconstruction 61
2.6 Definition sketch for the Thacker test with planar water surface . . 61

2.7 Thacker test case, water depth at non dimensional time steps t/T: (a) 0.015, (b) 0.253, (c) 0.505, (d) 0.742 and (e) 0.980 63

2.8 Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 5000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity 64

2.9 Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 6000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity 64

2.10 Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 7000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity 65

3.1 velocity field in the 1-D dam break in a rectangular channel over dry bed without splitting at time 20s 68

3.2 1-D Dam Break in a rectangular channel without splitting: (a) water depth and (b) v_x at the section y=500 m at time 20s 69

3.3 splitting patterns 69

3.4 2-D cubic kernel approximation using (a- unrefined and (b)-refined configuration 72

3.5 particle position for the numerical test of density calculation in presence of refined particles 75

3.6 Density calculation with refined particle in the central part of the square domain (a) using scatter (Equation (3.3)) and (b) gather (Equation (2.4)) formulation 76

3.7 velocity field in the Circular Dam Break test case over dry bed with splitting at time 30s. (a) the whole domain (b) zoom close to the splitting interface 77

3.8 Circular dam break with dry bed: water depth in radial direction at time (a)-10 s, (b)-30 s and (c)-50 s 78

3.9 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term with MUSCL reconstruction: velocity field in the 2-D dam break in a rectangular channel over dry bed at time 30s 79

3.10 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term with MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s . . . 80

3.11 1-D Dam Break in a rectangular channel with splitting; Lax-Friedrichs stabilization term with MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s 81
3.12 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term without MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s 82
3.13 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term without MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s 83
3.14 1-D Dam Break in a rectangular channel with splitting, Two-Schock Riemann solver with no MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s 84
3.15 1-D Dam Break in a rectangular channel with splitting, Two-Schock Riemann solver with MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s 85
3.16 CADAM test case: plane and profile view of the experimental setup 85
3.17 CADAM test case with no splitting: water level registered at different gauges 86
3.18 CADAM test case with no splitting: comparison between FV and SPH-SWEs models at gauges (a) 4 (b) 6 and (c) 9 87
3.19 CADAM test case with splitting:water level registered at different gauges 88
3.20 CADAM test case with splitting: comparison between FV and SPH-SWEs models at gauges (a) 4, (b) 6 and (c) 9 89
3.21 CADAM test case with splitting: comparison between FV and SPH-SWEs models at gauges (a) 4, (b) 6 and (c) 9 90

4.1 boundary at the inflow zone: boundary and fluid particle management 96
4.2 boundary at the inflow zone: boundary and fluid particle management 96
4.3 Steady transcritical flow over a bump with a shock (test i): velocity (a), water surface elevation (b) and discharge (c) 99
4.4 Steady transcritical flow over a bump without a shock (test ii): velocity (a), water surface elevation (b) and discharge (c) 100

4.5 Steady supercritical flow over a bump (test iii): velocity (a), water surface elevation (b) and discharge (c) 100
4.6 Steady supercritical flow over a bump (test ii): L_s water depth norm 101
4.7 2-D Uniform flow in a sloping rectangular channel: particle position and velocity vectors 102
4.8 2-D Uniform flow in a sloping rectangular channel: (a) water depth map, (b) velocity magnitude map 102
4.9 2-D Uniform flow in a sloping rectangular channel: water depth (a), velocity v_x (b) and velocity v_y (c) evaluated at sections y=100, y=200 and y=300 103
4.10 1993 Okushiri tsunami: bathymetry of the domain and three gauges position 104
4.11 1993 Okushiri tsunami: water level elevation imposed at x=0 . . . 105
4.12 1993 Okushiri tsunami simulation without splitting: 3D view (left) and contour (right) of water depth at times 10, 12, 14 and 15 s . . 107
4.13 1993 Okushiri tsunami simulation without splitting: 3D view (left) and contour (right) of water depth at times 16, 17, 18 and 20 s . . 108
4.14 1993 Okushiri tsunami simulation without splitting: surface elevation at three gauges 109
4.15 1993 Okushiri tsunami simulation with splitting: 3D view (left) and contour (right) of water depth at times 10, 12, 14 and 15 s . . 110
4.16 1993 Okushiri tsunami simulation with splitting: 3D view (left) and contour (right) of water depth at times 16, 17, 18 and 19 s . . 111
4.17 1993 Okushiri tsunami simulation with splitting: surface elevation at three gauges 112
4.18 1993 Okushiri tsunami simulation: zoom in the area with maximum water elevation at time 17 s without (a) and with particles splitting simulation (b) 112
4.19 Thamesmead test case: bathymetry of the site and position of the 6 gauges, the color axis is expressed in meter 113
4.20 Thamesmead test case: discharge boundary condition imposed through the breach 113
4.21 Thamesmead test case: contour plot of TUFLOW simulation at times 1, 2 and 3 hours 115
4.22 Thamesmead test case: contour plot of SPH simulation at times 1, 2 and 3 hours 116

4.23 Thamesmead test case: contour plot of TUFLOW simulation at times 4,5 and 6 hours . . . 117
4.24 Thamesmead test case: contour plot of SPH simulation at times 4, 5 and 6 hours . . . 118
4.25 Thamesmead test case: water depth registered at different gauges 119

5.1 example of free surface detection . . . 130
5.2 Stationary fluid in a box: particle displacement with PM (left) and IPPM (right) after 1 s of simulation . . . 131
5.3 Stationary fluid in a box: comparison of adimensional pressure obtained with PM and IPPM method after 1s of simulation at the vertical section x=0.5 m . . . 131
5.4 Taylor-Green flow with Re=1000: particles position and velocity field for a particle resolution of 80 × 80 at time 2 s . . . 132
5.5 Taylor-Green flow with Re=1000: comparison of analytical (red lines) and numerical (dots) v_x at $y = 0$ for a particle resolution of 80 × 80 at times 0.5,2,5,10 s . . . 133
5.6 Taylor-Green flow with Re=1000: comparison of analytical (red lines) and numerical (dots) v_y at $x = 0$ for a particle resolution of 80 × 80 at times 0.5,2,5,10 s . . . 133
5.7 Taylor-Green flow with Re=1000: variation of L_2 error with particle spacing at time 2s . . . 134
5.8 Evolution of an initially circular fluid patch: FPM solution at times $tA_0 = 0.2, 0.4, 0.6, 2.0$ s. The colour contours represent the non dimensional pressure field $p/\rho A_0^2 R^2$, the dashed red line is the ellipse calculated by the reference solution . . . 135
5.9 Non-dimensional pressure $p/(\rho A_0^2 R^2)$ of evolution of an initially circular fluid: patch: comparison between reference solution (dashed line) and FPM (dots) at x=0 (above) and y=0 (below) . . . 136
5.10 Non-dimensional velocities of evolution of an initially circular fluid patch: comparison between reference solution (dashed line) and FPM (dots); $v_y/(A_0 R)$ at x=0 (above) and $v_x/(A_0 R)$ at y=0 (below) 136
5.11 Initial stages of a dam break flow with initial water depth ratio of 0.1: profiles of free surface elevation obtained at successive times of 0.024, 0.04, 0.066 and 0.08 s. The FPM (dots) and the reference solution (continuous line) are plotted . . . 137

5.12 Initial stages of a dam break flow with initial water depth ratio of 0.45: profiles of free surface elevation obtained at successive times of 0.02, 0.03, 0.052 and 0.076 and 0.08 s. The FPM (dots) and the reference solution (continuous line) are plotted 137

List of Tables

1.1 $|m_0 - 1|$ calculated for different points using no boundary methods, mirrored particles, MVBP and original VBP 15

2.1 L_2 norm of non-dimensional water depth error calulated for 1-D dam break with wet bed at time 50 s and considering different stabilization terms: Artificial Viscosity (AV), Lax-Friedrichs (LF), Lax-Friedrichs with MUSCL reconstruction and Two-Shock Riemann solver (TS) with MUSCL reconstruction 58
2.2 L_2 non-dimensional norm of water depth, v_x and v_y error calulated 2-D cilindrical dam beak with wet bed at times 10, 30 and 50 s and considering different stabilization terms: Lax-Friedrichs (LF), Lax-Friedrichs with MUSCL reconstruction and Two-Shock Riemann solver (TS) with MUSCL reconstruction 62

3.1 results of particle split procedure: splitting parameters α and ϵ, global splitting error $\mathscr{E}$, relative optimal mass distribution for central particle λ_1 and for other particles $\lambda_{2\ldots7}$ 72

4.1 Boundary conditions for a 1D steady flow over a bump 98

Chapter 1

Zeroth-order Consistent Boundary Conditions

1.1 Literature review of closed boundary methods

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian meshless method originally introduced to simulate astrophysical problems by Gingold and Monaghan [1977], Lucy [1977] where no solid boundaries were present. A variety of new solutions have been suggested by many authors (Randles and Libersky [1996], Kulasegaram et al. [2004], Lee et al. [2008], Hieber and Koumoutsakos [2008]) and each one has advantages and drawbacks. The work in this session is motivated by trying to apply an SPH-based solver for the shallow-water equations to cases where other techniques for solid boundaries have failed, most notably the ghost-particle technique (Randles and Libersky [1996]). In this chapter certain methods for simulating solid boundary condition are analysed, and a novel method based on the idea of the virtual particles (Ferrari et al. [2009]) is introduced, enhanced and implemented in a shallow water equation model.

Imposing boundary conditions in meshless methods in general and in SPH in particular is still an open problem. This is due to the intrinsic nature of kernel based interpolation; the interpolated value of a function $\langle f \rangle_i$ is, in general, different to the exact value at the same point. The first approach as proposed in Monaghan [1994], Monaghan et al. [2003] is the repulsive force method; the key idea is to describe the wall by particles which exert a repulsive short-range force similar to a Leonard-Jones potential force on fluid particles.

Mirror or ghost particles as introduced by Randles and Libersky [1996] is another widely used way to describe boundaries in SPH. Kulasegaram et al. [2004] proposed a variant of this method; instead of using additional particles they introduced an additional term in the momentum equation in order to mimic the effect of the wall. This technique eventually uses an empirical function to approximate the force originating from variational principles. However, this becomes particularly unwieldy in calculations in 2-D and 3-D involving many boundaries that may be moving. These methods have the advantage of restoring zeroth-order consistency in the SPH interpolation; indeed the effect of kernel truncation near the boundaries is eliminated by introducing either some artificial particles positioned at a symmetric position (with respect to the boundaries) or an additional term in the momentum equation. Another form of the ghost particle method has been used in Incompressible SPH models (Shao and Lo [2003], Lee et al. [2008]) where dummy particles represent the solid: the drawback of this method is that special treatment for corners is needed and that sometimes particles penetrate the solid boundaries.

The repulsive force method is more flexible because it can be used to describe complex moving boundaries, but it can introduce a non-physical pressure oscillation and it does not reduce the effect of kernel truncation near the wall. In contrast the main drawback of mirror or ghost particle methods is that they are not able to deal with complex geometries in a straightforward way.

Hieber and Koumoutsakos [2008] adopted an immersed boundary technique in order to impose a non-slip boundary condition in an SPH method. This approach has some advantages because the conservation of physical quantities is ensured and it is able to handle complex boundaries. Unfortunately a key aspect of the immersed boundary method in an SPH formulation is a remeshing procedure. Therefore it cannot be easily applied to flows where moving interfaces are present such as free-surface flows or shallow water flow with wet/dry interfaces.

1.2 Zeroth-order consistency in SPH formalism

In the continuous domain the SPH interpolation method of a scalar function $f(\mathbf{x})$ is based on the following integral:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\, W(\mathbf{x}-\mathbf{x}', h)\, \mathrm{d}\mathbf{x}' \tag{1.1}$$

$$\nabla f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\ \nabla W(\mathbf{x}-\mathbf{x}', h)\ \mathrm{d}\mathbf{x}' \tag{1.2}$$

where the integral is over the domain Ω and the smoothing length h is the parameter that determines the size of the support for the weighting function, $W(\mathbf{x}-\mathbf{x}', h)$ (Monaghan [1992] demonstrated that the SPH summation is second-order accurate). In the discrete domain the integral of equations (1.1) and (1.2) are approximated numerically by summations:

$$\langle f \rangle_i = \sum_j^N f_j W_i(\mathbf{x}_j, h_i) V_j \tag{1.3}$$

$$\langle \nabla f \rangle_i = \sum_j^N f_j \nabla W_i(\mathbf{x}_j, h_i) V_j \tag{1.4}$$

where $\langle \cdots \rangle$ denotes the SPH approximation, V_j is the volume associated with the jth particle, N is the number of particles inside a circle with $2h$ radius and centred at point $\mathbf{x}_i$.

In general the exact value of the function f_i is different from the SPH interpolation $\langle f \rangle_i$. Many kernel functions are proposed in literature (Li and Liu [2003]), in this work a cubic spline kernel is adopted:

$$W(R) = \alpha_d \times \begin{cases} \frac{2}{3} - R^2 + 0.5R^3 & R \leq 1 \\ \frac{1}{6}\left(2-R\right)^3 & 1 < R \leq 2 \\ 0 & R > 2 \end{cases} \tag{1.5}$$

where $R = |x - x_i|/h$, in one and two dimensional space $\alpha_d = 1/h$, $\alpha_d = 15/(7\pi h^2)$.

One of the most important requirements for a kernel function is that the zeroth moment of the kernel function is equal to 1:

$$\int_{\Omega} W(\mathbf{x} - x', h)\ \mathrm{d}\mathbf{x}' = 1 \tag{1.6}$$

this is called the zeroth-order consistency property. Using the summation approximation of the integral in equation (1.6) the same property is defined as the zeroth moment:

$$m_0 = \sum_j^N W_i(\mathbf{x}_j, h_i) V_j = 1 \tag{1.7}$$

In general in SPH, because of the kernel truncation effect, in a bounded domain $m_0 \neq 1$ for particles close to the boundary. This means that the SPH interpolation is not zeroth-order consistent for those particles and numerical inaccuracies and instabilities arise. Therefore the approximation of m_0 to unity can be considered a basic criterion for assessing the quality of a method for imposing solid boundary conditions.

1.3 Virtual Boundary Particle method

Recently, Ferrari et al. [2009] introduced a new method called the Virtual Boundary Particle (VBP) method where virtual particles are placed along boundary walls which in turn are used to generate virtual interior mirror-like particles for each flow particle near the wall. In this way complex boundaries may be readily handled. These particles on the boundary walls are called *virtual* because they do not move with the fluid. The virtual wall particles generate virtual interior particles using a local point-symmetry: if the distance between a fluid particle i and a virtual wall particle v is less than $2h$, then a new virtual interior particle k is generated and it is positioned at $\mathbf{x}_k = 2\mathbf{x}_v - \mathbf{x}_i$. It is emphasised that the particles are called virtual because they are used just for interpolating the physical quantities of fluid particle i, and thereafter they are discarded.

We perform some analysis considering different shapes of boundary with the aim of finding out the best distance between virtual wall boundary particles, and a Modified Virtual Boundary Particle (MVBP) method is introduced to generalise application for arbitrary geometries while minimising the errors associated with kernel truncation.

The stencil plotted in Figure 1.1 is obtained for a generic particle positioned far away from the boundary, considering a 2-D domain discretized using an idealized set of particles positioned in a square grid of size dx and taking the smoothing length $h = 1.2dx$, typical in SPH. An ideal boundary method should be able to reproduce the same stencil for particles that are close to the boundaries, regardless of its shape.

Figure 1.2 shows neighbours (or stencil) for a particle close to a straight boundary, generated by (a) VBP and (b) MVBP. Taking the distance between virtual wall particles $dx_b = 0.5dx$ the interior column of the stencil of Figure 1.2-a exactly reproduces a local artificial rectangular array of particles. This is thus an appropriate value for dx_b.

In the original VBP method, if one flow particle is interacting with a virtual wall particle, then only one virtual interior particle is generated. In the MVBP method the number of virtual interior particles generated by each virtual wall particle is increased to 2: they are positioned at

$$\begin{aligned} \mathbf{x}_{k,1} &= 2\mathbf{x}_v - \mathbf{x}_i \\ \mathbf{x}_{k,2} &= 4\mathbf{x}_v - \mathbf{x}_i \end{aligned} \tag{1.8}$$

where $\mathbf{x}_{k,1}$, $\mathbf{x}_{k,2}$ are the positions of the two interior particles, $\mathbf{x}_v$ is the position of the virtual wall particle interacting with the fluid particle i.

The second modification introduced in the MVBP method is that two particles are added inside corners with internal angles $\leq 180°$ (see Figure 1.3) and this further reduces the kernel truncation effect for those singular points in comparison with the original VBP.

The Figures 1.2 - 1.5 show the comparison between the stencils obtained using the original VBP and the MVBP for different geometries of the boundaries: the modifications we introduce allow the reproduction of stencils that are more similar to the idealised one plotted in Figure 1.1. Strictly speaking, we can actually place virtual wall particles at spacings of $0.25dx$ along the boundary in order to generate virtual particles at $(x_i + 2dx, y_i \pm dx)$ where x and y are 2D Cartesian coordinates. However, it can be shown that these have a far smaller effect than the virtual particles at $(x_i + 2dx, y_i)$. The same comparison between the original and modified VBP method considering a curved boundary is plotted in Figure 1.4. Finally Figure 1.5 shows the stencil of a particle close to a 270° internal angle. In this case both the original and the modified VBP reproduce the same stencil.

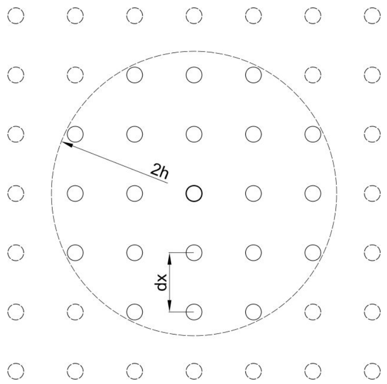
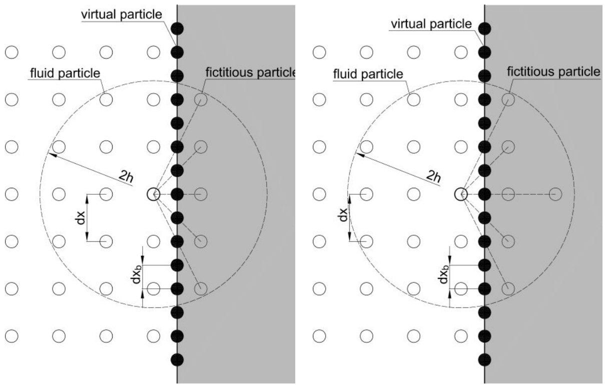
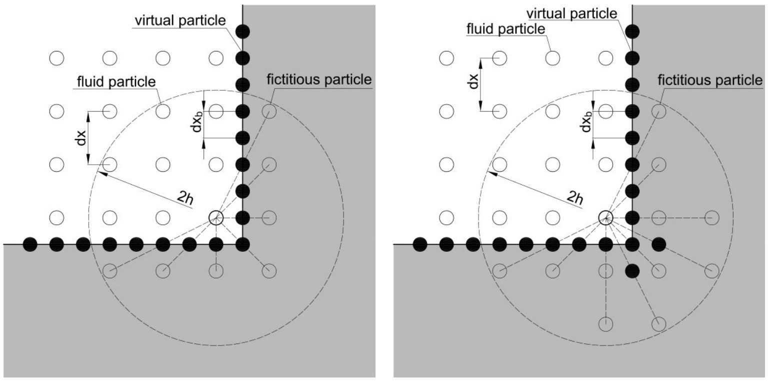


Figure 1.1. stencil of a internal particle


virtual particle
fluid particle
fictitious particle
2h
dx
dxb
virtual particle
fluid particle
fictitious particle
2h
dx
dxb

(a) (b)

Figure 1.2. particle close to a boundary straight line: stencil generated using (a) the original VBP and (b) the MVBP

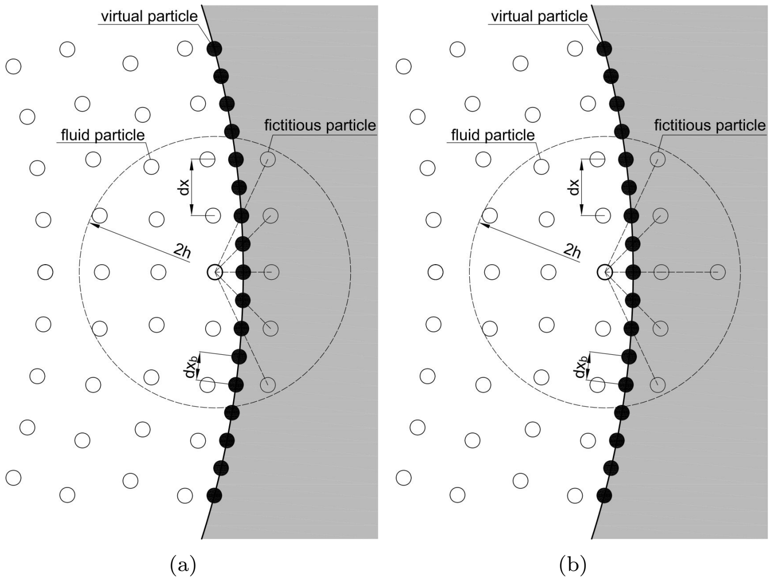

virtual particle
fluid particle
fictitious particle
dx
dxb
2h
virtual particle
fluid particle
fictitious particle
dx
dxb
2h

(a) (b)

Figure 1.3. particle close to 90° internal angle: stencil generated using (a) the original VBP and (b) the MVBP

1.4 Zeroth-order consistent evaluation for a range of possible wall configurations

With the aim of testing the capability of the MVBP method to deal with any 2-D boundary shape some numerical tests are performed by calculating the zeroth-order consistent condition m_0, (Equation 1.7), for different domains.
Figure 1.6 shows m_0 calculated in a square domain filled with 1600 disordered


virtual particle
fluid particle
fictitious particle
dx
2h
dx_b
virtual particle
fluid particle
fictitious particle
dx
2h
dx_b

(a) (b)

Figure 1.4. particle close to a curved boundary: stencil generated using (a) the original VBP and (b) the MVBP

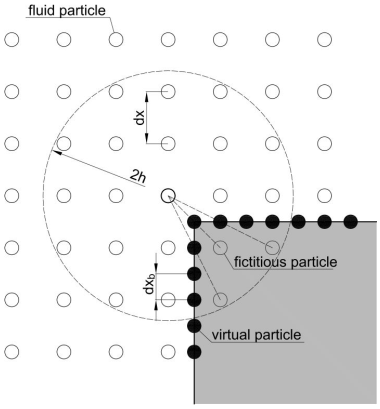

fluid particle
dx
2h
fictitious particle
dx_b
virtual particle

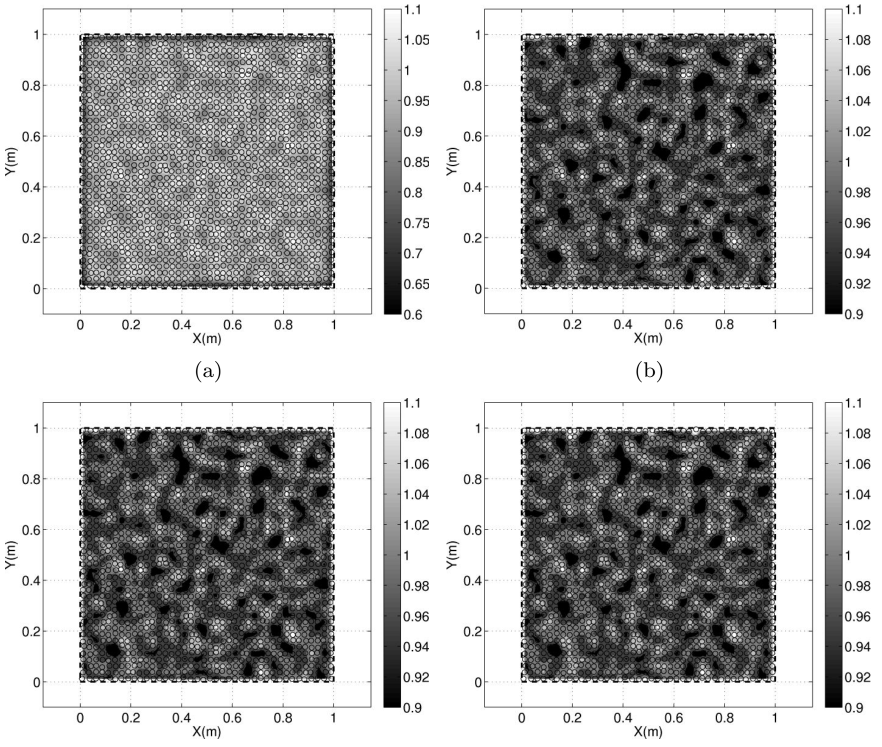
Figure 1.5. internal particle close to 270° internal angle: stencil generated using the original VBP and the MVBP (there is no difference) particles, considering (a) no boundary treatment, (b) the mirror ghost particle method (c) the VBP method and (d) the MVBP method: for this simple geometry (b), (c) and (d) methods reproduce approximately the same values and they are all able to restore approximate zeroth-order consistency near the boundaries. Here we have plotted the variation of m_0 between the values of 0.9 to 1.1 to compare with the no-boundary version in Figure 1.6-a. However it will be seen below that the choice of boundary treatment does make a difference in the corners that is not visible in these plots.

The same comparison, (but placing the particles over a uniform Cartesian grid) is then performed with an L-shaped domain (see Figure 1.7) and a square domain with an internal angle of 270° (see Figure 1.8). The mirror particle method (Randles and Libersky [1996]) is not able to deal with this kind of boundary because the value of m_0 in the proximity of internal angles $\geq 180°$ is overestimated: the maximum error of m_0 obtained by MVPM and VPM for both L-shape and internal angle of 270° geometries is two orders of magnitude less than that obtained by virtual particles (see Table 1.1 discussed below). Finally a domain with the curved boundaries of a toroid shape is considered (see Figure 1.9). In Figure 1.10 the value of m_0 along y=0 is plotted: the MVBP method produces the best estimate of m_0, the mirror particle method overestimates m_0 for particles close to the inner boundary.

Table (1.1) shows the zero-th moment error $|m_0 - 1|$ for different singular points, such as the particle closest to the corner in Figure 1.3, with the aim of assessing the quality of different boundary methods: no boundary treatment, mirror particles, original VBP and MVBP methods. The singular points considered are: a point close to a straight line (Figure 1.2), points closest to 90° corner (Figure 1.3), 270° corner in the L-shape (Figure 1.5) and the 270° (Figure 1.8) internal angles in the square, a point closest to an external circle (Figure 1.4) and an internal circle boundary (Figure 1.9).

For the straight line and the internal 90° angle, mirror particles and the MVBP method reproduce similar results, the error produced by mirror particles is slightly less because in the MVBP method the particles furthest apart in the stencil are still missing as explained in section 1.3. This difference will be seen not to influence the results of still water simulation presented in section 1.6 and therefore can be considered negligible. Conversely the error reproduced by the VBP method is remarkably bigger than the one obtained by MVBP.

For points closest to a 270° angles in the L-shape and square domains, both VBP and MVBP reproduce remarkably better results than mirror particles which over-estimate significantly m_0. For the external circular boundary the results obtained by MVBP are one order of magnitude better than those reproduced by mirror particles and VBP produces errors bigger than MVBP. Finally for the internal circular boundary both MVBP and VBP produces better results than mirror particles, for this case the difference between MVBP and VBP is negligible due to the particular choice of the smoothing length and to the radial distribution of fluid particles.



(a)

(b)

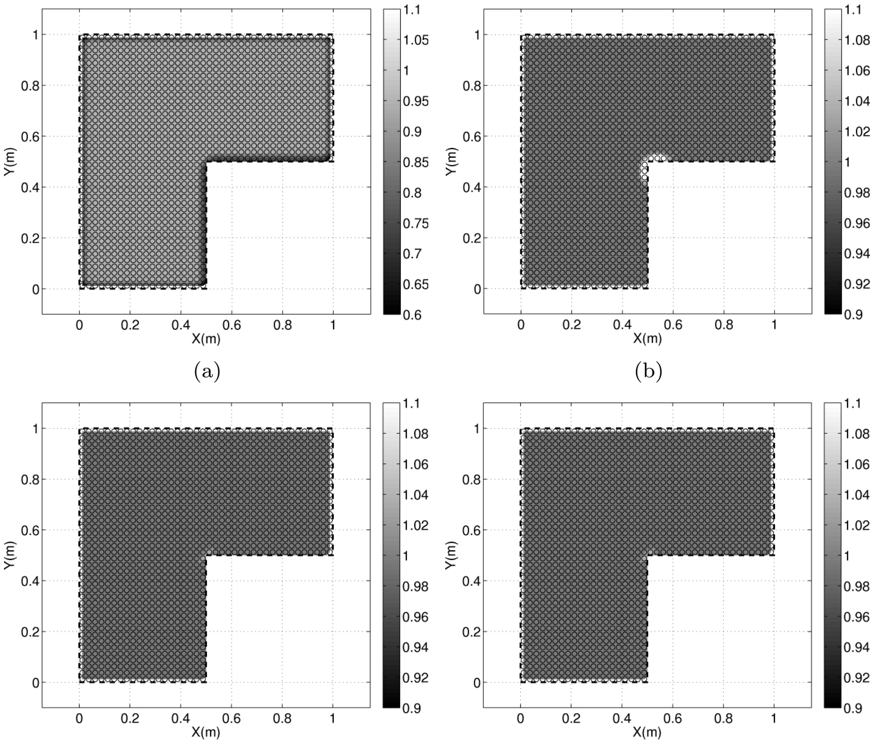
(c)

(d)

Figure 1.6. m_0 calculated for a set of disordered particles in a square bounded domain: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP

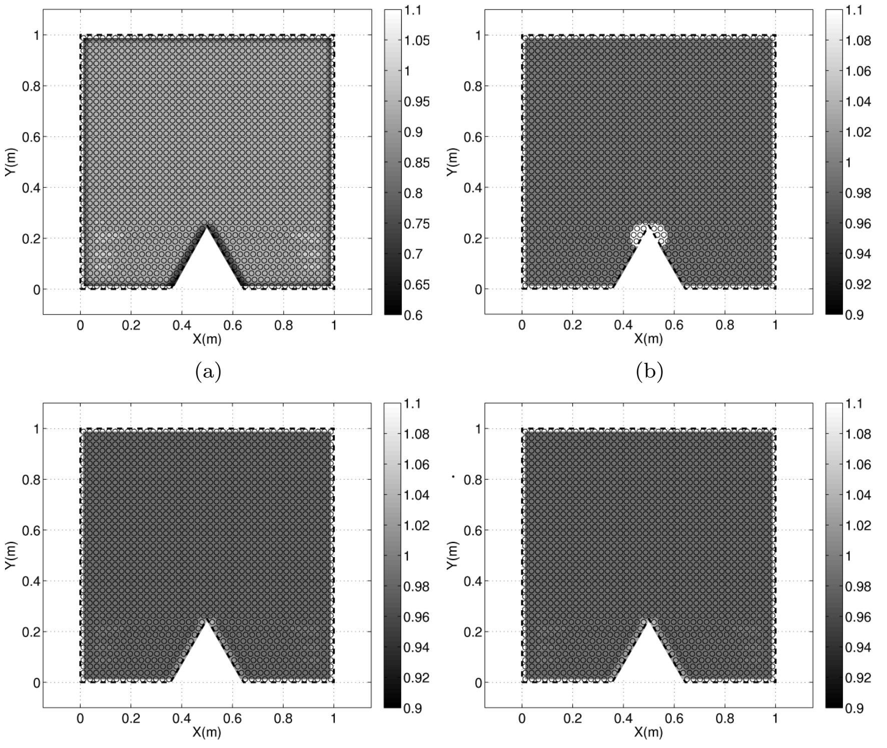
1.5 Correction for variable smoothing length

In the SPH models for Shallow Water Equations a variable smoothing length is considered (see chapter 2). At the begin of the simulation the initial smoothing length of the particles h_0 is $h_0 = cdx$ where dx is the initial interparticle spacing



(a) (b) (c) (d)

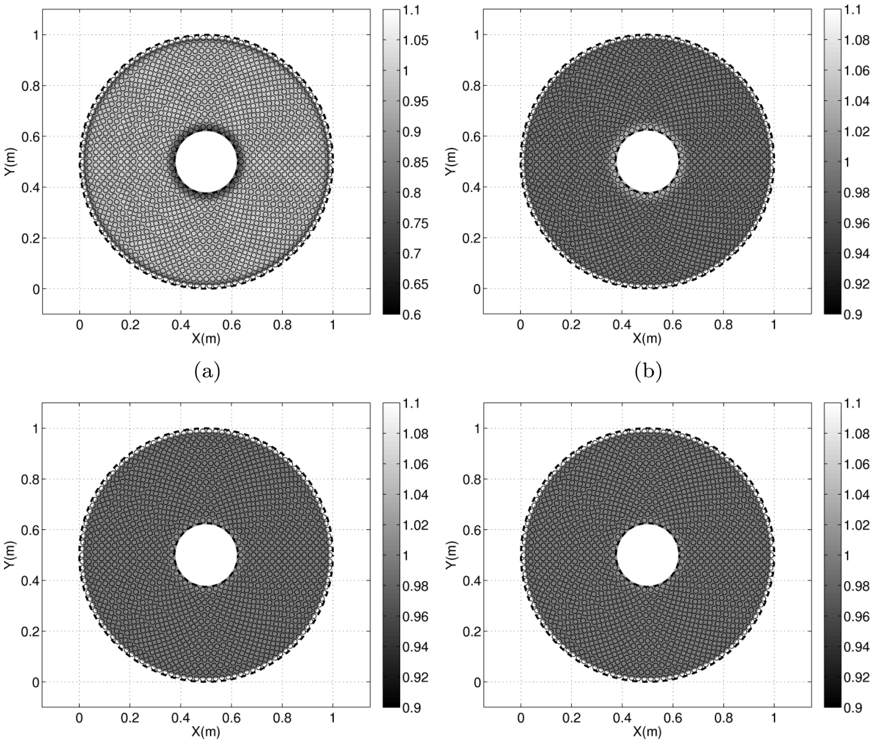
Figure 1.7. m_0 calculated in a bounded domain with an 270° internal angle with an L-shape geometry: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP



(a) (b)

(c) (d)

Figure 1.8. m_0 calculated in a bounded domain with 270° internal angle: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP



(a) (b)

(c) (d)

Figure 1.9. m_0 calculated in a bounded domain with a toroid shape: (a) no boundary condition, (b) mirrored particle method, (c) VBP and (d) MVBP

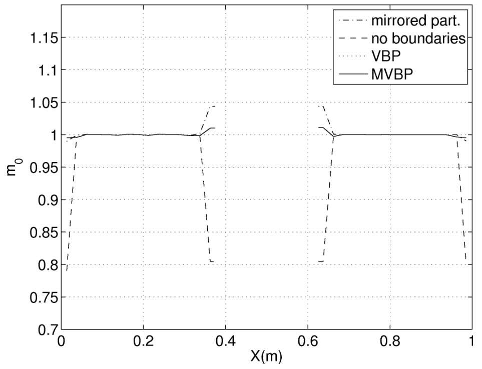
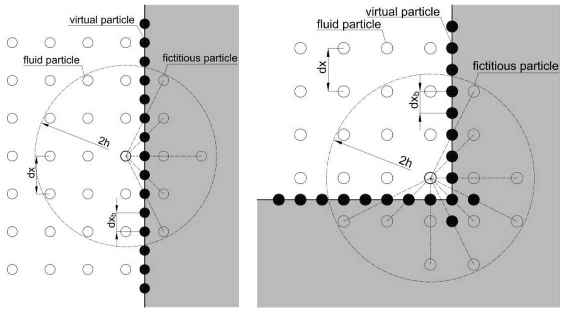


Figure 1.10. comparison of m_0 calculated in a bounded domain with a toroid shape in the section y=0 using no boundary condition, mirrored particle method, VBP and MVBP

Table 1.1. $|m_0 - 1|$ calculated for different points using no boundary methods, mirrored particles, MVBP and original VBP

	no boundary	mirrored part.	VBP	MVBP
straight line	$2.15E-001$	$2.00E-004$	$3.60E-003$	$6.00E-004$
90°	$3.86E-001$	$2.00E-004$	$6.90E-003$	$6.00E-004$
270° L-shape	$4.44E-002$	$4.39E-002$	$2.00E-004$	$2.00E-004$
270°	$2.53E-001$	$7.46E-001$	$3.15E-002$	$1.87E-002$
circ external	$2.21E-001$	$1.07E-002$	$4.80E-003$	$1.90E-003$
circ internal	$1.95E-001$	$4.40E-002$	$1.04E-002$	$1.35E-002$



(a) (b)

Figure 1.11. stencil generated for a particle close to a boundary: (a)-straight wall and (b)-internal angle of 90°

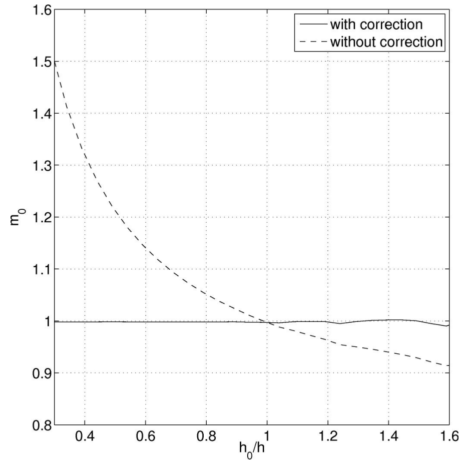
and $c = 1.2$ is a coefficient Balsara [1995], and the virtual boundary particles are placed along the boundaries at a distance $dx_b = 0.5dx$ because this is the best value to keep the zeroth-order consistency (see section 1.2).

When the smoothing length h of the i-th fluid particle become bigger than the initial value h_0 then the particle i interacts with an higher number of virtual particles and the zeroth-moment $m_0 > 1$. To restore the zeroth-order consistency even in presence of variable h the mass associated to the virtual particle j $m_{v,j}$ is corrected as:

$$m_{v,j} = \frac{h_0}{h_i} m_i \tag{1.9}$$

where m_i is the mass of i-th fluid particle.

Figure 1.12 shows m_0 calculated for a particle close to a straight wall (see Figure 1.11-a) using both the corrected and uncorrected mass for virtual particle: the correction of Equation (1.9) restore the zeroth-order consistency condition ($m_0 \simeq 1$) even in the presence of variable smoothing length.



(a)

Figure 1.12. zero-th moment m_0 calculated with and without the variable smoothing length correction of Equation (1.9)

1.6 Still shallow water test cases

In this section the results of several test cases are presented, they are carried out using the SPH Shallow Water numerical model described in detail in chapter 2. In all of the tests still water with different bottom topographies and with or without a wet-dry interface is simulated. All the tests omit bottom friction, which would damp any motion, because this is the most effective way to verify the source bed gradient treatment described in section 2.3 and to compare the different methods previously examined for boundary condition (see section 1.3). For every 1-D test case a convergence analysis is also performed calculating the norm of water depth error $L_2(d)$:

$$L_2(d) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - d_{i,ex})^2} \tag{1.10}$$

where N is the total number of particles in the domain, d_i and $d_{i,ex}$ are the numerical and analytical water depth of ith particle.

The norm of the velocity error $L_2(v)$ is also calculated as follows:

$$L_2(v) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i)^2} \tag{1.11}$$

In order to verify the attainment of a steady condition in time a global relative error L_s (Zhou et al. [2001b]) is calculated at each time step:

$$L_s = \sqrt{\sum_{i=1}^{N} \left(1 - \frac{d_i^{n-1}}{d_i^n}\right)^2} \tag{1.12}$$

where d_i^{n-1} and d_i^n are the water depth at the previous and current time step. When the norm has reached a value of $1 \cdot 10^{-3}$ then the steady condition is said to be achieved and the simulation is stopped.

1.6.1 1-D

1.6.1.1 Bed with submerged and surface-piercing humps

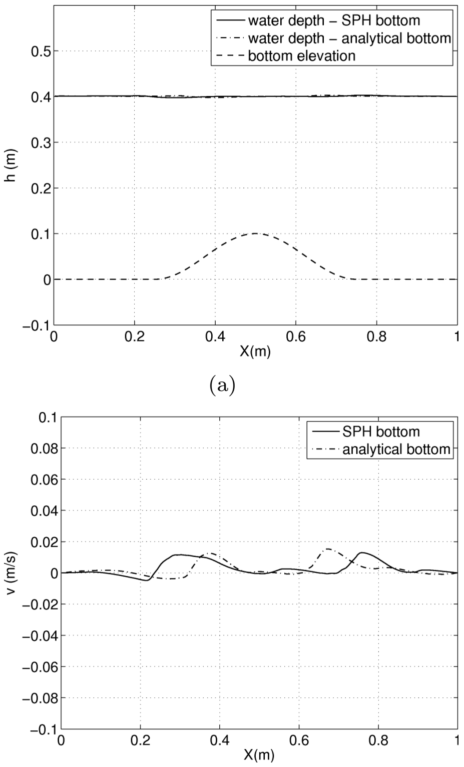
In the first two test cases a bottom with a hump is considered with and without a wet/dry interface, in both tests the equation of the bottom b is:

$$\begin{cases} b = c(1 + \sin(\pi(4x + 0.5))) & \text{if } 0.25 < x < 0.75 \\ b = 0 & \text{otherwise} \end{cases} \tag{1.13}$$

where c is equal to 0.05m and 0.25 m in the first and second test. Initial particle spacing $dx = 0.04$, 0.02, 0.01 and 0.005 m are considered and the particle spacing used for bottom particles is $dx_b = dx$ in every simulation.

Figure (1.13) shows the free surface and the velocity field for an initial particle spacing of $dx = 0.005$ m; the results are plotted using both the analytical and SPH interpolation of the bed gradient source term. The water depth obtained by the numerical model is in good agreement with the analytical solution. The two methods for the bed source term reproduce an analogous small maximum non-dimensional velocity $\left(|v_{max}|/\sqrt{d_{max}g}\right) < 0.01$. The variation of velocity might appear large in Figure 1.13-(b), but the variation is less than 1% of $\sqrt{d_{max}g}$. In Figures (1.14) the norms of water depth $L_2(d)$ and velocity $L_2(v)$ errors are plotted: both converge with a convergence rate almost equal to 1.

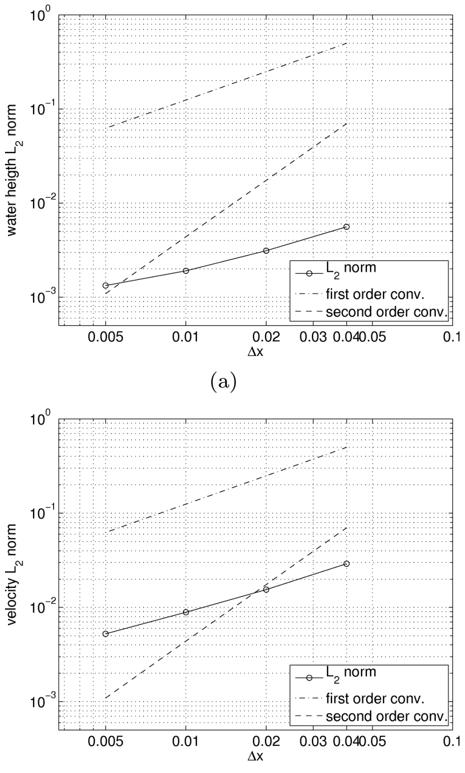
This analysis is repeated for a surface-piercing hump in Figures 1.15 and 1.16 where the depth and the velocity are shown to be convergent. The rate of convergence of the velocity in Figure 1.16-(b) is slow due to the wet-dry interface.



(a)

(b)

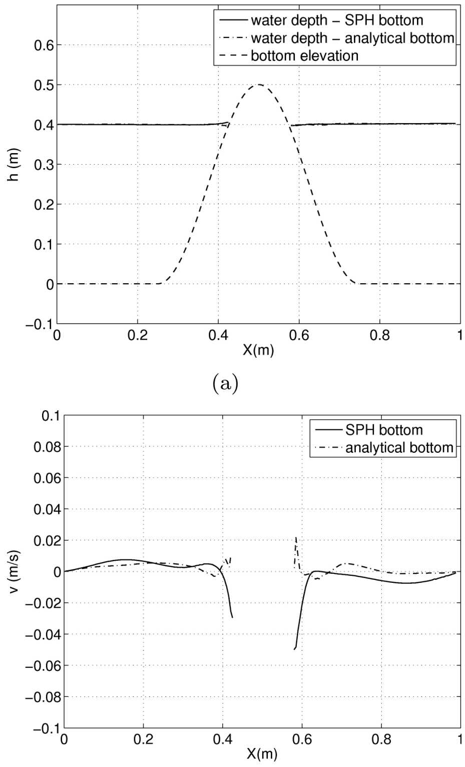
Figure 1.13. still water over a hump: (a) water depth, (b) velocity



(a)

(b)

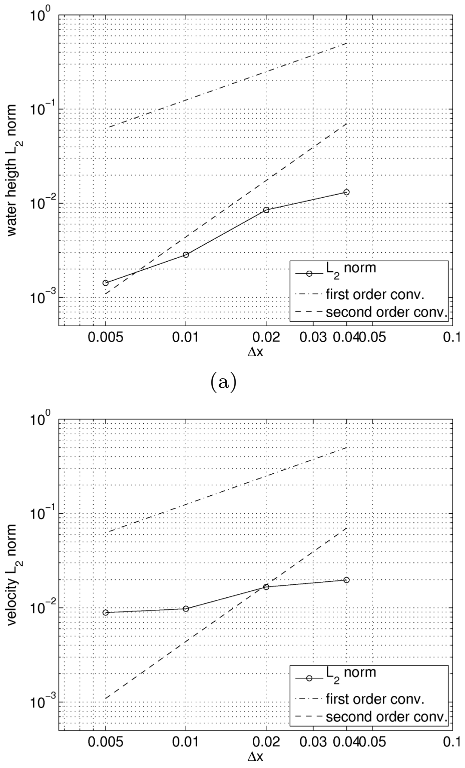
Figure 1.14. still water over a hump, L_2 norm of error: (a) water depth, (b) velocity



(a)

(b)

Figure 1.15. still water over a hump and wetting and drying interface: (a) water depth, (b) velocity



(a)

(b)

Figure 1.16. still water over a hump and wetting and drying interface, L_2 norm of error: (a) water depth, (b) velocity

1.6.1.2 Bed with step

In the third test case still water over a bottom step is simulated, this is a challenging test because of the discontinuity in the bottom elevation b. The SWEs are derived assuming that the bed is slowly varying. However, the SWEs are often applied to cases where there are abrupt changes in bed elevation and they are not strictly valid. The aim here is to examine the effect of applying the SPH-SWE without any form of the balancing techniques (Bermúdez and Vázquez [1994], Vázquez-Cendón [1999], Hubbard and García-Navarro [2000], García-Navarro and Vázquez-Cendón [2000], Zhou et al. [2001b], Rogers et al. [2003]).

Initial particle spacing $dx = 0.04$, 0.02, 0.01 and 0.005 m are considered for fluid particles. In order to deal with the abrupt change in the topography a particle spacing for bottom particles $dx_b = 0.25dx$ is used.

In Figure (1.17) both water elevation and velocity are plotted; we see that the maximum non-dimensional velocity $\left(|v_{max}|/\sqrt{d_{max}g}\right) < 0.02$. The free surface is in good agreement with the analytical solution everywhere but in the neighbourhood of the step: this is due to the fact that the balancing between the SPH approximations of the internal and external forces (equations 2.20 and 2.22) fail in presence of abrupt changes of bed elevation. Despite this, no instabilities are present, a steady state condition is obtained according to Equation (1.12), and both the velocity and the water depth converge (see Figure 1.18). A technique to circumvent this problem over the step is however needed.

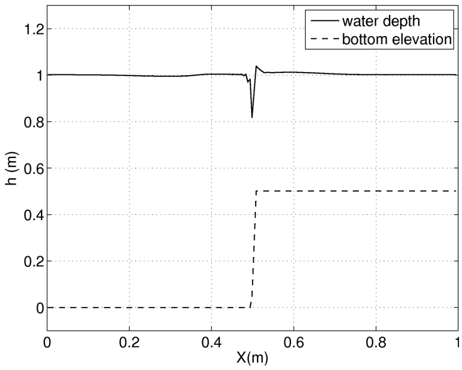
1.6.1.3 Parabolic submerged and surface-piercing beds

In the fourth test case a 1-D parabolic topography is considered and the capability for simulating still water in the presence of a wet/dry interface is also demonstrated. The equation of the bottom is:

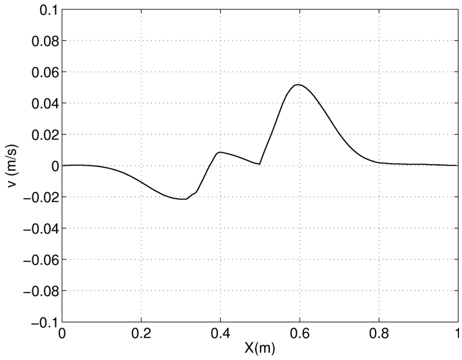
$$b = b_0 \left(\frac{x - 0.5L}{a} \right)^2$$

where b_0=10 m, a=3000 m and the dimension of the domain is L=10000 m. The still water level is 10 m.

Initial particle spacings of dx = 80, 40, 20 and 10 m are considered for fluid particles and the bottom particle spacing is $dx_b = dx$. The simulations are run using both an analytical and SPH discretization of the bed source term. In Figure 1.19 the water depth and velocity for the simulation with the smallest dx

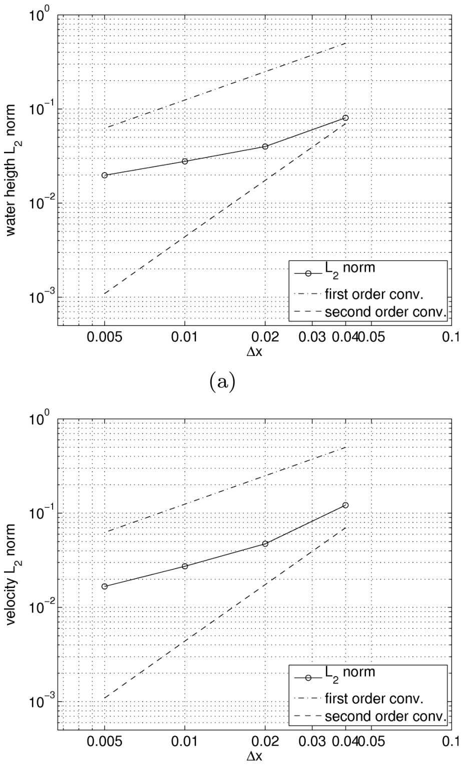


(a)



(b)

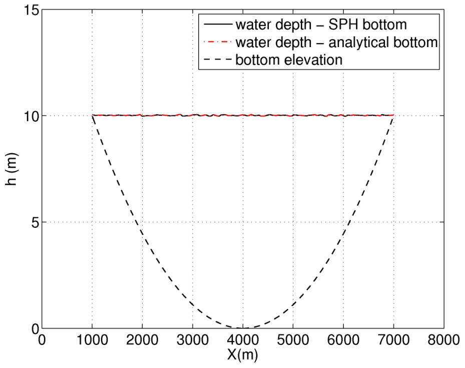
Figure 1.17. still water over a step: (a) water depth, (b) velocity



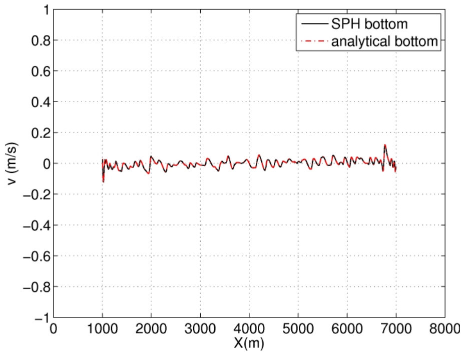
(a)

(b)

Figure 1.18. still water over a step, L_2 norm of error: (a) water depth, (b) velocity are plotted, the water depth is in good agreement with the analytical solution and the two methods for the bed source term reproduce the same velocity field, with the maximum non-dimensional velocity $\left(|v_{max}|/\sqrt{d_{max}g}\right) < 0.02$. In Figure (1.20) the norms $L_2(d)$ and $L_2(v)$ are plotted and they show that both the water depth and the velocity converge to the exact solution with a convergence rate that is less than 1. This is due to the fact that near the wet-dry interface there is a kernel truncation effect that reduces the accuracy of SPH interpolation. In the last 1-D test case the parabolic bottom and the same initial particle spacing is considered. However there is no wet/dry interface but there are side walls instead. The dimension of the domain is L=3000 m and two wall boundary conditions at x=0 m and x=3000 m are imposed. As in the previous test case both analytical and SPH discretizations of the bed source term are considered. In Figure (1.21) the free surface and the velocity are plotted: the water depth is in

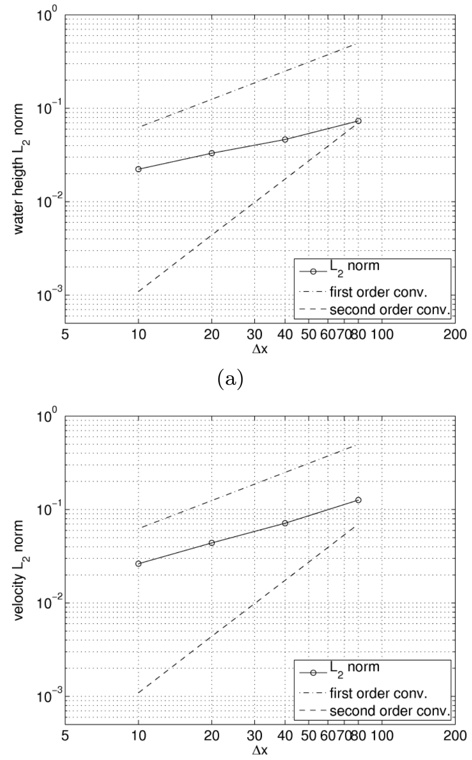


(a)



(b)

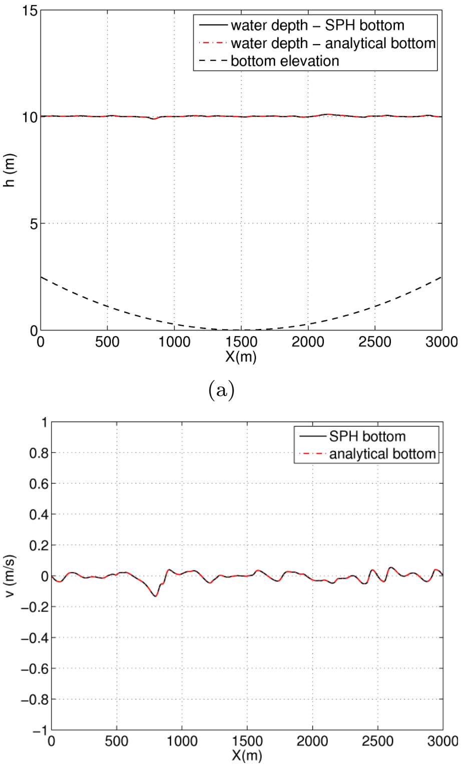
Figure 1.19. still water over a parabolic topography with wetting and drying interface: (a) water depth, (b) velocity



(a)

(b)

Figure 1.20. still water over a parabolic topography with wetting and drying interface, L_2 norm of error: (a) water depth, (b) velocity good agreement with the analytical solution and the maximum non-dimensional velocity $\left(|v_{max}|/\sqrt{d_{max}g}\right) < 0.006$. In Figure (1.22) the norm $L_2(d)$ and $L_2(v)$ are plotted; both converge to the exact solution.



(a)

(b)

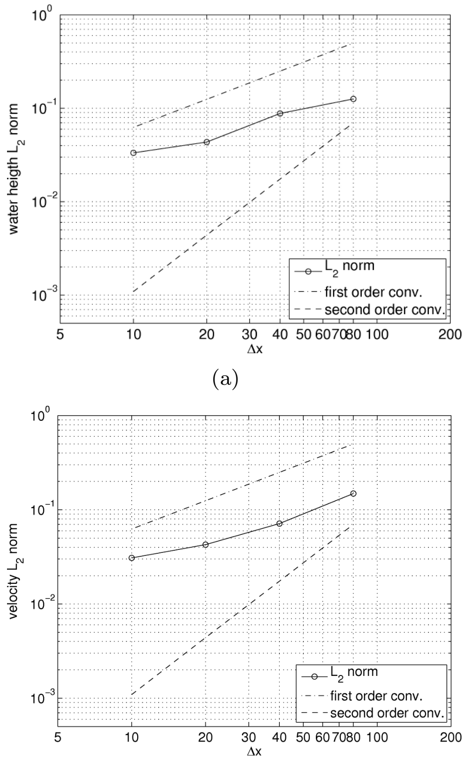
Figure 1.21. still water over a parabolic topography: (a) water depth, (b) velocity

1.6.2 2-D

In these two-dimensional cases both the boundary conditions introduced and examined in sections 1.3 and 1.4, and the bed discretization will be tested.

1.6.2.1 Parabolic basin with surface-piercing and submerged beds

In this paragraph the results of two 2-D test cases simulating still water over a 2-D parabolic topography are investigated; the SPH bottom discretization of Section 2.3 is applied.



(b)

Figure 1.22. still water over a parabolic topography, L_2 norm of error: (a) water depth, (b) velocity

In the first test case the presence of the wet/dry interface is considered and the equation of the bottom is:

$$b = b_0 \frac{(x - 0.5L_x)^2 + (y - 0.5L_y)^2}{a^2} \tag{1.14}$$

where b_0=10 m, a=3000 m and the dimension of the domain are $L_x = L_y$=10000 m. The simulation is run using 448 particles and stopped at time 1000 s, when the condition (1.12) applies. In Figure 1.23 the particle positions and the velocity field are plotted. The water elevation and velocity magnitude maps are plotted in Figure 1.24, where the maximum velocity magnitude is less than 1.5% of $\sqrt{gd_{max}}$. Finally Figure 1.25 shows the water surface elevation and the velocity magnitude at section $y = 0$. In the second 2-D test case, the parabolic bottom of Equation (1.14) with b_0=10 m, a=3000 m and the dimensions of the domain are $L_x = L_y$=3000 m. A circular solid boundary condition with a radius at 1500 m is imposed. This test case is simulated using 699 fluid particles and two different methods for boundary conditions: in Figures (1.26) and (1.27) the results are plotted using the MVBP method whereas Figures (1.28) and (1.29) show results obtained using the mirror particle method. We can see that the results obtained using the two method are very similar: the maximum velocity magnitude is less than 0.4% of $\sqrt{gd_{max}}$ and the free surface elevation is in good agreement with the analytical solution. Therefore for a circular concave geometry, the choice of boundary condition is not critical. As will be demonstrated next, the choice of boundary condition is important for more irregular geometries.

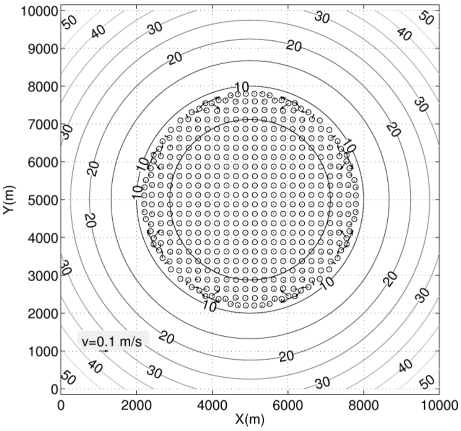
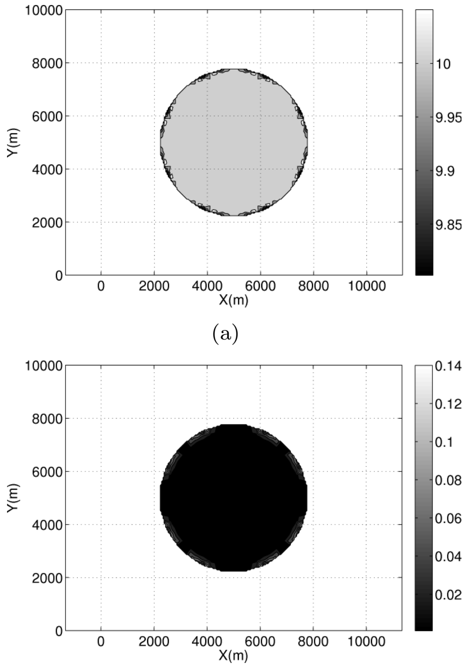


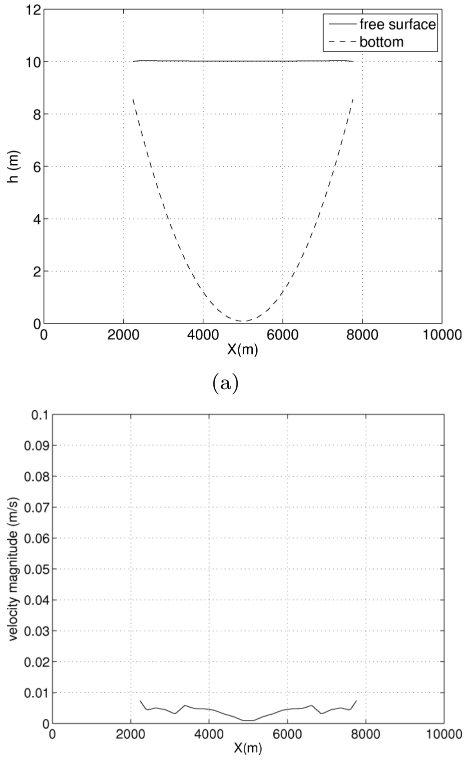
Figure 1.23. still water over a parabolic topography 2-D at time 1000 s: velocity field and particle positions



(a)

(b)

Figure 1.24. still water over a parabolic topography 2-D with wetting and drying interface at time 1000 s: (a) free surface elevation, (b) velocity magnitude



(a)

(b)

Figure 1.25. still water over a parabolic topography 2-D with wetting and drying interface:, section y=0: (a) free surface elevation, (b) velocity magnitude

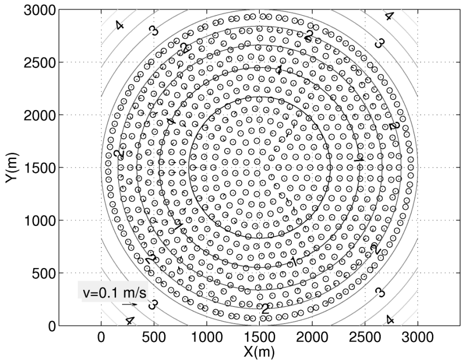
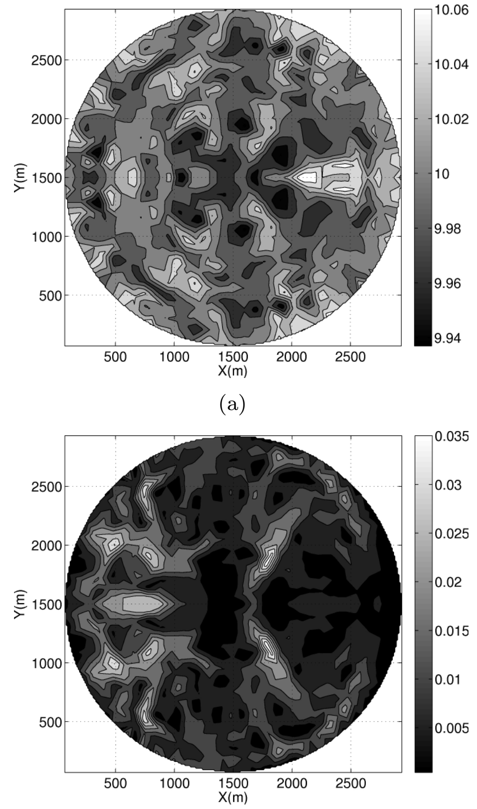


Figure 1.26. still water over a parabolic topography 2-D with MVBP boundary condition at time 1000 s: velocity field and particle positions



(a)

(b)

Figure 1.27. still water over a parabolic topography 2-D with MVBP method at time 1000 s: (a) free surface elevation, (b) velocity magnitude

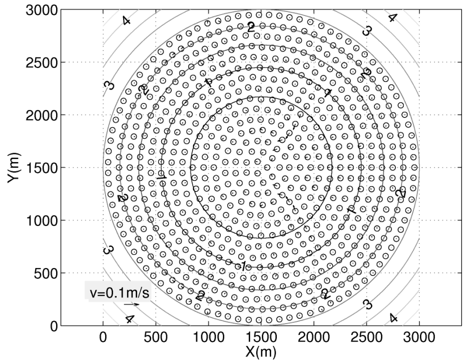
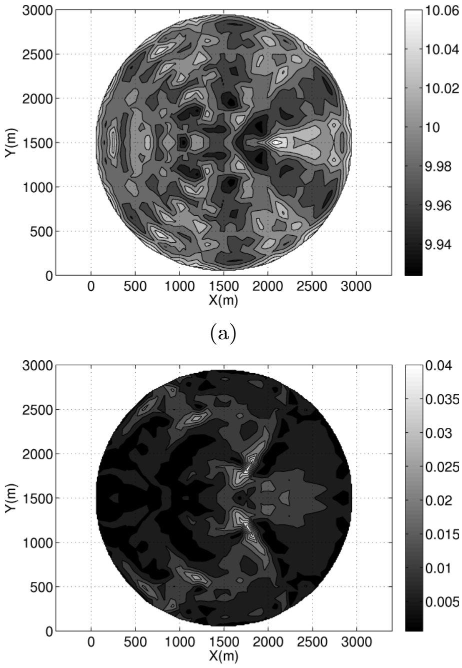


Figure 1.28. still water over a parabolic topography 2-D with mirrored particle boundary condition at time 1000 s: velocity field and particle positions



(a)

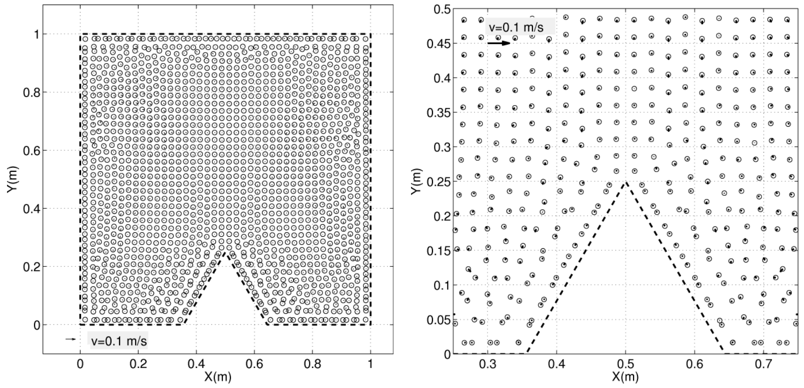
(b)

Figure 1.29. still water over a parabolic topography 2-D with mirrored particle boundary condition at time 1000 s: (a) free surface elevation, (b) velocity magnitude

1.6.2.2 Domains with vertical walls including acute angles

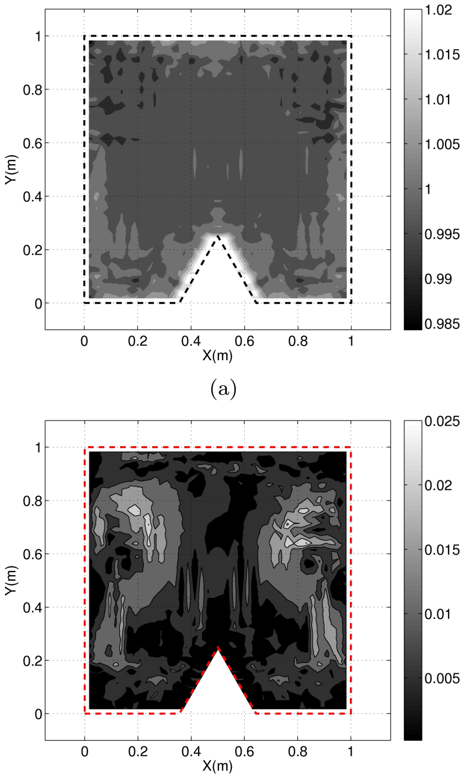
The last test case is done in order to compare the three methods for boundary conditions with a box with an internal angle of 300° and a flat bottom. The simulation is run using 1552 particles, and it is stopped at time 1 s, when (using MVBP) the condition of Equation (1.12) is satisfied. The results obtained using MVBP are plotted in Figure (1.30) and (1.31) and Figures (1.32) and (1.33) shows the particle positions, water level and velocity magnitude obtained using the mirror particle method. The comparison between the two methods shows that the MVBP achieves the best results. In the simulation with the mirror particle method particles near to the internal angle of 300° start to move and this happens because in the area close to that angle it is difficult to generate the mirror particles within the internal angle and avoid placing mirror particles upon fluid particles.

The original VBP method is not able to achieve the steady state condition; the kernel truncation effect generates instabilities and particles start to penetrate the walls. In Figure 1.34 the particle positions and the velocity field are plotted after 0.1 s of simulation. In order to prove the capability of MVBP to deal with large internal angles, i.e. acute solid angles, the same test is repeated with an internal angle equal to 345°. In Figure (1.35) the position of particles after 1 s of simulation is plotted. Figure (1.36) shows the water depth and the velocity magnitude fields at the same time; the steady state condition is reached according to equation (1.12) and the maximum velocity magnitude is less that



(a) (b)

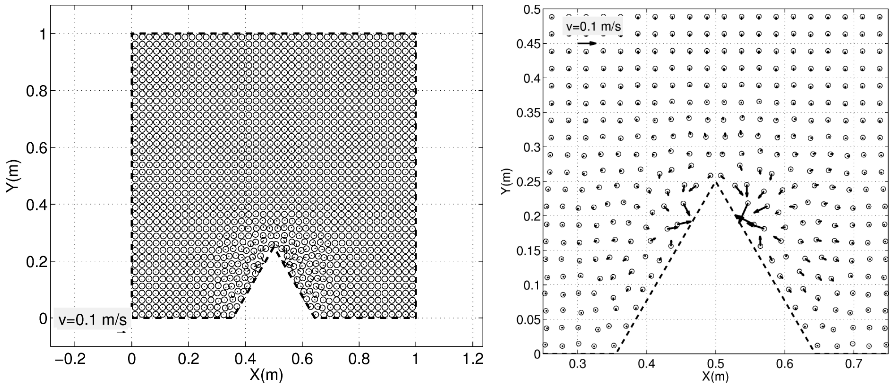
Figure 1.30. still water in a box with an internal angle of 300° with MVBP method at time 1 s: velocity field and particle positions, (a) whole domain (b) zoom



(a)

(b)

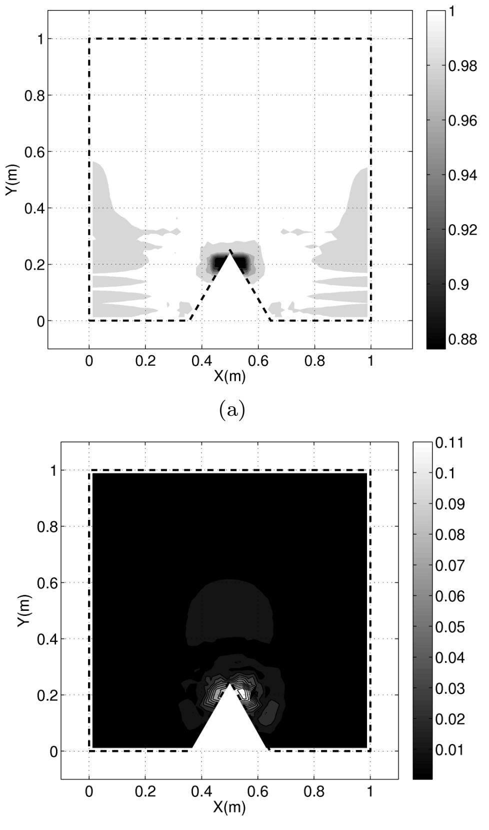
Figure 1.31. still water in a box with an internal angle of 300° with MVBP method at time 1 s: (a) free surface elevation, (b) velocity magnitude



(a)

(b)

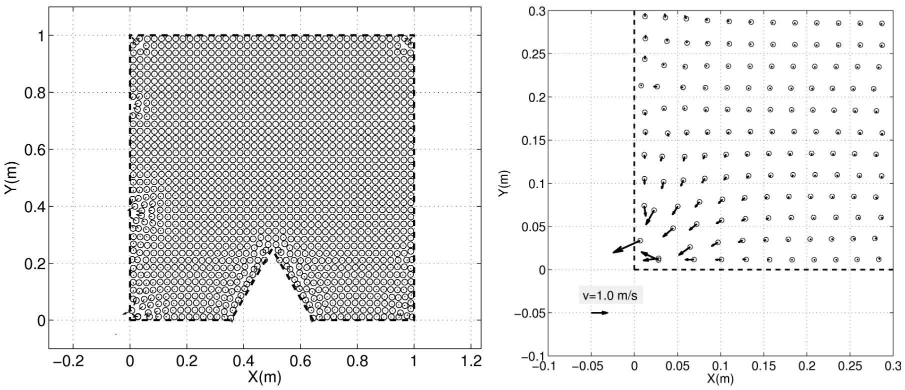
Figure 1.32. still water in a box with an internal angle of 300° with mirrored particle boundary condition: velocity field and particles position, (a) whole domain (b) zoom



(a)

(b)

Figure 1.33. still water in a box with an internal angle of 300° with mirrored particle boundary condition: velocity field and particles position: (a) free surface elevation, (b) velocity magnitude



(a) (b)

Figure 1.34. still water in a box with an internal angle of 300° with VBP: velocity field and particles position after 0.1 s of simulation, (a) whole domain (b) zoom

0.8% of $\sqrt{gd_{max}}$, and is similar to the value obtained in the simulation with an internal angle equal to 300° (see Figure 1.31-a).

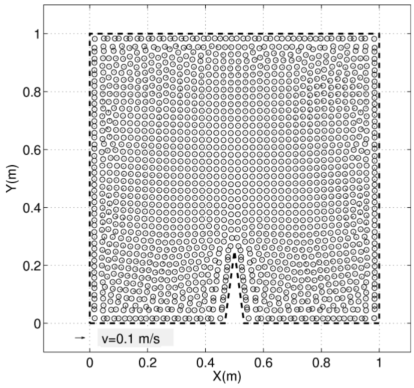
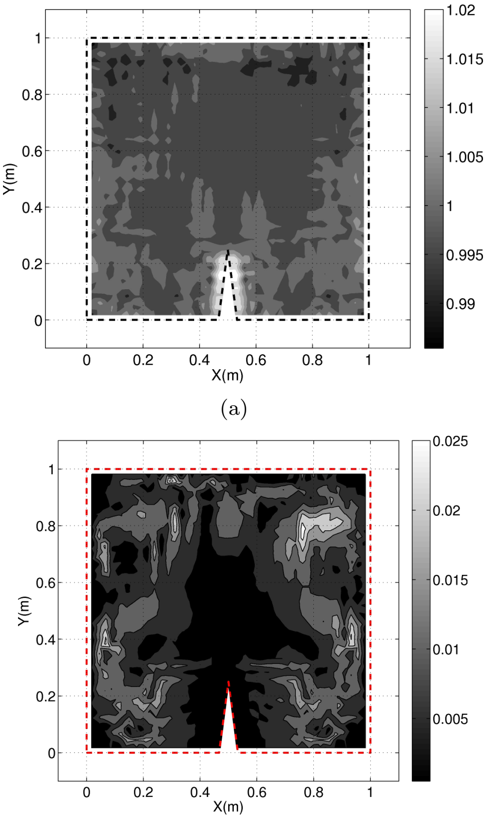


Figure 1.35. still water in a box with an internal angle of 345° with MVBP method at time 1 s: velocity field and particle positions

1.7 Concluding Remarks

An improved method for solid wall boundary conditions in 2-D SPH has been presented. This is motivated for application of SPH to the shallow water equations and is based on the idea of using virtual boundary particles to approximate closely the zeroth-order consistency condition. This has been assessed for various



(a)

(b)

Figure 1.36. still water in a box with an internal angle of 345° with MVBP method at time 1 s: (a) free surface elevation, (b) velocity magnitude shapes including some with large internal angles. The method has been implemented in a shallow water algorithm and tested by reproducing still water in 1-D and 2-D domains. Furthermore, in order to generalise the SPH model for irregular bathymetries, the bed gradient source term has been discretised by introducing a set of bottom particles to define the bed elevation and its derivatives in an SPH interpolation. This new method has been tested by simulating still water in domains of different shape with different bed topographies in the absence of bottom friction: a submerged and a surface-piercing hump, a bed step, a submerged and surface-piercing parabolic bed and domains with an acute solid angle. The bed treatment is able to reproduce motionless water with an accuracy similar to an analytical treatment of the bed source term.

Chapter 2

SPH numerical model for Shallow Water equations

2.1 Introduction

The 2-D SWEs are a widely used description of flows over shallow domains for a great range of rapidly (and slowly) varying free surface flows, for example, dam break flood waves, flood waves in rivers, tides in estuaries. Such equations are derived from the conservation principles of mass and momentum by depth integrating the continuity and Navier Stokes equations over the water depth d. Recently SPH methods have been applied to the SWEs (Rodriguez-Paz and Bonet [2005], Ata and Soulaïmani [2004]) obtaining promising results; these Lagrangian models have some distinct advantages: no mesh is needed, the wet/dry interfaces require no special treatment and the mass is automatically conserved.

In this chapter a 2D Shallow Water code based on the SPH interpolation is derived by means of the variational formulation proposed by Bonet and Lok [1999]. A novel bed gradient source term SPH dicretization is also presented, and finally different formulations of the stabilization term are derived and tested against 1D and 2D circular Dam Breaks reference solutions.

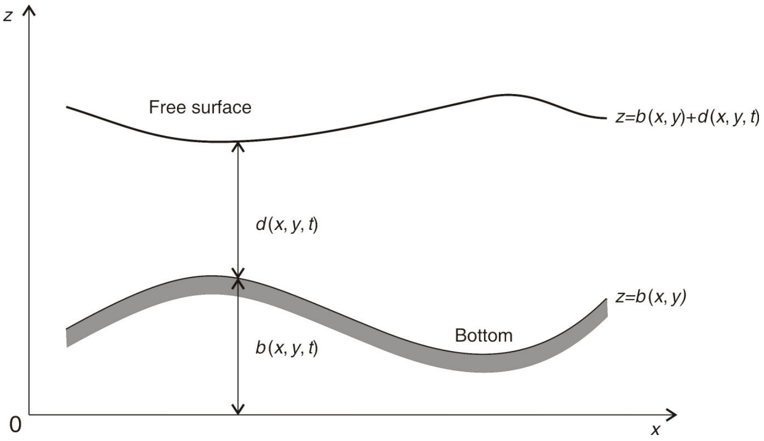


Figure 2.1. flow with a free surface under the effect of gravity

2.2 SPH for Shallow water: Variational formulation

SWEs are formally identical to Euler equation if we re-define the density ρ as the amount of fluid per unit of area in a 2-D domain; given this new definition of ρ we can connect it to the depth of water d with :

$$\rho = \rho_w d \tag{2.1}$$

where ρ_w denote the constant 3D density. Using the definition of Equation 2.1 SWEs can be written as follow:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho \nabla \cdot \mathbf{v} \tag{2.2}$$

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\frac{g}{\rho_w} \nabla \rho + g\left(-\nabla b + \mathbf{S}_f\right) \tag{2.3}$$

where $\mathbf{v}$ is the horizontal velocity vector, b is the bottom elevation, g is the acceleration due to gravity and $\mathbf{S}_f$ is the bed friction source term.

There is a linear dependence of the density ρ to the water depth and ρ_i of a particle i can vary enormously during a simulation, therefore a SPH scheme with variable smoothing length h in time and space should be used in order to keep the number of neighbour particles roughly constant during both inundation and drying stages. Nelson and Papaloizou [1994] have derived an SPH h-variable formulation where ∇h correction terms are included in the momentum equation but this introduces some numerical oscillations (Springel and Hernquist [2002]) and therefore has not found a widespread usage in SPH codes. Bonet and Lok [1999] have derived the SPH equations for a continuum using a variational approach, Springel and Hernquist [2002] extended this new approach for a system of particles with variable smoothing lengths and this leads to an elegant formulation of the momentum equation where a correction factor α is introduced (section 2.2.2). The latter approach is used in this work because is simple and robust and conserves both the mass and the momentum.

2.2.1 Density evaluation

SPH approximation for density of the i-th particle ρ_i is (Monaghan [1992]) :

$$\rho_i = \sum_j m_j W_i(\mathbf{x}_j, h_i) \tag{2.4}$$

In general, h is connected to the density (Benz [1990]) with :

$$h_i = h_0 \left(\frac{\rho_0}{\rho_i}\right)^{1/d_m} \tag{2.5}$$

where ρ_0, h_0 are the initial density and smoothing length for the i-th particle and d_m is the number of space dimensions (1 in 1D and 2 in 2D).

The above equation is implicit because the density is itself a function of h_i as reported in Equation (2.4). In this paper a simple Newton - Raphson iteration is adopted in order to solve this system of two Equations (2.4) and (2.5).

The residual of the density ψ is defined at the kth iterative cycle as

$$\psi\left(\rho_i^k\right) = \rho_i^k - \sum_j m_j W_i\left(\mathbf{x}_j, h_i\right) \tag{2.6}$$

The root of Equation (2.6) can be found using Newton Raphson iterative formula:

$$\rho_i^{k+1} = \rho_i^k - \frac{\rho_i^k - \sum_j m_j W_i\left(\mathbf{x}_i, h_i\right)}{\left[\frac{d\psi}{d\rho}\right]_i^k} \tag{2.7}$$

the derivative of the residual is calculated differentiating Equation (2.6) and using the chain rule:

$$\frac{d\psi}{d\rho} = 1 - \sum_j m_j \frac{dW_i\left(\mathbf{x}_i, h_i\right)}{dh_i}\frac{dh_i}{d\rho_i} \tag{2.8}$$

substituting Equation (2.27) into Equation (2.8) and remembering that $(dh/d\rho) = -h/(d_m\rho)$ leads to

$$\frac{d\psi}{d\rho} = 1 - \frac{1}{\rho_i}\sum_j m_j W_i\left(\mathbf{x}_i, h_i\right) + \alpha_i \tag{2.9}$$

where α is defined in Equation (2.32).
Substituting Equation (2.9) into Equation (2.7) gives the final iterative formula for ρ_i

$$\rho_i^{k+1} = \rho_i^k\left[1 - \frac{\psi_i^k}{\psi_i^k + \rho_i^k\alpha_i^k}\right] \tag{2.10}$$

The initial guess $\rho_{i,n+1}^0$ for starting the iteration is calculated integrating in time Equation(2.30):

$$\rho_{i,n+1}^0 = \rho_{i,n}e^{\gamma_n} \tag{2.11}$$

where

$$\gamma_n = \frac{1}{\alpha_i\rho_i}\left[\sum_j m_j \frac{dW_{ij}}{dr_{ij}}\left(\mathbf{x}_i - \mathbf{x}_j\right)\cdot\left(\mathbf{v}_i - \mathbf{v}_j\right)\right]$$

The Newton Raphson iterations can be conducted independently for each particle and it will be stopped when

$$\frac{|\psi_i^{k+1}|}{\rho_i^k} \leq \varepsilon_\Psi \tag{2.12}$$

The coefficient ε_Ψ affects remarkably the speed of the code because it controls the number of iteration needed to calculate the water depth d of each particle. Taking ε_Ψ $1\cdot 10^{-3}$ allows to reduce the simulation time and does not affect very much the precision of the results, therefore we use this value in our simulations.

2.2.2 Momentum equation

The Euler-Lagrange equation of motion for a particle i is (Marion and Thornton [1988]):

$$\frac{d}{dt}\frac{\partial L}{\partial \mathbf{v}_i} - \frac{\partial L}{\partial \mathbf{x}_i} = 0 \tag{2.13}$$

where the Lagrangian functional L is defined in term of kinetic energy K and potential energy π as: $L = K - \pi$. π is a function of particles position but not of their velocity, so substituting this expression into Equation (2.13) leads to:

$$\frac{d}{dt}\frac{\partial K}{\partial \mathbf{v}_i} - \frac{\partial K}{\partial \mathbf{x}_i} = \frac{\partial \pi}{\partial \mathbf{x}_i} \tag{2.14}$$

The kinetic energy for a system of particles can be approximated as the sum of energy of each particle:

$$K = \frac{1}{2}\sum_i m_i \left[\mathbf{v}_i \cdot \mathbf{v}_i + v_z^2\right]; \qquad v_z = \mathbf{v}_i \cdot \nabla b_i \tag{2.15}$$

where v_z is the vertical component of the velocity. This term is usually neglected in classical SWE, but due to the Hamiltonian approach it is possible to include it in our analysis.

2.2.2.1 Potential Energy

Defining a system of coordinates as in Figure (2.1) the potential energy of each column of water can be evaluated in the baricenter (this is because of the hydrostatic pressure assumption). So π can be expressed as a sum of potential energy of each particle:

$$\pi = \pi_{ext} + \pi_{int} = \sum_i m_i g b_i + \frac{1}{2}\sum_i m_i g h_i \tag{2.16}$$

where g is the gravity acceleration and b_i is the bottom elevation of particle i. The first term π_{ext} is an external potential energy and the second π_{int} is considered as an internal one.
The Equation (2.14) is equivalent to Newton's second law as:

$$\mathbf{I}_i = \mathbf{F}_i - \mathbf{T}_i \tag{2.17}$$

where $\mathbf{I}_i$, $\mathbf{F}_i$ and $\mathbf{T}_i$ are inertial, external and internal forces and they are:

$$\mathbf{I}_i = \frac{d}{dt}\frac{\partial K}{\partial \mathbf{v}_i} - \frac{\partial K}{\partial \mathbf{x}_i} \quad ; \mathbf{F}_i = \frac{\partial \pi_{ext}}{\partial \mathbf{x}_i} \quad ; \mathbf{T}_i = \frac{\partial \pi_{int}}{\partial \mathbf{x}_i} \tag{2.18}$$

Inertial forces can be evaluated with the help of Equation (2.15) as:

$$\mathbf{I}_i = \frac{d}{dt}\left[m_i \mathbf{v}_i + m_i \left(\mathbf{v}_i \cdot \nabla b_i\right) \nabla b_i\right] - m_i \left(\mathbf{v}_i \cdot \nabla b_i\right) \mathbf{k}_i \mathbf{v}_i \tag{2.19}$$

where $\mathbf{k}_i = \nabla\left(\nabla b_i\right)$ is the curvature tensor of $b(x, y)$.
Substituting the Equation (2.16) in second Equation of (2.18) gives :

$$\mathbf{F}_i = m_i g \nabla b_i; \tag{2.20}$$

The discretization of ∇b_i is obtained by the method described in section 2.3. The formulation for internal force $\mathbf{T}$ is obtained using the continuity equation and the internal energy expressed in term of energy per unit mass (see 2.2.3):

$$\mathbf{T}_i = \sum_j m_i m_j \left(\frac{p_j}{\alpha_j \rho_j^2} \nabla W_j(\mathbf{x}_i, h_j) - \frac{p_i}{\alpha_i \rho_i^2} \nabla W_i(\mathbf{x}_j, h_i)\right) \tag{2.21}$$

substituting the pressure p obtained by means of hydrostatic law: $p = 0.5 g d^2$ the final formulation for $\mathbf{T}$ is:

$$\mathbf{T}_i = \sum_j m_i m_j \frac{g}{2\rho_w} \left(\frac{1}{\alpha_j} \nabla W_j(\mathbf{x}_i, h_j) - \frac{1}{\alpha_i} \nabla W_i(\mathbf{x}_j, h_i)\right) \tag{2.22}$$

Finally particles accelerations $\mathbf{a}$ can be found by substituting Equations (2.19), (2.20) and (2.22) in (2.17)

$$\mathbf{a}_i = \frac{g + \mathbf{v}_i \cdot \mathbf{k}_i \mathbf{v}_i + \mathbf{t}_i \cdot \nabla b_i}{1 + \nabla b_i \cdot \nabla b_i} \nabla b_i - \mathbf{t}_i \tag{2.23}$$

where $\mathbf{t}_i = \mathbf{T}_i / m_i$.

2.2.3 Internal force calculations

Differentiating Equation(2.4) and using the chain rule for the kernel leads to:

$$D_v\rho = \sum_j m_j D_v W_i(x_j, h_i) = \sum_j m_j \left[\frac{d\,W_i(x_j, h_i)}{d\,r_{ij}} D_v r_{ij} + \frac{d\,W_i(x_j, h_i)}{d\,h_i} D_v h_i \right] \tag{2.24}$$

where D_v indicates the directional derivative and r_{ij} is the distance between particle i and particle j. Differentiating Equation(2.5) and using the chain rule again we can obtain de directional derivative of h:

$$D_v h = -\frac{h}{d_m \rho} D_v \rho \tag{2.25}$$

the kernel is a function of $R = r_{ij}/h_i$ and in general has the form $W_{ij} = 1/h^{d_m} f(R)$. where W_{ij} is a short form for $W_i(x_j, h_i)$ The derivative of the kernel function is:

$$\frac{dW_{ij}}{dr_{ij}} = \frac{1}{h^{d_m+1}} \frac{dW_{ij}}{dR} \tag{2.26}$$

and the derivative of the kernel function respect to h is obtained by means of the chain rule:

$$\frac{dW_{ij}}{dh_i} = -\frac{d_m}{h_i^{d_m+1}} \frac{dW_{ij}}{dR} - \frac{r_{ij}}{h_i^{d_m+2}} W_{ij} = -\frac{1}{h_i} \left(d_m W_{ij} + r_{ij} \frac{dW_{ij}}{dr_{ij}} \right) \tag{2.27}$$

and differentiating r_{ij} we obtain

$$D_v r_{ij} = \frac{1}{r_{ij}} \left(\mathbf{x}_i - \mathbf{x}_j \right) \cdot \left(\delta \mathbf{v}_i - \delta \mathbf{v}_j \right) \tag{2.28}$$

where $D_v \mathbf{x} = \delta \mathbf{v}_i$ because of definition of directional derivative.
Substituting Equations (2.25) (2.27) (2.28) into Equation (2.24) leads to:

$$\begin{aligned} D_v\rho &= \sum_j m_j \frac{dW_{ij}}{dr_{ij}} \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}} \cdot \left(\delta \mathbf{v}_i - \delta \mathbf{v}_j \right) \\ &+ \sum_j m_j \frac{1}{h_i} \left(d_m W_{ij} + r_{ij} \frac{dW_{ij}}{dr_{ij}} \right) \frac{h_i}{\rho_i d_m} D_v \rho_i \end{aligned} \tag{2.29}$$

splitting in two the second summation in the RHS of previous equation leads to:

$$\begin{aligned}\left(1-\sum_j \frac{m_j}{\rho_i} W_{ij}\right) D_v\rho_i = \sum_j m_j \frac{dW_{ij}}{dr_{ij}} \cdot \frac{\mathbf{x}_i-\mathbf{x}_j}{r_{ij}} \cdot (\delta\mathbf{v}_i-\delta\mathbf{v}_j) \\ +\frac{1}{d_m}\sum_j \frac{m_j}{\rho_i} r_{ij} \frac{dW_{ij}}{dr_{ij}} D_v\rho_i\end{aligned} \tag{2.30}$$

since $\rho_i = \sum_j m_j W_{ij}$ then the LHS of Equation (2.30) is zero, and we can obtain:

$$D_v\rho_i = \frac{1}{\alpha_i}\left[\sum_j m_j \frac{dW_{ij}}{dr_{ij}} \cdot \frac{\mathbf{x}_i-\mathbf{x}_j}{r_{ij}} \cdot (\delta\mathbf{v}_i-\delta\mathbf{v}_j)\right] \tag{2.31}$$

where α_i is defined as

$$\alpha_i = -\frac{1}{\rho_i d_m}\sum_j m_j r_{ij} \frac{dW_{ij}}{dr_{ij}} \tag{2.32}$$

and remembering that

$$\nabla W_{ij} = \frac{dW_{ij}}{dr_{ij}} \frac{\mathbf{x}_i-\mathbf{x}_j}{r_{ij}} \tag{2.33}$$

it is possible to manipulate Equation(2.31) and obtaining

$$D_v\rho_i = \frac{1}{\alpha_i}\left[\sum_j m_j \nabla W_{ij} \cdot (\delta\mathbf{v}_i-\delta\mathbf{v}_j)\right] \tag{2.34}$$

noting that taking $\delta\mathbf{v}_i = v_i$ and $\delta\mathbf{v}_j = v_j$ gives the derivative of the density

$$\frac{d\rho_i}{dt} = \frac{1}{\alpha_i}\left[\sum_j m_j \frac{dW_{ij}}{dr_{ij}} (\mathbf{x}_i-\mathbf{x}_j) \cdot (\mathbf{v}_i-\mathbf{v}_j)\right] \tag{2.35}$$

which is similar to the standard SPH formulation for the density (Monaghan [1992]), apart for the α_i correction factor.
The total internal energy stored in the group of particles is:

$$\pi_{int} = \sum_{i=1}^{N} V_i^0 U(J_i) \tag{2.36}$$

where N is the total number of particles in the domain, V_i^0 is the volume at initial time for ith particle, U is the stored internal energy per unit of volume, J_i is the compression ratio between initial and current state:

$$J_i = \frac{V_i^0}{V_i} = \frac{\rho_i^0}{\rho_i} \tag{2.37}$$

the pressure p can be defined as: $p = dU/dJ$.
Recalling third equation of (2.18), the directional derivative of the internal energy functional is

$$D_v \pi_{int} = \sum_{i=1}^{N} \mathbf{T}_i \cdot \delta \mathbf{v}_i \tag{2.38}$$

using Equation(2.36), (2.37) and the definition of pressure the directional derivative of π_{int} is (see Bonet and Lok [1999])

$$D_v \pi_{int} = -\sum_{i=1}^{N} m_i \left(\frac{p_i}{\rho_i^2} \right) D_v \rho_i \tag{2.39}$$

substituting in Equation(2.39) the directional derivative of ρ (Equation (2.34)) and rearranging the summations gives

$$D_v \pi_{int} = \sum_i \left[\sum_j m_i m_j \left(\frac{p_j}{\alpha_j \rho_j^2} \nabla W_{ji} - \frac{p_i}{\alpha_i \rho_i^2} \nabla W_{ij} \right) \right] \cdot \delta \mathbf{v}_i \tag{2.40}$$

The comparison of Equation(2.40) with Equation(2.38) gives the expression for the internal force T_i of Equation (2.21).

2.3 Bed gradient source term

In order to deal with irregular bathymetries we introduce a general method for discretizing the bed gradient source term: in Equation (2.23), we need to calculate the gradient ∇b and the tensor $\mathbf{k}$ for every particle of the domain. Rodriguez-Paz and Bonet [2005] used an analytical function for the bed description. For general application we present here an SPH-based interpolation technique that is applicable to any geometry. We thus discretize these two terms ∇b and $\mathbf{k}$ by an SPH interpolation.
This interpolation is performed using not the fluid particles but a new set of

interpolation points called bottom particles. These points are introduced at the beginning of the simulation, they are distributed on a Cartesian uniform grid over the domain and they do not move during the simulation. The only physical quantity associated with bottom particles is the bottom height b and an associated volume V_j (dx_b in 1-D and $dx_b \cdot dy_b$ in 2-D). The bottom elevation of the ith fluid particle b_i is calculated using a SPH summation formula using the bottom particles:

$$b_i = \sum_j b_j^b \bar{W}_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) V_j \tag{2.41}$$

where b_j^b indicates the bottom elevation of the jth bottom particle located at $\mathbf{x}_j^b$, h^b is the constant smoothing length of bottom particles and $\bar{W}_i$ is the kernel for ith particle corrected using a Shepard filter (Randles and Libersky [1996]):

$$\bar{W}_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) = \frac{W_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b)}{\sum_j W_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) V_j} \tag{2.42}$$

To improve the accuracy of the SPH interpolation of the bottom, the gradient of the kernel is corrected by introducing a correction matrix $\mathbf{L}$ as proposed by Bonet and Lok [1999]:

$$\tilde{\nabla} W_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) = \mathbf{L}_i \nabla W_i(\mathbf{x}_i - \mathbf{x}_j^b, h^b) \tag{2.43}$$

the correction matrix $\mathbf{L}$ is calculated as follows:

$$\mathbf{L}_i = \left[\sum_j \nabla W_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) \otimes \left(\mathbf{x}_j^b - \mathbf{x}_i \right) V_j \right]^{-1} \tag{2.44}$$

This correction ensures first order zero-consistency, or in other words that the gradient of any linear function is exactly evaluated. After correcting the kernel, the gradient of the bottom ∇b_i is evaluated using a classical SPH interpolation:

$$\nabla b_i = \sum_j b_j^b \tilde{\nabla} W_i(\mathbf{x}_i - \mathbf{x}_j^b, h^b) V_j \tag{2.45}$$

Several methods are proposed in the literature to approximate second-order derivatives, a recent review of these methods is reported by Basa et al. [2008] and Schwaiger [2008]. Two approaches are possible, the first is based on the evaluation of the second derivatives using a second-order kernel derivative in the SPH interpolation, but this approach produces instabilities and inaccuracy; a second approach is the possibility to use both a finite difference and an SPH interpolation for first derivatives, but here we use a formulation based on an integral approximation (Cleary and Monaghan [1999], Monaghan [2005]):

$$\left(\frac{\partial^2 b}{\partial x^\alpha \partial x^\beta}\right)_i = \sum_j \left(4\frac{x_{ij}^\alpha x_{ij}^\beta}{r_{ij}^2} - \delta^{\alpha\beta}\right) \frac{b_i - b_j^b}{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij} + \eta^2} \mathbf{r}_{ij} \cdot \tilde{\nabla} W_i(\mathbf{x}_i - \mathbf{x}_j^b, h^b) V_j \quad (2.46)$$

where α and β are two generic coordinates, $\eta = 0.01 h_b$, $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j^b$.
The effectiveness of equations (2.41-2.46) is investigated in section 1.6.

2.4 Friction source term

In Equation (2.3) the last term that needs to be discretized is the bed friction source term $\mathbf{S}_f$. It can be rewritten as:

$$\mathbf{S}_f = \mathbf{v}\frac{g n^2 |\mathbf{v}|}{d} \quad (2.47)$$

where n is the Manning coefficient.
The Manning coefficient n is an empirical coefficient that describes bed friction. In order to vary arbitrarly the Manning coefficient all over the domain the same SPH-based interpolation technique presented in section 2.3 is used to discretize the friction source term: a value of the n coefficient is assigned to each bottom particles, and it is calculated at each time step at the ith fluid particles by means of the following SPH interpolation:

$$n_i = \sum_j n_j^b \bar{W}_i(\mathbf{x}_i - \mathbf{x}_j^b, h_b) V_j \quad (2.48)$$

where n_j^b indicates the manning coefficient of the jth bottom particle located at $\mathbf{x}_j^b$.

2.5 Time integration scheme

In order to integrate in time particle positions and velocities we use the leap-frog time integration scheme defined as (Hernquist and Katz [1989]):

$$\begin{aligned}
&\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^{n-1/2} + \Delta t \mathbf{a}_i^n \\
&\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t^{n+1} \mathbf{v}_i^{n+1/2} \\
&\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \tfrac{1}{2} \Delta t \mathbf{a}_i^n
\end{aligned} \tag{2.49}$$

where Δt is the time step. For explicit methods the time step must satisfy a Courant-Friedrichs-Levy (CFL) condition (Toro [1999]). In SPH this condition is imposed considering the smoothing length as a reference length :

$$\Delta t = C_{CFL} \min_{i=1}^{N} \left(\frac{h_i}{c_i + \|\mathbf{v}_i\|} \right) \tag{2.50}$$

where c is the wave propagation speed that is equal to $c = \sqrt{gd}$. C_{CFL} is the Courant number.

The stabilization term described in the following section influences the value of the maximum C_{CFL} that can guarantee the stability: with the artificial viscosity if a Courant numbers bigger then 0.1 is used a numerical instability can be observed, conversely the Lax-Friedrichs flux or the Riemann solver guarantee the numerical stability even if the $C_{CFL} = 0.5$ is used. Therefore using one of the two last stabilization term a significant acceleleration of the numerical code is obtained.

2.6 Postprocessing output

In SPH models the physical quantites are calculated at the fluid particles and no mesh (or particles' connections) are present. The visualization of results obtained in SPH (or any other meshless model) is still a challenging subject (Jang et al. [2008]). Neverthless the possibility to analyze contour or 3-D maps of physical quantities represents an indispensable tool to assess the quality of the results obtained and to compare them with reference solutions. These difficulties are overcame calculating the main physical quantities not only at fluid particles but also over a uniform cartesian grid with side length equal to dx_g; so the contour maps can be drawn by the classical tools avaliable for Eulerian models.

The physical quantities are calculated at the cartesian grid vertex by means of an SPH interpolation based on the fluid particles such as in a remeshing procedure. It is important to underline that this remeshing is used only within the scope of postprocessing the results, and no fluid particle reinizialization is adopted during the simulation.
The generic physical quantity f_v is calculated at vertex with coordinates $\mathbf{x}_v$ by means of the following SPH interpolation:

$$\begin{cases} f_v = \sum_j f_j \bar{W}(\mathbf{x}_v - \mathbf{x}_j, h_j)\frac{m_j}{\rho_j} & \text{if } dx_v < dx_g \\ f_v = \text{blank value} & \text{elsewhere} \end{cases} \tag{2.51}$$

where j is the subcript for fluid particles and $\bar{W}_i$ is the kernel corrected by means of a Shepard filter Randles and Libersky [1996] and dx_v is defined as the minimum distance between the vertex v and the N fluid particles in its neighborhood:

$$dx_v = min\left[\mathbf{x}_v - \mathbf{x}_j, \mathbf{x}_v - \mathbf{x}_2, \cdots \mathbf{x}_v - \mathbf{x}_N\right]$$

As will be seen in chapter 4, this procedure of postprocessing the SPH data is effective: water depth and velocities contour maps can be easly created. The only element that is not satisfactory is the position of the wet-dry front over complicated bathymetry (as in sections 4.5 and 4.6). If the density of the particles at the front is very low the position of the front is not well defined; the reason of that is due to the fact that the smoothing length of the particles close to the front is bigger than the side length of the cartesian grid dx_g therefore the conditon $dx_v < dx_g$ of Equation (2.51) is not satisfied.

2.7 Stabilization term

Balsara showed via Von Neumann stability analysis (Balsara [1995]) that the SPH method can be interpreted as a central finite difference scheme and some viscosity is needed in order to avoid numerical oscillation in presence of shock waves, therefore Equation (2.21) should be modified as follow:

$$\begin{aligned} \mathbf{T}_i = \sum_j m_i m_j \frac{g}{2\rho_w} \Bigg[\left(\frac{1}{\alpha_j} + \Pi_{ij}\right) \nabla W_j(\mathbf{x}_i, h_j) - \\ \left(\frac{1}{\alpha_i} + \Pi_{ij}\right) \nabla W_i(\mathbf{x}_j, h_i) \Bigg] \end{aligned} \tag{2.52}$$

where the parameter Π_{ij} represents the additional numerical viscosity added in order to preserve the stability.
In the original SPH formulation introduced by Monaghan [1988], Π_{ij} is an artificial viscosity turned on when two particles are approaching:

$$\Pi_{ij} = \begin{cases} \frac{-a\bar{c}_{ij}\mu_{ij}+b\bar{c}_{ij}\mu_{ij}^2}{\rho_{ij}} & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} < 0 \\ 0 & \text{if} \quad \mathbf{v}_{ij}\cdot\mathbf{x}_{ij} \geq 0 \end{cases} \tag{2.53}$$

where a and b are parameters that regulate the strength of the artificial viscosity, c is the speed of sound which is $c = \sqrt{gd}$ in SWE and μ_{ij} is:

$$\mu_{ij} = \frac{\bar{h}_{ij}\mathbf{v}_{ij}\cdot\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2+\zeta^2}$$

where ζ is a small quantity to prevent division by zero. Note that in these expressions the notation $f_{ij} = f_i - f_j$ and $\bar{f}_{ij} = 0.5(f_i + f_j)$ has been used.
The main drawback of the artificial viscosity is that the two parameters a and b have to be tuned according to the necessary numerical viscosity which is different for every test case. Ata and Soulaïmani [2004] introduced a new stabilized SPH formulation based on the idea of the Lax-Friedrichs flux where the centred flux $0.5\,[F(x_i) + F(x_j)]$ is replaced by:

$$0.5\,[F(x_i) + F(x_j)] - \lambda\frac{\mathbf{v}_{ij}\cdot\mathbf{x}_{ij}}{|\mathbf{x}_{ij}^2|}$$

where λ is a characteristic wave speed. After simple algebra the following expression of the stabilizing term Π_{ij} is obtained:

$$\Pi_{ij} = \frac{\bar{c}_{ij}\mathbf{v}_{ij}\cdot\mathbf{x}_{ij}}{\rho_{ij}\sqrt{|\mathbf{x}_{ij}|^2+\zeta^2}} \tag{2.54}$$

The main advantage of this formulation, in comparison with Equation (2.53), is that there are no parameters that have to be tuned but the necessary level of viscosity is automatically introduced thanks to the Lax-Friedrichs flux.
An alternative way to deal with shock waves in hyperbolic equation are the Riemann Solvers. They are widely used in finite volume schemes for hyperbolic equations (Toro [1997, 1999]) and there are some attempt to introduce them in SPH formalism: Inutsuka [2002] introduced a Riemann solver for gas dynamic equations, whereas Cha and Whitworth [2003] applied the non-iterative Riemann

solver proposed by Balsara [1994] to isothermal gas. Comparing this approach with the artificial viscosity method, the advantage of the Riemann Solvers is that no extra numerical dissipation is introduced.
In this work we introduced the two-Shock Riemann Solver (Toro [1995]) into the Shallow Water models: the main idea is consider each interaction between i-th and j-th particles as a Riemann Problem and therefore to replace the pressures p_i and p_j in Equation (2.21) with the resultant pressure p^*:

$$\mathbf{T}_i = \sum_j m_i m_j p^* \left(\frac{1}{\alpha_j \rho_j^2} \nabla W_j(\mathbf{x}_i, h_j) - \frac{1}{\alpha_i \rho_i^2} \nabla W_i(\mathbf{x}_j, h_i) \right) \tag{2.55}$$

The pressure p^* is obtained applying the hydrostatic law $p^* = 0.5g(d^*)^2$ and the depth d^* is obtained by the exact Riemann Solver under the assumption that both the non-linear waves are shocks:

$$d^* = \frac{g_l d_l + g_r d_r + v_{l,n} - v_{r,n}}{g_l + g_r} \tag{2.56}$$

where, according to the two-Shock Riemann Solver, d_l and d_r are the left and right water depth, and g_l and g_r are defined as follow:

$$g_k = \sqrt{0.5 \frac{g(d_0 + d_k)}{d_0 d_k}}$$

with $k = l$ or $k = r$ for the left and right state, d_0 is an estimate of the water depth that can be obtained from some other direct Riemann Solvers, for example a two-rarefaction one:

$$d_0 = \frac{1}{g} \left[0.5(c_l + c_r) + 0.25(v_{l,n} - v_{r,n}) \right]^2$$

In order to reduce the level of numerical viscosity a MUSCL-non upwind procedure (Edwards [2006]) is used to reconstruct a generic phisical quantity f of the left and right state of the Riemann problem :

$$\begin{aligned} f_l &= f_j + \frac{1}{2}\Phi(\xi^+) f_{ij} \\ f_r &= f_i + \frac{1}{2}\Phi(\xi^-) f_{ij} \end{aligned} \tag{2.57}$$

where Φ is the minmod slope limiter function (Toro [1999]), $\xi^+ = \nabla_j f \cdot \mathbf{x}_{ij} / f_{ij}$ and $\xi^- = \nabla_i f \cdot \mathbf{x}_{ij} / f_{ij}$. Please nothe that this is a non upwind reconstruction and it is different from the one proposed by Vila [2005] where:

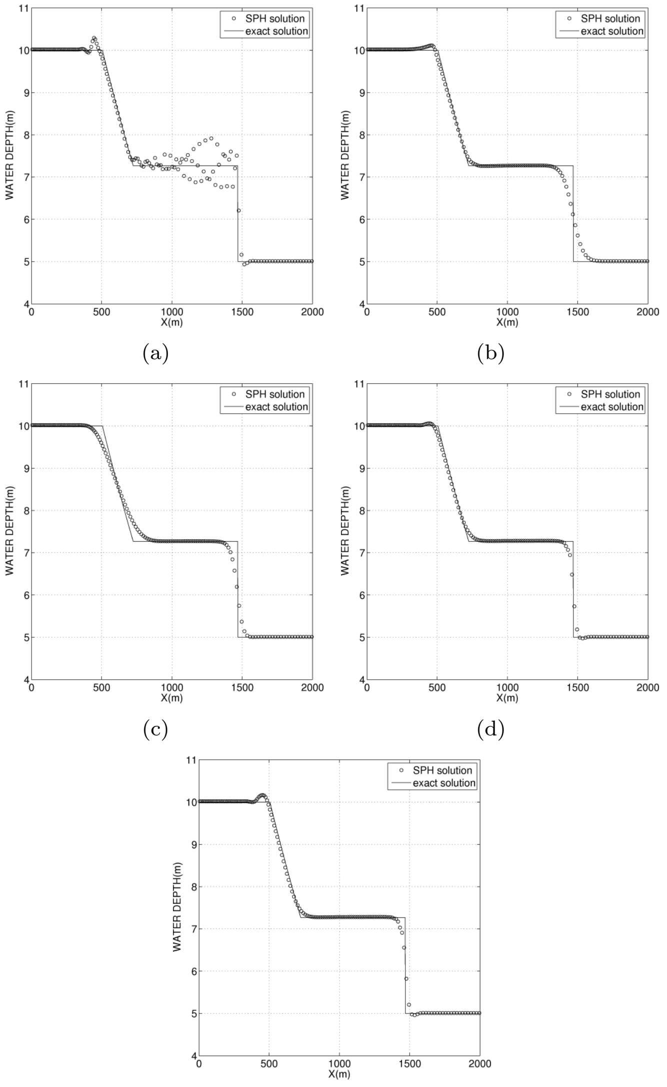
$$
\begin{aligned}
f_l &= f_j + \frac{1}{2}\Phi(\xi^+)\nabla f_{ij} \cdot \mathbf{x}_{ij} \\
f_r &= f_i - \frac{1}{2}\Phi(\xi^-)\nabla f_{ij} \cdot \mathbf{x}_{ij}
\end{aligned}
$$

the approach of Equation (2.57) guarantees that the reconstructed values are $min[f_i, f_j] \leq f_l, f_r \geq max[f_i, f_j]$ and therefore no further limiting has to be applied. The MUSCL reconstruction of Equation (2.57) is used to reconstruct velocities and water depth in the Two-Shock Riemann solver and in the Π_{ij} term of the Lax-Friedrichs flux (see equation 2.54) where $\mathbf{v}_{ij}$ is replaced with $\mathbf{v}_r - \mathbf{v}_l$.
In order to check the differences between the stabilization schemes two different test cases were carried out. The first one is a 1-D dam break with the following initial condition: still water with a water depth $d = 10$ m in the upstream part of the domain ($x < 1000m$) and $d = 5m$ in the downstream part, the domain is dicretized by 150 particles. Figure 2.2 shows the comparison between the Stoker analytical solution and the numerical one obtained using different kind of numerical viscosity. The results in Figure 2.2-a show that some kind of stabilization term is necessary if a shock wave is present, the artificial viscosity term (Equation (2.53), Figure 2.2-b) is able to stabilize the solution but the shock wave is remarkably smeared out, the Lax-Friedrichs flux (Equation (2.54) Figure 2.2-c) is able to reproduce a sharper shock wave but an additionally unnecessary viscosity is introduced in the rarefaction wave. The Lax-Friedrichs flux and the two-shock Riemann solver (Equation (2.56)) with the MUSCL reconstruction (Figures 2.2-d and 2.2-e) are both able to reproduce the sharper shock without introducing any viscosity in the rarefaction wave, but the two-shock Riemann solver overpredict the water depth in the intial part of the rarefaction wave.
For this test case an error analysis is also performed calculating the norm of non-dimensional water depth error L_2:

$$
L_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{d_i - d_i^e}{d_i^e}\right)^2} \tag{2.58}
$$

where N is the total number of particles in the domain, d_i and d_i^e are the numerical and the reference water depth of ith particle.
In Table 2.1 the norm L_2 is shown for different stabilization terms, considering four intial particle interspacing: the MUSCL reconstruction is able to reduce the overall error using both the Lax-Friedrichs and the Two-Shock Riemann Solver in

comparison with other stabilization term where no reconstruction is done. The



(a)

(b)

(c)

(d)

(e)

Figure 2.2. Water depth for 1-D Dam Break with wet bed at time 50 s: (a) no viscosity, (b) artificial viscosity (c) Lax-Friedrichs flux (d) Lax-Friedrichs flux with MUSCL reconstruction (e) two-shock Riemann solver with MUSCL reconstruction

comparison between the different stabilization terms is made also against a 2-D cylindrical Dam Break with wet bed: the water is motionless at the beginning of the simulation and the water depth d is 10 m in the central part of the domain

Table 2.1. L_2 norm of non-dimensional water depth error calulated for 1-D dam break with wet bed at time 50 s and considering different stabilization terms: Artificial Viscosity (AV), Lax-Friedrichs (LF), Lax-Friedrichs with MUSCL reconstruction and Two-Shock Riemann solver (TS) with MUSCL reconstruction

dx	AV	LF	LF MUSCL	TS MUSCL
2.5 m	3.70E-003	4.30E-003	2.60E-003	2.70E-003
5 m	5.80E-003	6.40E-003	3.90E-003	4.10E-003
10 m	9.30E-003	9.80E-003	5.90E-003	6.00E-003
20 m	1.52E-002	1.57E-002	1.00E-002	1.02E-002

($|\mathbf{x} - \mathbf{x}_0| < 100m$) and 5 m outside this circle. The domain is discretized using 39,781 particles. The reference solution is obtained using a classical Eulerian Finite volume scheme.

Figure 2.3 shows the results using a Lax-Friedrichs flux whereas the results with the same flux but introducing the MUSCL reconstruction are plotted in Figure 2.4; finally figure 2.5 shows the results using the two-shock Riemann solver. For this test case the L_2 norm of the non-dimensional error of the water depth and the velocity components are calculated; the L_2 of the water depth is calculated according to Equation 2.58 whereas the same norm of the velocity component v_k is obtained from the following formulation:

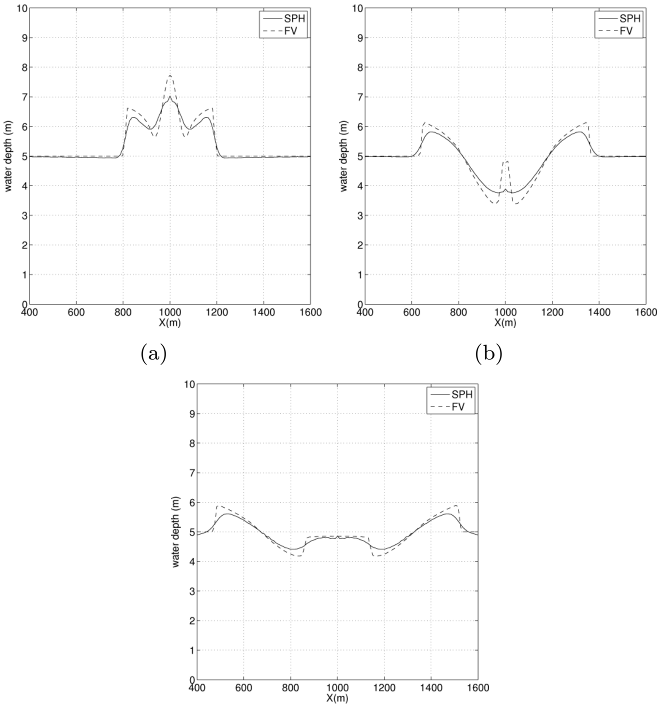
$$L_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{v_{i,k} - v_{i,k}^e}{\sqrt{g d_{i,k}^e}}\right)^2} \tag{2.59}$$

where $v_{i,k}^e$ is the k velocity component of the reference solution.

The L_2 norms of errors are plotted in Table 2.2 for different time steps; the results obtained using the MUSCL reconstruction (both with the Riemann solver and the Lax-Friedrichs flux) are more accurate than the ones obtained without this procedure even for this second test case.

2.8 Wetting - drying test case: Thacker basin

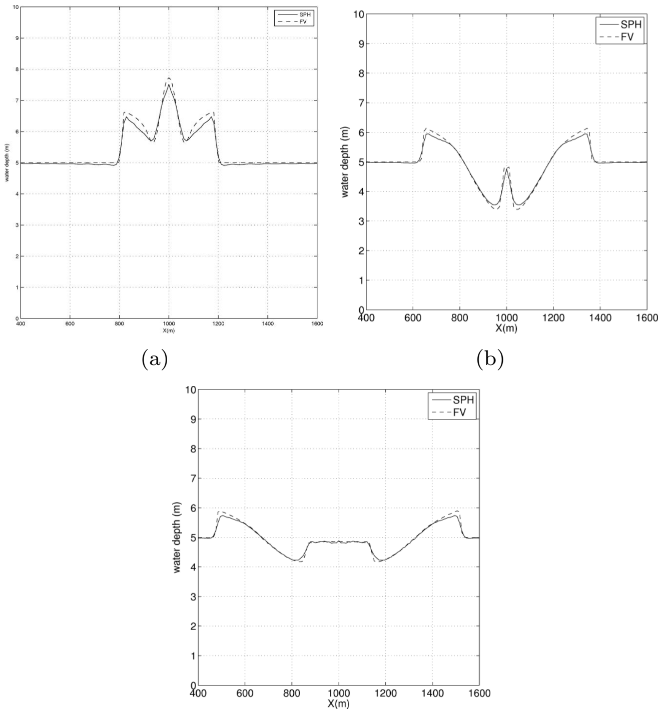
The capability of the proposed method of providing accurate results in the presence of 2D wetting and drying moving boundaries on non-flat topographies is an important test case to demonstrate the SPH-SWE is capable of simulating real cases. Here we can compare numerical results with the exact solutions given by Thacker [1981], which concerns the oscillation of a water volume in a frictionless



(a) (b)

(c)

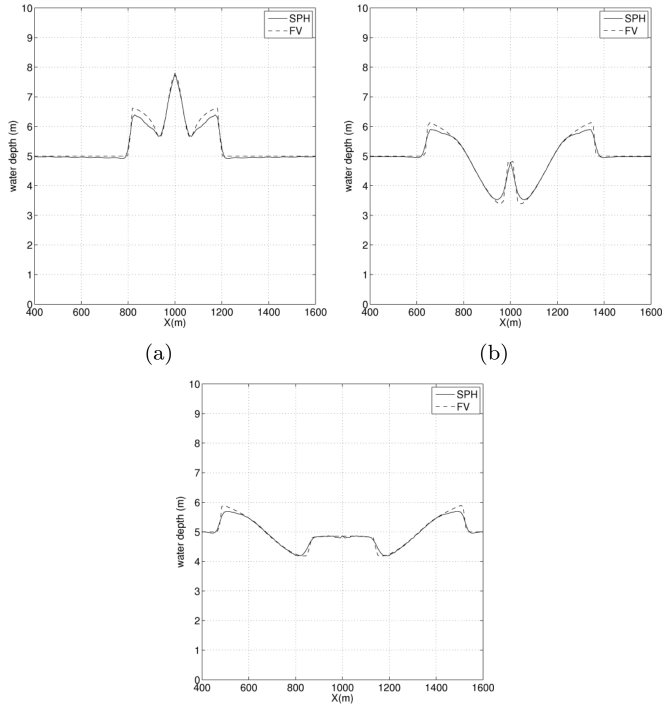
Figure 2.3. Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Lax-Friedrichs flux



(a) (b)

(c)

Figure 2.4. Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Lax-Friedrichs flux and MUSCL reconstruction



(a)

(b)

(c)

Figure 2.5. Circular dam break with wet bed: water depth in radial direction at time steps 10, 30, 50 s, with Riemann solver and MUSCL reconstruction

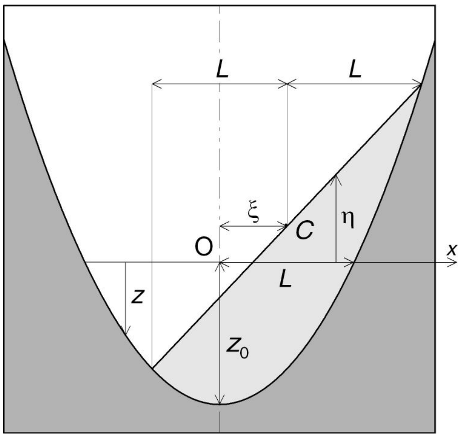


Figure 2.6. Definition sketch for the Thacker test with planar water surface

Table 2.2. L_2 non-dimensional norm of water depth, v_x and v_y error calulated 2-D cilindrical dam beak with wet bed at times 10, 30 and 50 s and considering different stabilization terms: Lax-Friedrichs (LF), Lax-Friedrichs with MUSCL reconstruction and Two-Shock Riemann solver (TS) with MUSCL reconstruction

Stabilization term	time(s)	$L_2(d)$	$L_2(v_x)$	$L_2(v_y)$
LF	10	0.018	0.012	0.012
LF with MUSCL	10	0.014	0.007	0.007
TS with MUSCL	10	0.014	0.007	0.007
LF	30	0.026	0.018	0.018
LF with MUSCL	30	0.015	0.009	0.009
TS with MUSCL	30	0.016	0.010	0.010
LF	50	0.029	0.020	0.020
LF with MUSCL	50	0.016	0.011	0.011
TS with MUSCL	50	0.017	0.012	0.012

paraboloid basin having equation:

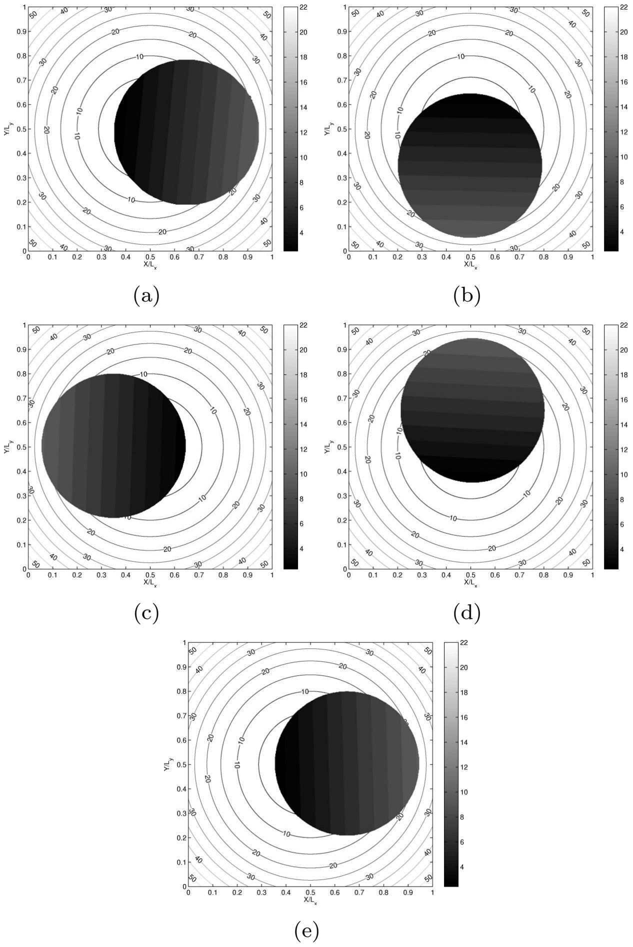
$$z = z_0 \left(1 - \frac{x^2 + y^2}{L^2}\right). \tag{2.60}$$

In Equation (2.60) the depth function z is positive below the equilibrium level, z_0 is the depth of the vertex of the paraboloid and L is the radius at $z = 0$. In the particular case considered the water body is initially planar and the velocity field is uniform; the analytical solution and the initial conditions are given by:

$$\begin{cases} \eta\left(x, y, t\right) = 2\xi \dfrac{z_0}{L} \left[\dfrac{x}{L} \cos \omega t - \dfrac{y}{L} \sin \omega t - \dfrac{\xi}{2L}\right] \\ u\left(x, y, t\right) = -\xi\omega \sin \omega t; \; v\left(x, y, t\right) = -\xi\omega \cos \omega t, \end{cases} \tag{2.61}$$

where η is the surface elevation, positive above the equilibrium level and $\omega = \sqrt{2gz_0}/L$ is the frequency of the rotation around the centre of the basin. The magnitude of the velocity vectors is constant over time at the value $|\mathbf{V}| = \xi\omega$, whereas the direction rotates over time.

The test was performed in a square domain $[0\,m \leq (x, y) \leq 10000\,m]$ with $z_0 = 10\,m$, $L = 3000\,m$ and $\xi = 1500\,m$; the basin dimensions are such that the water never reaches the boundaries. The numerical simulation was carried out for one periods $T = 2\pi/\omega$ with 11304 fluid particles. Figure 2.7 shows the contour maps of the water depth at different times: the shoreline is circular, the surface almost perfectly planar. The comparison between numerical and analytical water depth and velocities are plotted in Figures 2.8, 2.9 and 2.10 in the following three points: (5000; 5000), (5000; 6000), (5000; 70000). The first and the second point remain wet all the time whereas the third point gets wet and dry during the periodic motion, in all three point the numerical solution is able to reproduce the analytical solution.

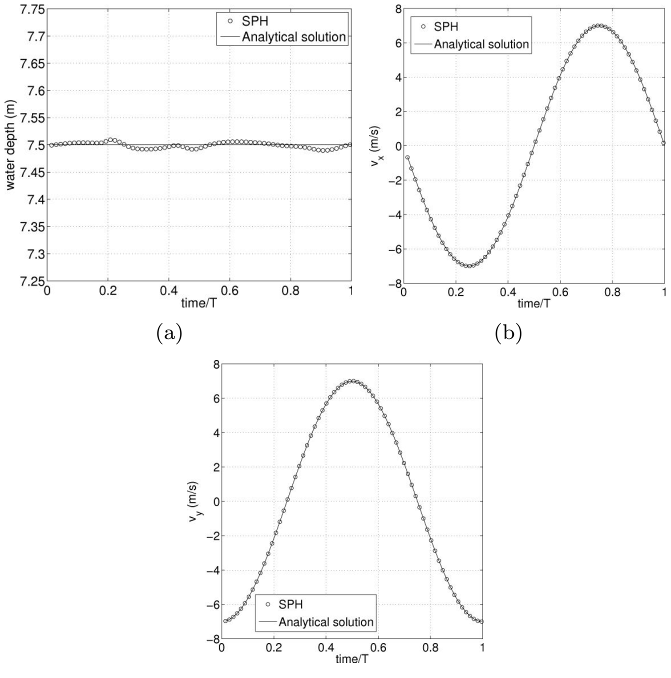


(a) (b) (c) (d) (e)

Figure 2.7. Thacker test case, water depth at non dimensional time steps t/T: (a) 0.015, (b) 0.253, (c) 0.505, (d) 0.742 and (e) 0.980

2.9 Concluding Remarks

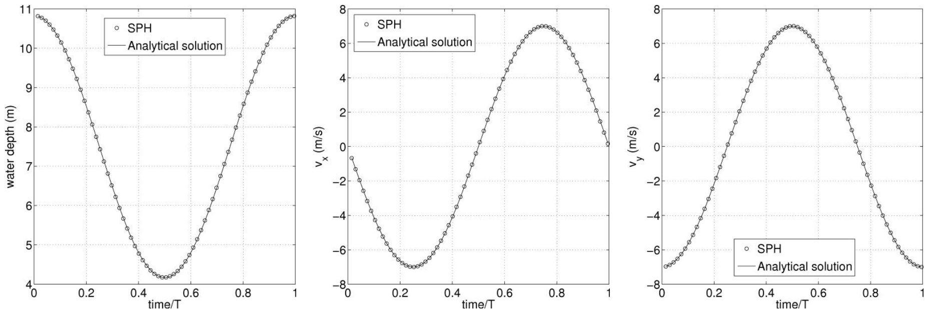
In this chapter an SPH method for numerical discretization of shallow water equations has been presented. The method presents some attractive features in comparison with classical Eulerian methods because no mesh and no special



(a) (b)

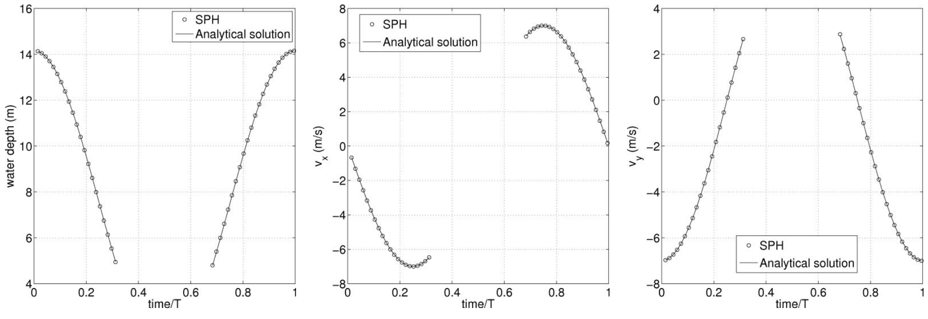
(c)

Figure 2.8. Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 5000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity



(a) (b) (c)

Figure 2.9. Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 6000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity



(a) (b) (c)

Figure 2.10. Thacker test case, comparison between analytical and numerical solution at point of coordinates (5000 m, 7000 m): (a) water depth, (b) v_x velocity and (c) v_y velocity

treatment of wet/dry interface are needed.

The capability of the method to deal with shock waves has been improved by removing artificial viscosity and considering particle interactions as a Riemann problem. Moreover non-upwind MUSCL reconstruction procedures have been introduced with the aim of obtaining a good representation of rarefaction waves. In order to extend the method to real case problems the slope and the friction source terms are calculated by means of a SPH interpolation method based on bottom particles which can be applied for any bathymetry.

The code has been tested against different 1-D and 2-D dam break problems showing satisfactory results. Thacker's 2-D analytical solution has been also simulated showing the capability of the method to reproduce accurate results with moving 2-D wetting and drying boundaries on non-flat topographies.

Chapter 3

Particle Splitting

3.1 The need for particle splitting

One of the problems in SPH-SWE code is lack of resolution when inundation of the fluid occurs over an initially dry bottom: whereas in Eulerian models the area of the cells is usually constant during the simulation, this is not valid for SPH models with variable smoothing length. The area A_i of i-th particle can be defined, using an SPH formalism, as

$$A_i = \frac{m_i}{\rho_i} \tag{3.1}$$

or, substituting Equation (2.1) into Equation (3.1), as

$$A_i = \frac{m_i}{d_i \rho_w} \tag{3.2}$$

since the mass of each particle is constant but the density ρ (or the water depth d) is not, A_i varies during the simulation: when a particle i is moving from a region with an initially high water depth to one with a shallow water A_i increases accordingly to Equation (3.2); this means that the resolution that is present in the region with high water is much bigger than the one with small water depth. In order to overcome this problem the following splitting criteria is introduced: when $A_i \geq \bar{A}$ where $\bar{A}$ is a fixed value, then the original particle is splitted into 7 daughter particles.

The capability of the 2D numerical model to simulate flow expansion is tested against a 1-D dam break with a dry bed in an infinite rectangular channel. This test case was previously simulated by Crespo et al. Crespo et al. [2007] by a fully 3-D SPH code. The channel is 1000 m wide and the water domain behind the dam is 1000 m long with an initial water depth of 10 m; the numerical results are compared with the analytical solution of the 1-D Ritter test case Ritter [1892]. In Figure (3.1) the particles position and the velocity field at time 20 s are plotted, whereas Figure (3.2) shows the water depth and the velocity v_x at the section y=500 m, at the same time. In this test case the particles remain at the same distance along the y direction but they drift apart in the x direction and therefore there is no interaction in x direction. This leads to an overestimation of the water depth and a consequent underestimation of the x-component of the velocity, as shown in Figure 3.2. The particle splitting procedure will be able to overcome this issue, as proven in section 3.4.2.

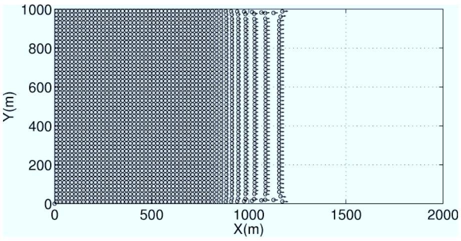
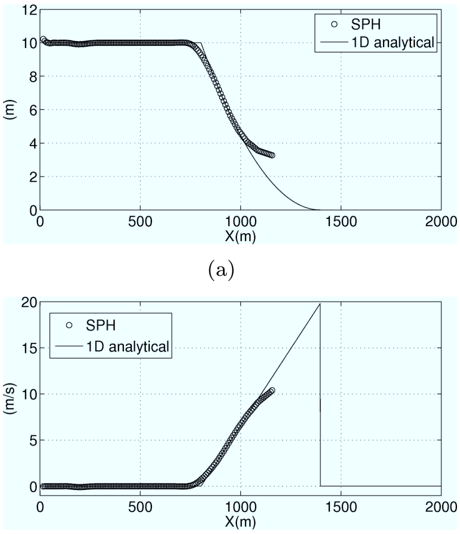


Figure 3.1. velocity field in the 1-D dam break in a rectangular channel over dry bed without splitting at time 20s

3.2 Splitting procedure

Once a splitting criteria is defined, then an appropriate procedure should be assessed: in 2D SWE Eulerian models mesh adaptivity is becoming more and more popular (Rogers et al. [2001]); there are also some attempts to introduce a splitting strategy in SPH models: Kitsionas and Whitworth [2002] have applied particle splitting to astrophysics problems, Lastiwka et al. [2005] applied an adaptive particle distribution to 1-D shock tube problem. Feldman and Bonet [2007] defined a dynamic splitting procedure for particles in an SPH model for Navier-Stokes equations which is conservative, and is suitable also for multidimensional domains but without considering the variable smoothing length. In this section we recall briefly the particle splitting procedure introduced in Feldman and Bonet [2007] and we adapt it to our SWE model with a new extension taking into account also the effects of variable smoothing length: the key idea is



(a)

(b)

Figure 3.2. 1-D Dam Break in a rectangular channel without splitting: (a) water depth and (b) v_x at the section y=500 m at time 20s

to define a splitting algorithm that is able to conserve both the mass (m) and the momentum ($m \cdot \mathbf{v}$) and minimize the error in the density and velocities fields.

If one particle is splitted into M daughter particles we have to define the mass

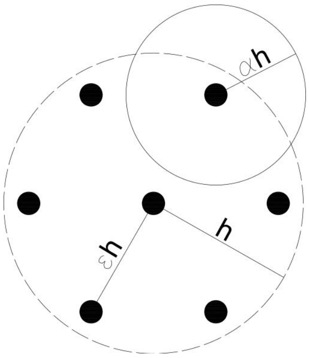


Figure 3.3. splitting patterns

m_k position $\mathbf{x}_k$, velocity $\mathbf{v}_k$, and the smoothing length h_k for any $k = 1 \ldots M$ refined particle, therefore the total number of degrees of freedom is 6 for each k-th daughter particle. In order to reduce the degrees of freedom of the problem the number of new particles M and their relative positions are defined by using

a fixed splitting pattern which identifies the relative particle position of daughter particles and their masses.
The hexagonal splitting pattern (plotted in Figure 3.3) is adopted in this work because it is a good balance between the total number of particles and the reduction of the density error (Feldman and Bonet [2007]) where the parameter ϵ defines the position of the new daughter particles placed on the vertices of the hexagon inscribed in a circle with radius equal to ϵh. The initial smoothing lengths of the daughter particles are defined as $h_k = \alpha h$ where α is another scalar parameter called smoothing ratio. The framework of this procedure is to fix arbitrarily the parameters ϵ and α and then to calculate the masses of the daughter particles m_k by minimizing the error between the refined and unrefined local density field. This local density $\rho(\mathbf{x})$ is computed using a scatter formulation Hernquist and Katz [1989]:

$$\rho\left(\mathbf{x}\right) = \sum_{j=1}^{N} m_j W_j\left(\mathbf{x}, h_j\right) \tag{3.3}$$

If the N-th particle is refined into M daughter particles then the density distribution changes because of the splitting as follows:

$$\rho^*\left(\mathbf{x}\right) = \sum_{j=1}^{N-1} m_j W_j\left(\mathbf{x}, h_j\right) + \sum_{k=1}^{M} m_k^* W_k\left(x, h_k\right) \tag{3.4}$$

where the m_k^* $k = 1 \dots M$ are the unknown.
the local splitting error e at generic point $\mathbf{x}$ is defined as:

$$e\left(\mathbf{x}\right) = \rho\left(\mathbf{x}\right) - \rho^*\left(\mathbf{x}\right) = m_N W_N\left(\mathbf{x}, h_N\right) - \sum_{k=1}^{M} m_k^* W_k\left(\mathbf{x}, h_k\right) \tag{3.5}$$

Note that the error of the density field is due only to the refined particles $k = 1 \dots M$. The global splitting error $\mathscr{E}$ is also defined as the integral over the whole domain of the local splitting error $e(\mathbf{x})$:

$$\mathscr{E} = \int_\Omega e\left(\mathbf{x}\right)^2 d\mathbf{x} \tag{3.6}$$

The unknown masses of the daughter particles can be rewritten as $m_k^* = \lambda_k m_N$ where the constraint $\sum_{k=1}^{M} \lambda_k = 1$ holds in order to conserve the total mass.

Rewriting also the total error terms of the unknown coefficients λ_k gives

$$\begin{aligned}\mathscr{E} = m_N^2 \Bigg[\int_\Omega W_N^2\left(\mathbf{x}, h_N\right) d\mathbf{x} - 2\sum_{k=1}^{M} \lambda_k \int_\Omega W_N\left(\mathbf{x}, h_N\right) W_k\left(\mathbf{x}, h_k\right) d\mathbf{x} \\ + \sum_{k,l=1}^{M} \lambda_k \lambda_l \int_\Omega W_k\left(\mathbf{x}, h_k\right) W_l\left(\mathbf{x}, h_l\right) d\mathbf{x} \Bigg]\end{aligned} \tag{3.7}$$

defining the following quantities:

$$\begin{aligned}\overline{C} &= \int_\Omega W_N^2\left(\mathbf{x}, h_N\right) d\mathbf{x} \\ \overline{b}_k &= \int_\Omega W_N\left(\mathbf{x}, h_N\right) W_k\left(\mathbf{x}, h_k\right) d\mathbf{x} \\ \overline{A}_{lk} &= \int_\Omega W_k\left(\mathbf{x}, h_k\right) W_l\left(\mathbf{x}, h_l\right) d\mathbf{x}\end{aligned}$$

it is possible to rewrite the global error as:

$$\mathscr{E} = \overline{C} - 2\boldsymbol{\lambda}^T \overline{\mathbf{b}} + \boldsymbol{\lambda}^T \overline{\mathbf{A}} \boldsymbol{\lambda} \tag{3.8}$$

where $\boldsymbol{\lambda}$ is the vector of λ coefficients.
Given a splitting pattern with splitting parameters α and ϵ the best vector of coefficients $\boldsymbol{\lambda}^*$ is calculated minimizing the global error $\mathscr{E}$:

$$\mathscr{E}^* = \min_{\boldsymbol{\lambda}} \left\{ \overline{C} - 2\boldsymbol{\lambda}^T \overline{\mathbf{b}} + \boldsymbol{\lambda}^T \overline{\mathbf{A}} \boldsymbol{\lambda} \right\} \tag{3.9}$$

with the constraint:

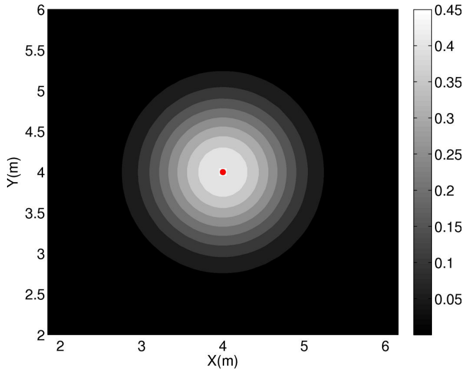
$$\sum_{j=1}^{M} \lambda_j = 1 \tag{3.10}$$

The splitting error depends on the choice of the splitting parameters α and ϵ and on the vector of coefficient $\boldsymbol{\lambda}$ and it is independent of the initial mass m_N and smoothing length h_N; therefore it is possible to define the optimal mass distribution $\boldsymbol{\lambda}^*$ before the beginning of the simulation solving a model problem of equations (3.9) and (3.10).
In order to asses the minimum value of global density error the model problem is solved using different values of α and ϵ. Using the classical cubic kernel (Li and Liu [2003]) the minimum value of $\mathscr{E}$ is obtained with $\alpha = 0.9$ and $\epsilon = 0.4$ (see

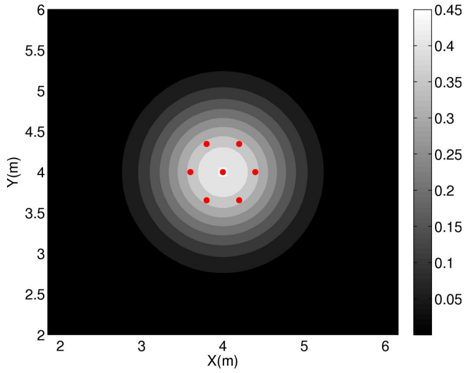
Table 3.1) and therefore this values are used in the SPH-SWEs numerical code. Figure 3.4 shows that the refined pattern with $\alpha = 0.9$ and $\epsilon = 0.4$ is able to reproduce the original kernel shape with a good level of approximation.

Table 3.1. results of particle split procedure: splitting parameters α and ϵ, global splitting error $\mathscr{E}$, relative optimal mass distribution for central particle λ_1 and for other particles $\lambda_{2...7}$

α	ϵ	$\mathscr{E}$	λ_1	$\lambda_{2...7}$
0.90	0.40	$1.76E-05$	0.1787	0.1369
0.90	0.50	$3.80E-05$	0.4476	0.0921
0.60	0.60	$2.41E-03$	0.1203	0.1466
0.80	0.60	$3.31E-05$	0.3239	0.1127



(a)



(b)

Figure 3.4. 2-D cubic kernel approximation using (a- unrefined and (b)-refined configuration

The velocities of the daughter particles of particle N are defined in order to conserve the total momentum, therefore the following constraints holds:

$$d_N \mathbf{v}_N A_N = \sum_{k=1}^{M} d_k \mathbf{v}_k A_k \tag{3.11}$$

where A_N is the area of the original particle and A_k are the areas of the daughter particles.
In addition to the constraint (3.11) we introduce the assumption that:

$$\mathbf{v}_k \cdot d_k = c_v d_N \mathbf{v}_N \tag{3.12}$$

Substituting Equation (3.12) into Equation (3.11) it is possible to obtain the following expression for c_v:

$$c_v = \frac{A_N}{\sum_{k=1}^{M} A_k} \tag{3.13}$$

and the final expression of the velocities of the daughter particles is:

$$\mathbf{v}_k = c_v \frac{d_N}{d_k} \mathbf{v}_N \tag{3.14}$$

3.3 SWE-SPH code with splitting procedure

We recall that the mass of daughter particles are defined through the procedure described in section 3.2 before the beginning of the simulation: the best values of λ_j (with $j = 1 \dots M$) are calculated solving the model problem based on the scatter evaluation of the density (Equation 3.3). Conversely the density ρ is calculated in the SPH-SWE, by the gather formulation (2.4) (see Hernquist and Katz [1989] for details about the difference of the two formulations).

3.3.1 Theoretical test case

In order to test the error introduced by the splitting in the density field a simple numerical test has been set up: ρ is calculated in a square domain where the particles in the centre of the domain are refined (see Figure 3.5), the originally unrefined particles are equispaced and positioned in a square Cartesian grid. The smoothing length h of unrefined particles is taken as $h = 1.2dx$ where $dx = 50$ m is the distance between particles; finally the mass of each unrefined particle is

$m = 25 \cdot 10^6$ kg. The exact value of the density is $\rho_{ex} = 1 \cdot 10^4$ kg/m^3. Figure 3.6 shows the comparison of ρ calculated using (3.3) and (2.4): the maximum error in the density field calculated using equation (3.3) is equal to 0.3% whereas, using the Equation (2.4) it increases to 5%, this is due to the fact that the optimal mass distribution of daughter particles is calculated starting from the gather formulation.

If scatter formulation is used then ρ_i depends not only to h_i but also to the $h_1, h_2, \ldots, h_N$ where N is the number of particles in the neighbourhood of particle i and a new iterative implicit procedure has to be used to update the density and the smoothing length.

If we wish to know the density ρ_i^k and the smoothing length h_i^k at the k-th iterative loop for every i-th particle in the domain, then the density and the non-dimensional residual Ψ_i^{k+1} at the next iterative loop $k+1$ can be calculated using (3.3):

$$\rho_i^{k+1} = \sum_{j=1}^{N} m_j W_j \left(\mathbf{x}_i - \mathbf{x}_j, h_j^k\right)$$

also the non-dimensional residual Ψ_i^{k+1} is calculated at $k+1$ iterative loop as:

$$\Psi_i^{k+1} = \frac{|\rho_i^{k+1} - \rho_i^k|}{\rho_i^{k+1}} \tag{3.15}$$

Once the densities and the non dimensional residuals are calculated for every particle in the domain, then the smoothing lengths are updated using Equation (5.21):

$$h_i^{k+1} = h_0 \left(\frac{\rho_0}{\rho_i^{k+1}}\right)^{1/d_m}$$

Finally the L_2 norm of the residual is calculated as:

$$L_2(\Psi_i^{k+1}) = \sum_{i=1}^{N} \sqrt{\frac{\left(\Psi_i^{k+1}\right)^2}{N}} \tag{3.16}$$

The iterative process is stopped when $L_2(\Psi_i^{k+1}) < \varepsilon_\Psi$. As already stated in section 2.2.1, the coefficient ε affects the speed of the code because it modify the number of iteration necessary in each time step: in the numerical tests presented

in this work a value of $\varepsilon = 1 \cdot 10^{-3}$ is used.
The iteration procedure starts from the initial guess ρ_i^0 calculated using Equation (2.11), this reduces the number of iterations in every time step in comparison with using the density calculated at the previous time step.

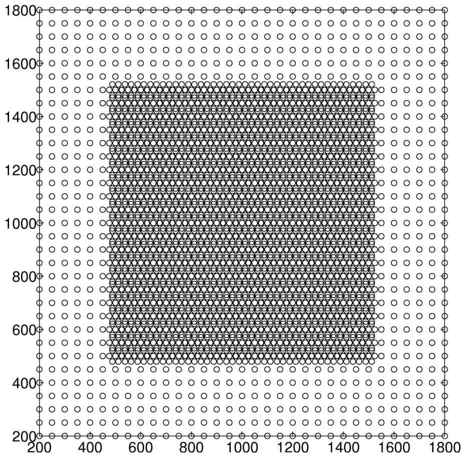
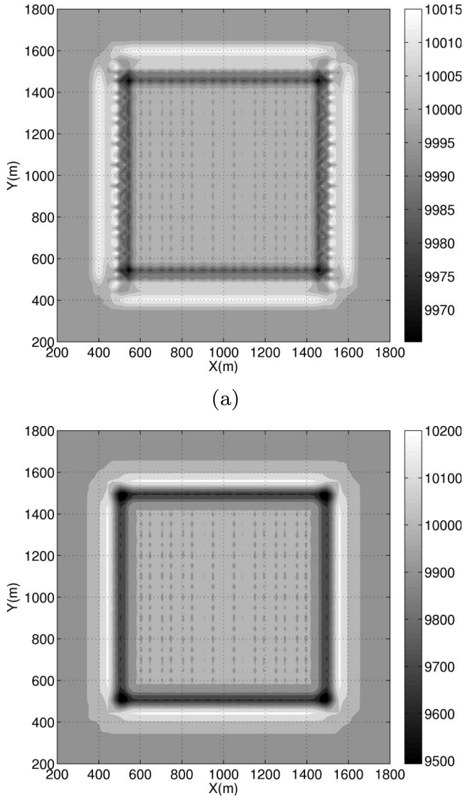


Figure 3.5. particle position for the numerical test of density calculation in presence of refined particles

3.4 Test Cases

3.4.1 Circular Dam Break over dry bed

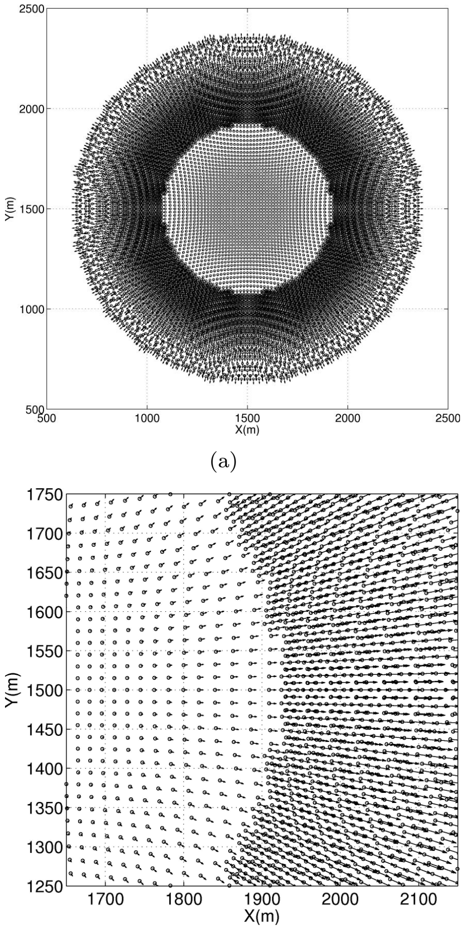
In the first test case the evolution of a circular dam break over a dry bed is analysed. A cylindrical dam with radius 1000 m is considered and the domain is discretized using 5000 particles at the begin of the simulation; the reference solution is obtained using a classical finite volume Eulerian code. The test is run with and without the splitting procedure, the value $\bar{A}$ in this test case is taken equal to 2 in the whole domain. In Figure 3.7 the particle position is plotted after 30 s of simulation when the splitting is activated: the radial symmetry of the solution is preserved. In Figure 3.8 a comparison of the water depth along radial direction at different times 10, 30 and 50 s is made, the two numerical solutions obtained with and without the splitting are in good agreement with the reference solution. This means that the error introduced by the splitting procedure in the water depth is negligible thanks to the procedure described in section 3.2.



(a)

(b)

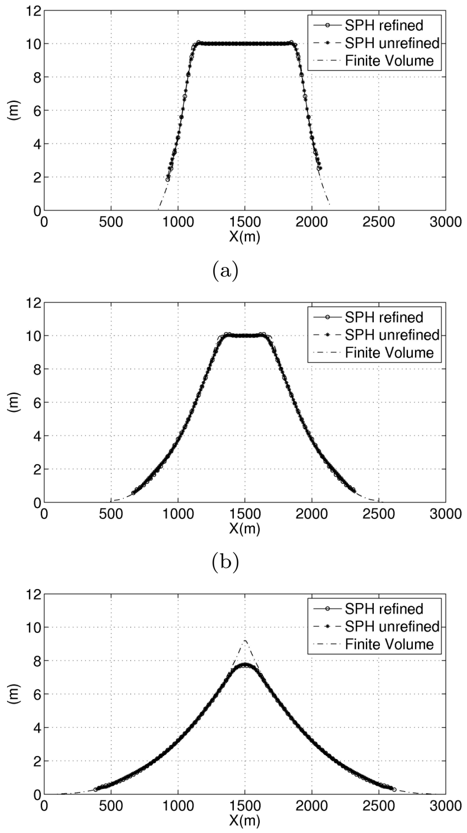
Figure 3.6. Density calculation with refined particle in the central part of the square domain (a) using scatter (Equation (3.3)) and (b) gather (Equation (2.4)) formulation



(a)

(b)

Figure 3.7. velocity field in the Circular Dam Break test case over dry bed with splitting at time 30s. (a) the whole domain (b) zoom close to the splitting interface



(a)

(b)

(c)

Figure 3.8. Circular dam break with dry bed: water depth in radial direction at time (a)-10 s, (b)-30 s and (c)-50 s

3.4.2 1D-2D dam break over dry bed

The same 1D-2D dam break test case presented in section 3.1 has been repeated in this section using the particle splitting. We showed before that the anisotropy of particle distribution causes a remarkable underestimation of the water depth close to the wet-dry front. de Leffe et al. [2008] solved this problem introducing an anisotropic kernel together with a periodic remeshing procedure, however it is well known that the remeshing introduces some not negligible numerical viscosity (Fang and Parriaux [2008]), moreover the free surface should be detected before re-assigning the particles position and therefore the capability of the SPH method to naturally deal with interfaces (which is one of the most attractive aspect of the method) is lost. We simulate this test case without introducing any corrections but refining the particles in the rarefaction wave as shown in Figure 3.9, the threshold value of area $\bar{A}$ (see section 3.1) is taken equal to 1.5 in this test case. With this approach the interaction between particles in the x-direction is kept without any remeshing. Different stabilization terms have also been considered: the Lax-Friedrichs flux together with the MUSCL reconstruction (Figures 3.10 and 3.11) is able to reproduces the best results, whereas (as already noticed in section 2.7) using no MUSCL reconstruction leads to a less accurate reproduction of the rarefaction wave (see Figures 3.12 and 3.13). Finally if the Two-Shock Riemann solver is applied some numerical fluctuations in the velocity are present. Regardless of the kind of stabilization term used with the splitting procedure the 1D-2D dam break over dry bed can be simulated in a satisfactory way.

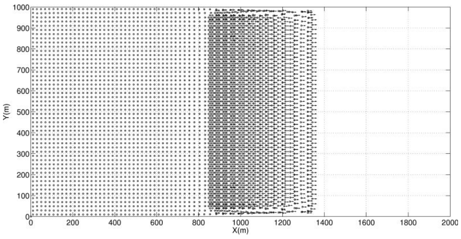
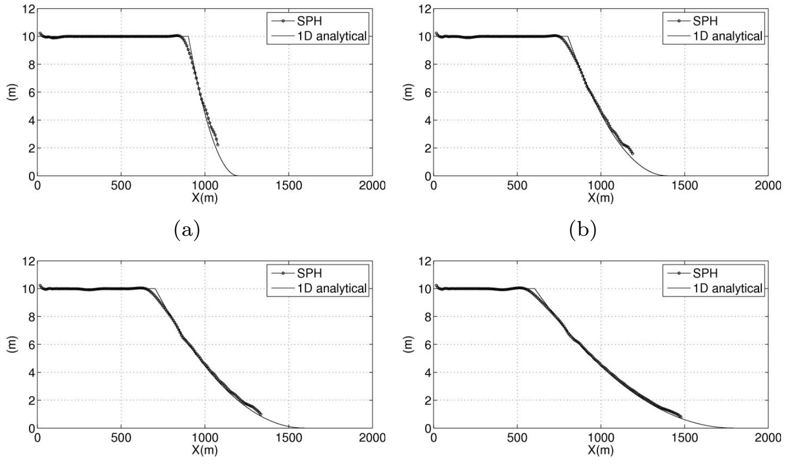


Figure 3.9. 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term with MUSCL reconstruction: velocity field in the 2-D dam break in a rectangular channel over dry bed at time 30s

3.4.3 CADAM test case with a 45° bend

The European Concerted Action on DAm break Modelling (CADAM) made an experiment where a dam break flow occurs along an initially dry channel (Soares



(a) (b)

(c) (d)

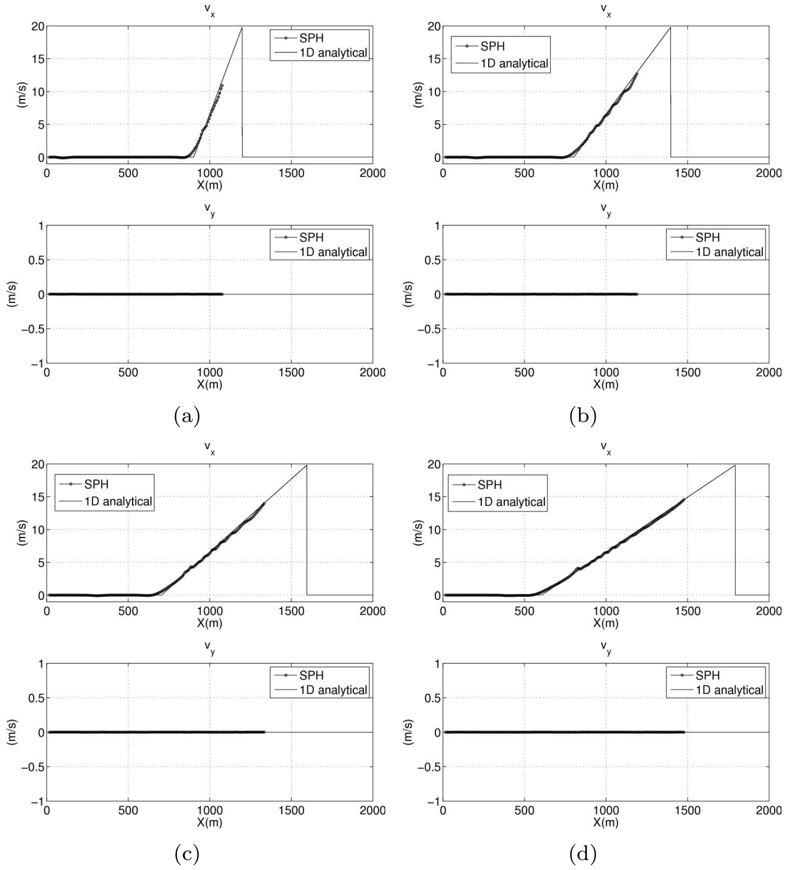
Figure 3.10. 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term with MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s

Frazão et al. [1998]). That channel has a rectangular cross section of 0.495 m, is connected upstream with a square reservoir and is 8.4 m long. A 45° bend is located after 4.25 m, the bottom is flat and is 0.33 m higher than the one of the reservoir, as plotted in Figure 3.16. The water levels are registered during the experiment in the reservoir and along the channel using 9 gauges. At the begin of the simulation the gate is suddenly removed and the water starts to flow from the reservoir in the channel and it reaches the bend after roughly 3 s; then a bore forms and starts to travel back to the reservoir, after 20s the bore reaches the reservoir and it disappears. This case has been used extensively by other SWE researchers (Zhou et al. [2004], Loukili and Soulaïmani [2007]) for benchmarking since the case has a variety of difficult aspects.

This first simulation is performed using no splitting and 9603 particles are initially positioned in the reservoir, the manning coefficient of the friction source term is taken as $n = 0.01sm^{-1/3}$ (see section 2.4).

The step located between the bottom of the reservoir and the channel has been discretized using the SPH interpolation method of bottom particles described in session 2.3. The only expedient introduced here is a smoothing length for the bottom particles which is two times the smoothing length initially assigned to the fluid particles.

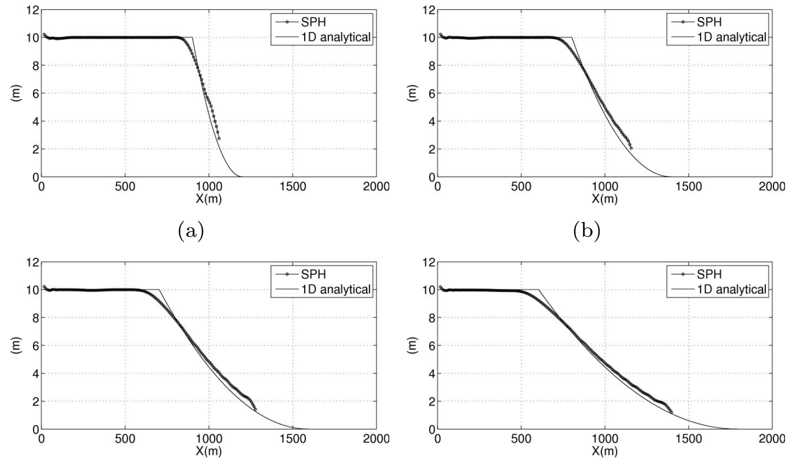
Figure 3.17 shows the comparison between the experimental and numerical water



(a) (b)

(c) (d)

Figure 3.11. 1-D Dam Break in a rectangular channel with splitting; Lax-Friedrichs stabilization term with MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s

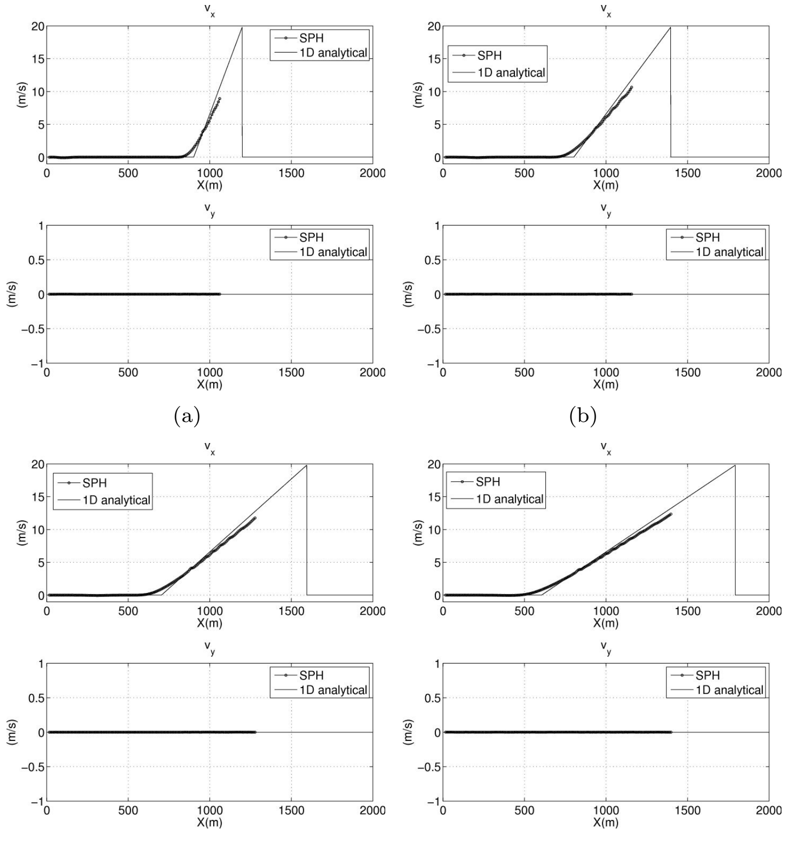


(a) (b)

(c) (d)

Figure 3.12. 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term without MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s

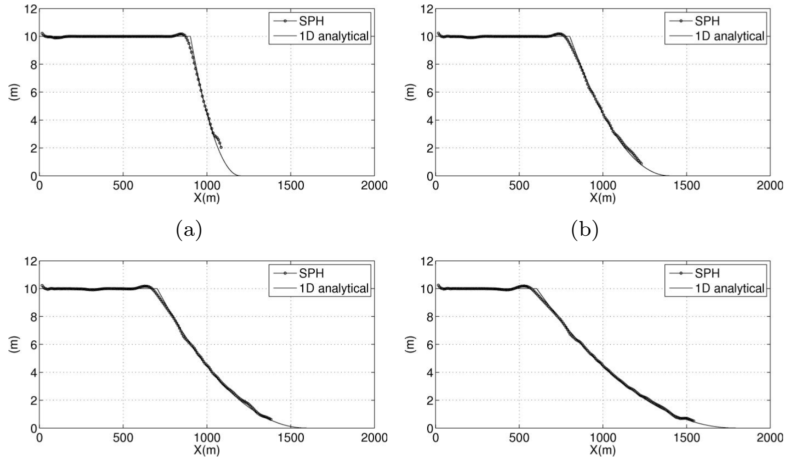
levels: gauge 1 is placed inside the reservoir near the channel; the good agreement of registered data with the numerical results means that the discharge that is entering in the channel is correct. Gauges 2, 3 and 4 are placed along the channel upstream the bend, therefore they registered the abrupt water level elevation due to the reflected wave that is travelling upstream to the reservoir. The numerical model is able to reproduce the water level at gauges 3 and 4 and at gauge 2 after 20 s whereas there is a difference with the experimental data of gauge 2 in the first half of the experiment, this difference is presents also in the results obtained by other authors (Aureli et al. [2004], Vázquez-Cendón [1998]) and it can be explained by the strong gradient of the water surface that is present close to gauge 2 during the first stages of the simulation: in this part of the domain the hydrostatic assumption is probably a rough approximation. Gauges 5, 6 and 7 are placed in the bend and the numerical model is able to reproduce the registered water level in this tree gauges, in particular the surface inclination in the bend is correctly simulated. Gauges 8 and 9 are placed downstream of the bend and the overall comparison of the water level is satisfactory although the numerical model slightly underpredicts the water level at the gauge 9. Figure 3.18 shows the water elevation field at different times 5, 10 and 20 s, in the first time step the bore formation near the bend can be seen, in the second one the bore is travelling upstream and in the last one it disappears.



(a) (b)

(c) (d)

Figure 3.13. 1-D Dam Break in a rectangular channel with splitting, Lax-Friedrichs stabilization term without MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s



(a)

(b)

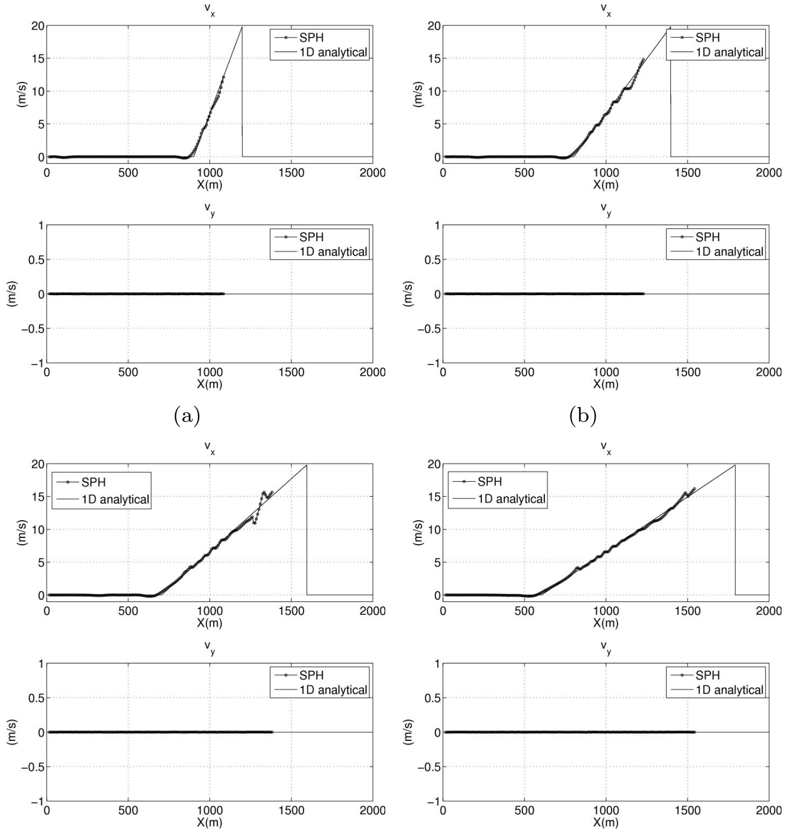
(c)

(d)

Figure 3.14. 1-D Dam Break in a rectangular channel with splitting, Two-Schock Riemann solver with no MUSCL reconstruction: water depth at the section y=500 at times (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s

In order to reduce the computational time the same test case has been simulated using bigger particles inside the reservoir and splitting them when they approach the channel. In this second simulation 2450 particles are initially placed in the reservoir and the same manning coefficient is applyed. Despite of the reduced resolution the results of this second simulation are analogous to the results obtained in the first one where much more particle are used (see Figure 3.19), this is due to the splitting procedure adopted that increase the resolution just in the part of the domain where the strong modifications in the water depth and in the velocity field occur. The computational time of the simulation with no splitting is 147 minutes, whereas it is equal to 86 minutes in the simulation with bigger particles and splitting procedure: therefore with the splitting procedure the computational time is reduced of 40%.

Figures 3.20 and 3.21 show the comparison between the water levels obtained by Zhou et al. [2004] using a Finite Volume algorithm and by the SPH-SWEs presented in this work. At gauges 4 and 6 the SPH-SWEs results are in good agreement with the experimental data whereas FV code overestimates them, whereas at gauges 9 the experimental data are slightly underestimates by the SPH method and slightly overestimates by the FV code.



(a) (b)

(c) (d)

Figure 3.15. 1-D Dam Break in a rectangular channel with splitting, Two-Schock Riemann solver with MUSCL reconstruction: velocities y=500 at time steps (a)-10 s, (b)-20 s, (c)-30 s and (d)-40 s

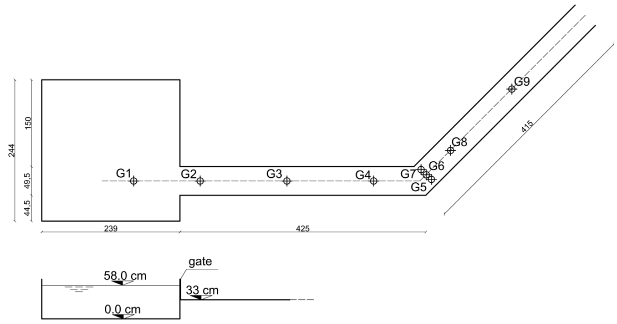

G1
G2
G3
G4
G5
G6
G7
G8
G9
244
150
49.5
44.5
239
425
415
gate
58.0 cm
33 cm
0.0 cm

Figure 3.16. CADAM test case: plane and profile view of the experimental setup

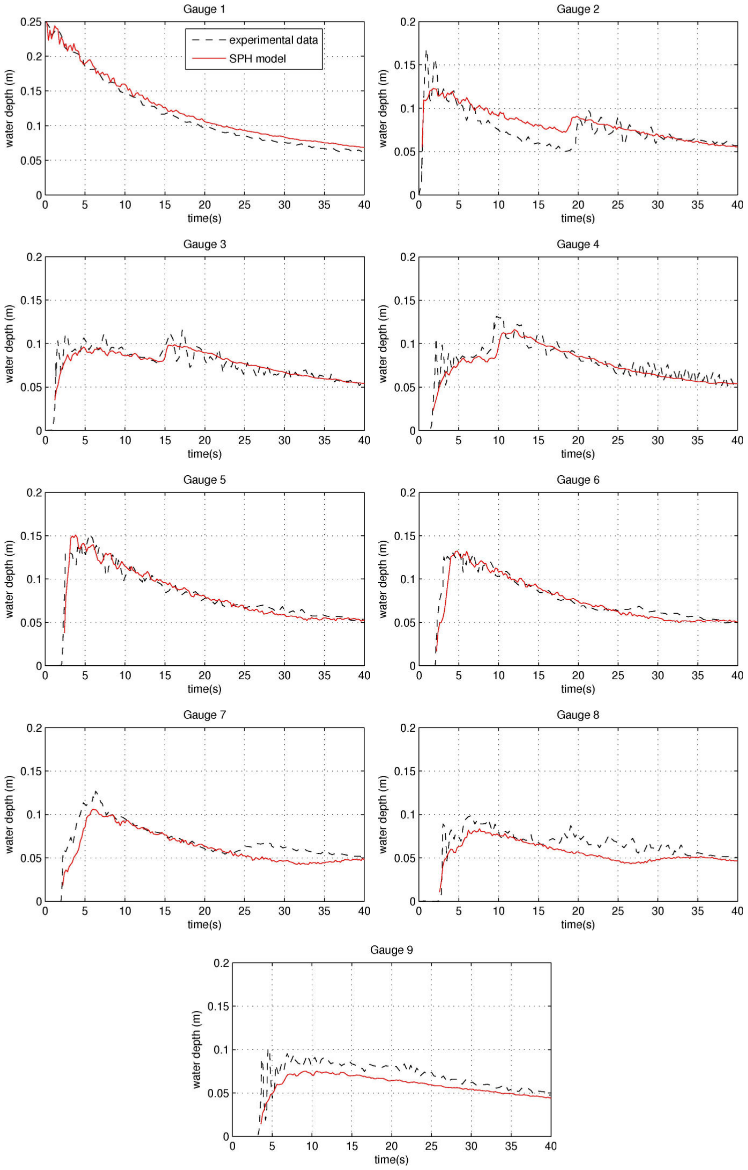


Figure 3.17. CADAM test case with no splitting: water level registered at different gauges

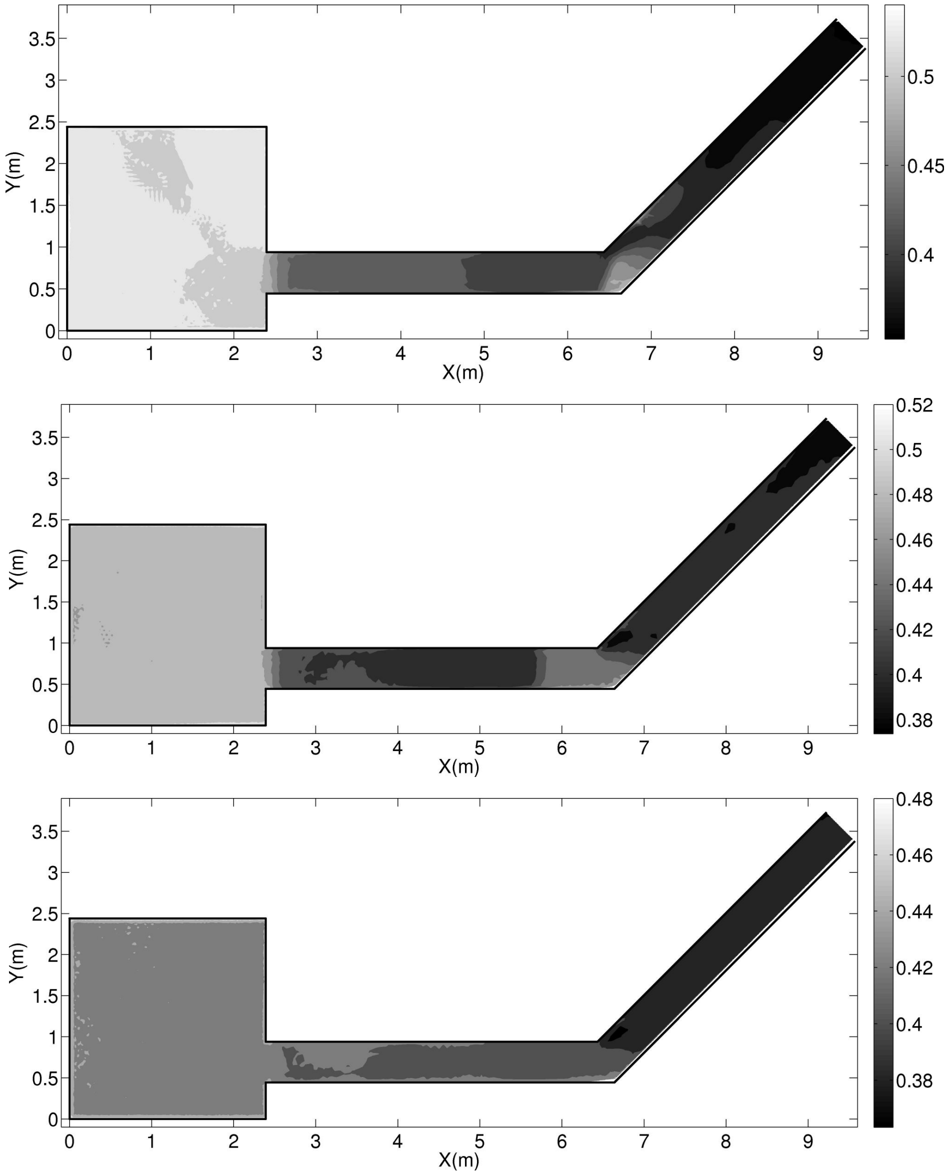


Figure 3.18. CADAM test case with no splitting: comparison between FV and SPH-SWEs models at gauges (a) 4 (b) 6 and (c) 9

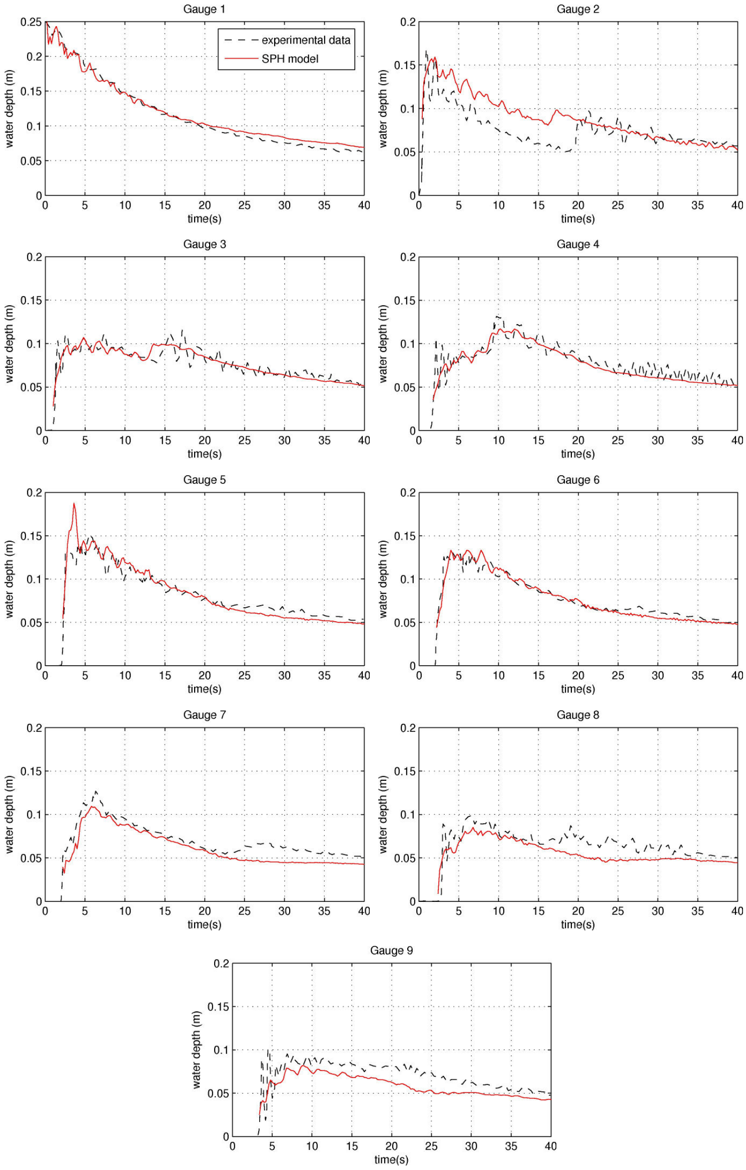


Figure 3.19. CADAM test case with splitting:water level registered at different gauges

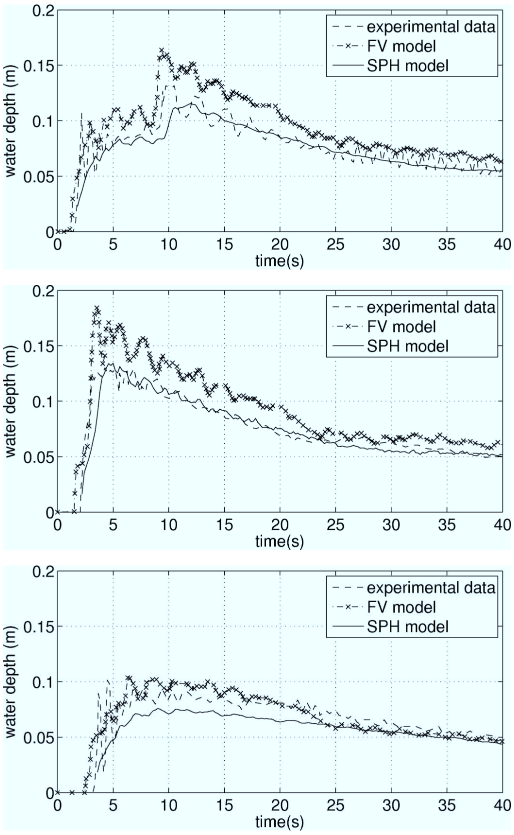


Figure 3.20. CADAM test case with splitting: comparison between FV and SPH-SWEs models at gauges (a) 4, (b) 6 and (c) 9

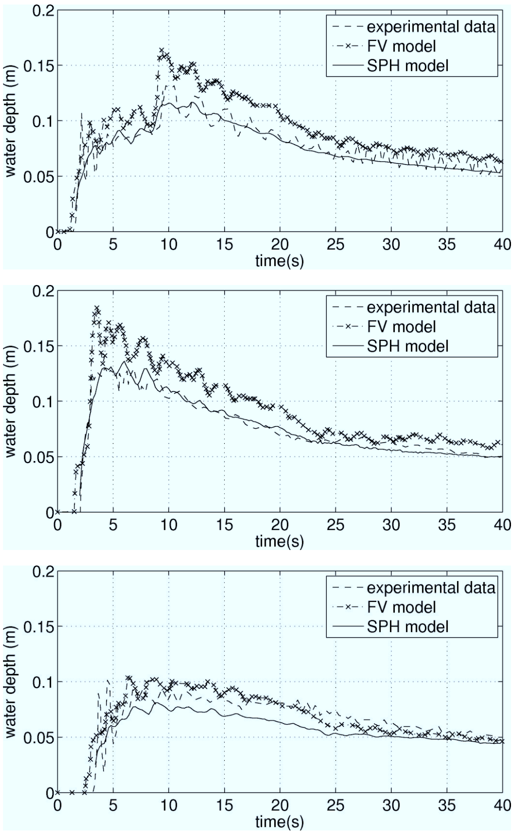


Figure 3.21. CADAM test case with splitting: comparison between FV and SPH-SWEs models at gauges (a) 4, (b) 6 and (c) 9

3.5 Concluding Remarks

The main limitation of the SPH-SWEs numerical scheme is the lack of resolution in zones with a reduced water depth; this has been overcome in this paper by introducing a particle splitting procedure: if one particle has an area which is more than a fixed value it is divided into seven daughter particles. The masses, velocities and water depth of daughter particles are assigned by conserving the both the mass and momentum. A new procedure has been proposed that significantly reduces the error due to particle splitting when using variable smoothing lengths.

The numerical code has been tested against the CADAM test case where a dam break flow occurs along an initially dry channel; the step located at the inflow and the bend positioned along the channel creates a quite complicated geometry representative of a real test case; nevertheless the numerical model is able to simulate the phenomenon in a satisfactory way both with and without the splitting procedure activated, nevertheless it remarkably reduces the computational time.

Chapter 4

Development of open boundaries for SPH-SWEs and Application to real problems

4.1 Characteristic boundary method

The one-dimensional SWEs written in quasi-linear form are:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x} = \mathbf{S} \tag{4.1}$$

where:

$$\mathbf{A}(\mathbf{W}) = \begin{bmatrix} v & d \\ g & v \end{bmatrix} \mathbf{W} = \begin{bmatrix} d \\ v \end{bmatrix}; \mathbf{S} = \begin{bmatrix} 0 \\ -g\frac{\partial b}{\partial x} + \mathbf{S}_f \end{bmatrix}$$

where v is the 1D velocity, d is the water depth and b is the bottom elevation and $\mathbf{S}_f$ is the friction source term.
the matrix $\mathbf{A}(\mathbf{W})$ has got two distinct and real eigenvalues: $v - c$ and $v + c$ where c is the wave propagation speed. These two eigenvalues are the characteristic directions of the SWEs and along these directions the fluid information are transported as a combination of primitive variables v and d. Therefore only vari-

ables transported from the boundaries in the domain can be imposed as physical boundary conditions, whereas all other variables (which are transported from the domain to the boundary) should be calculated and not imposed. Along the characteristic directions the Riemann invariants ($v \pm 2c$) remain constant if no source terms are present, therefore imposing the steadiness of these values represents the most theoretically correct way to calculate the missing informations.
In other areas of CFD, the method of characteristics has been used extensively to model both solid and transmissive boundaries (e.g. Giles [1990]). Nevertheless this method is difficult to apply because the characteristic lines are not known a priori. In Finite Volume methods many authors calculate the characteristic directions just to define the number of variables that should be prescribed at the boundary then the method of ghost cells is used and the primitive or conserved variables are directly imposed in the ghost cells Toro [1997]. For example in a supercritical flow ($u > c$) in 1D domain both the v and d have to be imposed at the inflow because both the characteristic directions are entering in the domain, whereas in a outflow supercritical boundary condition both v and d have to be calculated. In a subcritical flow both at the inflow and at outflow just one of the two characteristic lines points towards the domain and therefore just one of the two variables have to be imposed and the other one should be calculated.
Herein, for the SPH-SWEs numerical code the approach of Fujihara and Borthwick [2000] is applied. In this approach at the open boundaries the Riemann invariants are imposed according to the local Froude number as follow:

1. subcritical outflow condition (the water depth d is imposed):

$$\begin{aligned} v_{b,n} &= v_{i,n} + 2\sqrt{g}\left(\sqrt{d_i} - \sqrt{d_b}\right) \\ v_{b,t} &= v_{i,t} \end{aligned} \tag{4.2}$$

   where $v_{b,n}, v_{b,t}$ and d_b are the velocities and the water depth calculated at the boundary, whereas the subscript i indicates the inner Riemann state value.

2. subcritical inflow condition (velocity $\mathbf{v}_b$ is imposed):

$$d_b = \left[\frac{1}{2\sqrt{g}}\left(v_{i,n} - v_{b,n}\right) + \sqrt{d_i}\right]^2 \tag{4.3}$$

3. supercritical outflow condition:

$$v_{b,n} = v_{i,n};\ v_{b,t} = v_{i,t};\ d_b = d_i \tag{4.4}$$

4. supercritical inflow condition: both the velocity $\mathbf{v}_b$ and d_b have to be imposed.

In this way characteristic Riemann invariants are imposed at the boundaries using a simplified 1D approach and no bicharacteristics lines are used. Moreover the Riemann invariants are not imposed along the characteristic lines but an SPH interpolation is used in order to transfer the information from the fluid to the boundaries. These represent two approximations of the rigorous characteristic boundary method, but in this way the characteristic boundary method can be applied in a meshless Lagrangian code without overcomplicating the algorithm. The only limitation that this approach has is that the fluid near the boundaries should be unidirectional, so that one-dimensional approximation can be still considered valid.
In the next session the procedure to calculate the inner Riemann state of velocities and water depth is explained.

4.2 Open Boundaries in SPH

Open boundaries in meshless Lagrangian numerical method for Computational Fluids Dynamics have received only small attention so far; nevertheless the imposition of Open Boundaries represents a crucial element in any Shallow Water numerical model because it allows the computational domain to be limited to the area of interest: phenomena like flood waves in rivers and tides in estuaries can be described only if reliable open boundaries are available. In Eulerian models the imposition of inflow and outflow Boundary Conditions is relatively straightforward because each cell of the mesh describes a part of the domain with inflow and outflow fluids and ghost cells can be used to apply open BC. In SPH methods the particles move during the simulation therefore they have to be conveniently inserted in and removed from the domain and this, together with the approximate nature of the SPH interpolation makes the implementation of this kind of boundary rather difficult. Lastiwka and Quinland [2008] introduced the permeable non reflecting boundary condition in a gas dynamic SPH numerical model; Ramos-Becerra et al. [2009] built an Incompressible SPH numerical model and they showed some application with an inlet and outlet boundary condition.

de Leffe et al. [2009] developed a Finite Volume with Characteristic Flux Scheme for open boundaries in the framework of SPH method proposed by Vila [2000] for integration of Navier-Stokes equations.
In this chapter a method to handle both subcritical and supercritical Open Boundary Conditions in an SPH numerical code for Shallow Water Equations is presented.

4.2.1 SPH interpolation at Open Boundary particles

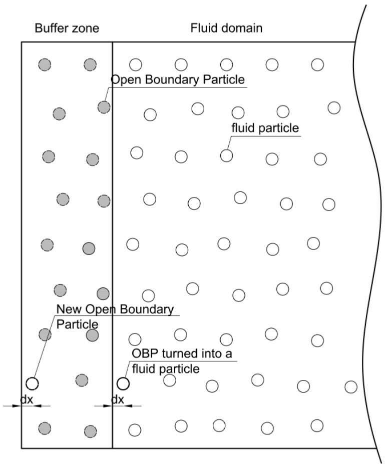

Buffer zone
Fluid domain
Open Boundary Particle
fluid particle
New Open Boundary Particle
OBP turned into a fluid particle
dx
dx

Figure 4.1. boundary at the inflow zone: boundary and fluid particle management

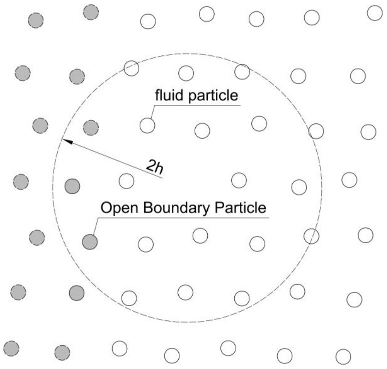

fluid particle
2h
Open Boundary Particle

Figure 4.2. boundary at the inflow zone: boundary and fluid particle management

In order to define an open boundary a buffer zone has to be defined and filled with Open Boundary Particles; as shown in Figure 4.1 this buffer zone is located just outside the fluid domain. These Open Boundary Particles are added in the SPH summation to update the acceleration and the water depth of the fluid particles as shown in Figure 4.2.

In an inflow subcritical BC the particle velocity of the OBP normal to the boundary $\mathbf{v}_b$ is imposed and the water depth d_b is calculated by means of Equation 4.3; whereas in an inflow supercritical BC both $\mathbf{v}_b$ and d_b are imposed. In an outflow BC if the flow is subcritical the velocities are calculated using Equation 4.2 and d_b is imposed. Finally for outflow supercritical BC both the particle velocities and water depth are calculated using Equation 4.4.

The Open Boundary Particles interact with the fluid particles, the water depths and the velocities of these particles are updated accordingly to the characteristic boundary method as explained in section (4.1). The inner Riemann state values of velocity $\mathbf{v}_{i,o}$ and water depth $d_{i,o}$ are calculated for each Open Boundary Particle by means of an SPH interpolation conducted using just the fluid particles:

$$\begin{aligned} \mathbf{v}_i^o &= \textstyle\sum_j \mathbf{v}_j^f \bar{W}_i(\mathbf{x}_i^o - \mathbf{x}_j^f, h_o)\frac{m_j^f}{\rho_j^f} \\ d_i^o &= \textstyle\sum_j d_j^f \bar{W}_i(\mathbf{x}_i^o - \mathbf{x}_j^f, h_o)\frac{m_j^f}{\rho_j^f} \end{aligned} \tag{4.5}$$

where the superscript o indicates the open boundary particles and the fluid particles are referred with the superscript f for the sake of clarity, $\bar{W}_i$ is the kernel corrected using the Shepard filter (Randles and Libersky [1996]). This filter is used because the kernel of the open boundary particles is not completely filled and therefore the classical SPH interpolation would generate too big interpolation errors.

Since the fluid particles are entering in the buffer zone of the open boundaries, and the Open Boundaries Particles are entering in the fluid domain a particle management algorithm has to be used: each time an Open Boundary Particle enters in the fluid domain it is transformed in a fluid particles and a new Open Boundary Particle is inserted in the upstream part of the buffer zone (Lastiwka and Quinland [2008]) as shown in Figure 4.1. This procedure keeps the buffer zone always filled with OBP and avoids any kernel truncation error for the fluid particle. When a fluid particle is entering in the buffer zone the procedure is more simple: it is transformed into a new Open Boundary Particle. Finally Open Boundary Particles that are leaving the buffer zone are deleted and no longer used in calculations.

4.3 1-D Steady flow over a bump

These set of benchmark tests concern steady flow along a 10 m long one-dimensional, frictionless channel with a bottom characterized by the presence of a bump:

$$b(x) = \begin{cases} b_0 \left[1 - \frac{(x-5)^2}{4}\right] & \text{if } 3\text{m} < x < 7\text{m} \\ 0 & \text{elsewhere} \end{cases} \tag{4.6}$$

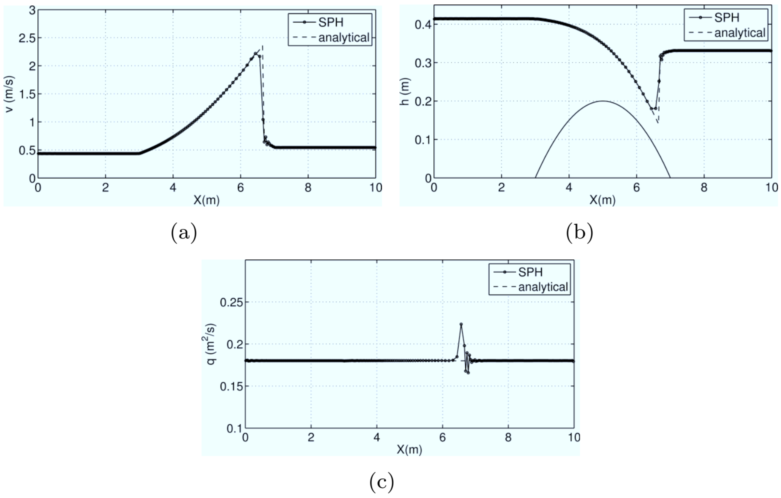
where x is the distance along the channel, b_0=0.2 m. At the beginning of the simulation the domain is dicretized using a particle spacing equal to 0.05 m, the Courant number C_{CFL} =0.4. The Boundary Conditions determine the flow conditions which can be subcritical, transcritical with or without a shock, or supercritical (Aureli et al. [2008]), an analytical solution is available for this test (Goutal and Maurel [1997]). In order to test the capability of the numerical model to reproduce different kind of open BCs all the different possibilities have been simulated: In test (i) subcritical inflow and outflow are imposed and a transcritical flow with a shock over the hump is present. The transition from subcritical to supercritical flow is present also in test (ii) but without any shock because a supercritical outflow is imposed. Finally in test (iii) the supercritical inflow and outflow are imposed and the Froude number is greater than 1 all over the domain. Table 4.1 shows the details of the BCs of each test. Figures 4.3-4.5 show the converged steady-state numerical water depth, velocity and discharge along the channel. The agreement between the numerical and the analytical solutions is satisfactory. In order to verify the attainment of a steady condition in time a global relative error L_s is calculated at each time step using the Equation 1.12; when the norm has reached a value of 1 x 10^{-3} then the steady condition is said to be achieved and the simulation is stopped, in Figure 4.6 the norm L_1 calculated for the test (ii) is plotted.

Table 4.1. Boundary conditions for a 1D steady flow over a bump

test	inflow upstream BC	outflow downstream BC
i	0.435 m/s	0.33 m
ii	0.435 m/s	0.10 m
iii	4.000 m/s	0.10 m

Many authors simulated these test cases using Finite Volume schemes in order to validate the discretization method of the slope source term: Vázquez-Cendón [1999] and other prosed an upwinding of the bed source term, Zhou et al. [2001a] introduced an Surface Gradient Method where the water depth is reconstructed starting from the water surface level, Aureli et al. [2008] combined the SGM and the Depth Gradient Method reconstruction, finally Rogers et al. [2003] proposed a different approach where a mathematical balancing of the flux gradient and source terms is introduced.

The simulation of test case (i) is the most severe and some small oscillations are present in the region near the shock (particularly in the discharge), but this is a well known problem reported also by the authors previously cited, despite of this small deviation the agreement between the numerical and the analytical solution is very close. For test cases (ii) and (iii) Aureli et al. [2008] showed that in the results obtained by the DGM reconstruction some spurious oscillations are present in the discharge. Conversely the solution obtained by the SPH-SWEs for these two test cases presents no oscillations and they are in good agreement with the analytical solution.



(a)

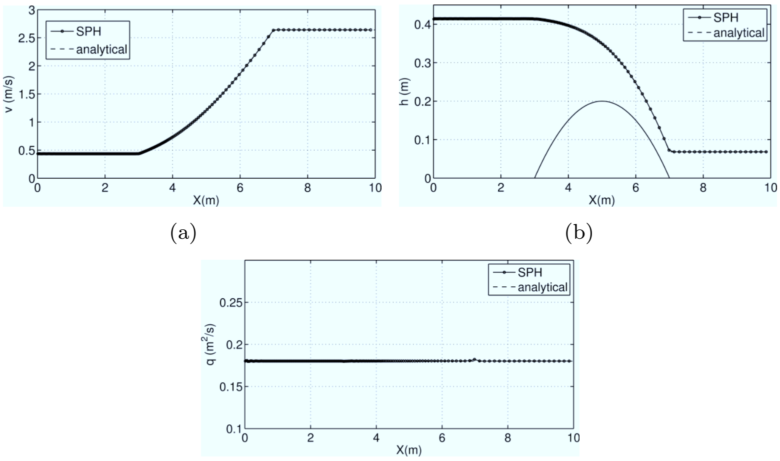
(b)

(c)

Figure 4.3. Steady transcritical flow over a bump with a shock (test i): velocity (a), water surface elevation (b) and discharge (c)

4.4 2-D Uniform flow in a sloping rectangular channel

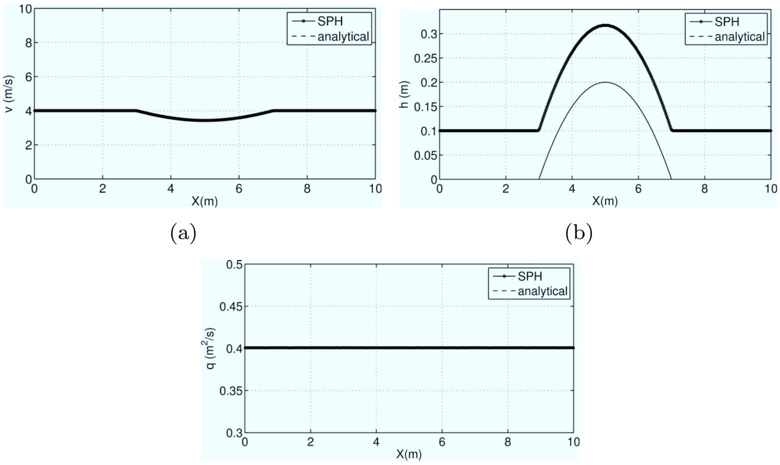
The simulation of a steady flow in a rectangular straight open channel 1000 m long and 400 m wide, with a constant slope of $s_0 = 0.001$ is performed in this



(a) (b)

(c)

Figure 4.4. Steady transcritical flow over a bump without a shock (test ii): velocity (a), water surface elevation (b) and discharge (c)



(a) (b)

(c)

Figure 4.5. Steady supercritical flow over a bump (test iii): velocity (a), water surface elevation (b) and discharge (c)

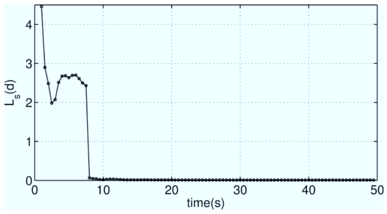


Figure 4.6. Steady supercritical flow over a bump (test ii): L_s water depth norm

section. The purpose of this test is to verify the capability of the 2-D SPH-SWEs code to simulate the inflow and outflow boundary conditions, moreover both the friction and the slope source terms have to be correctly simulated in order to obtain agreement between the numerical and the analytical solutions.
The analytical solution for this problem is given the Chezy formula:

$$\begin{aligned} v_x^{ex} &= \tfrac{1}{n} R^{1/6} \sqrt{R s_0} = 2.929 \text{ m/s} \\ v_y^{ex} &= 0 \\ d^{ex} &= 5. \end{aligned}$$

where R is the hydraulic radius (which is equal to the water depth d), s_0 is the constant slope of the channel, and n is the Manning coefficient taken equal to 0.0316 $\text{sm}^{-1/3}$.
At the begin of the simulation 4,141 motionless particles where placed in the domain over an uniform Cartesian grid, an inflow boundary condition with a constant velocity equal to the analytical solution is imposed upstream whereas an outflow boundary condition ($d = 5$ m) is assigned downstream. The bottom and friction source terms are discretize using 13,832 bottom particles.
The simulation was run for 2000 s when the steady state condition is reached. In Figure 4.7 the particles position and velocity vectors are plotted and Figure 4.8 shows the water depth and velocity magnitude maps. Finally in figure 4.9 the water depth and the velocities are plotted at three different sections along the channel: y=100, y=200 and y=300 m.
The maximum nondimensional deviations of v_x v_y and d_w are:

$$\begin{aligned} &\max\left[\frac{v_x - v_x^{ex}}{\sqrt{g d^{ex}}}\right] = 0.26 \\ &\max\left[\frac{v_y}{\sqrt{g d^{ex}}}\right] = 0.66 \\ &\max\left[\frac{d - d^{ex}}{d^{ex}}\right] = 1.09 \end{aligned}$$

and therefore the numerical model is able to reproduce the analytical solution in a satisfactory way. This is obtained because the numerical approximation of the slope and the friction source terms introduced in sections 2.3-2.4 are adequate. Moreover the treatment of the inflow and outflow boundary condition presented in this chapter is validated also for the 2-D SPH-SWEs model.

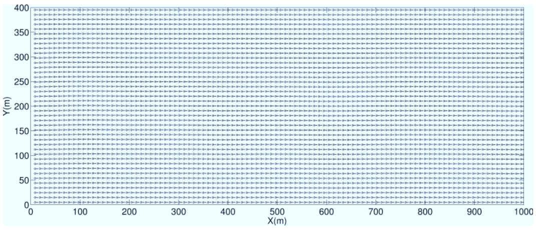
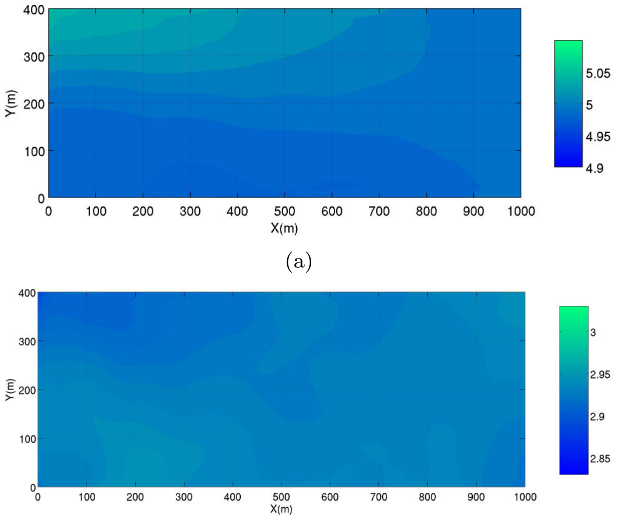


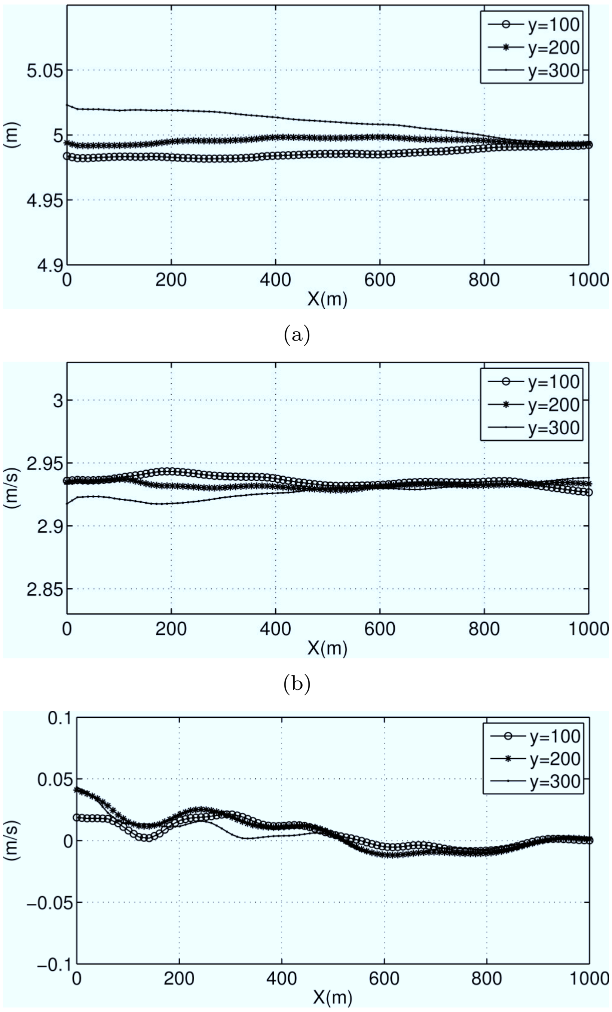
Figure 4.7. 2-D Uniform flow in a sloping rectangular channel: particle position and velocity vectors



(a)

(b)

Figure 4.8. 2-D Uniform flow in a sloping rectangular channel: (a) water depth map, (b) velocity magnitude map



(a)

(b)

(c)

Figure 4.9. 2-D Uniform flow in a sloping rectangular channel: water depth (a), velocity v_x (b) and velocity v_y (c) evaluated at sections y=100, y=200 and y=300

4.5 1993 Okushiri tsunami

This test case concerns the numerical simulation of a laboratory experiment. In 1993 the Okushiri tsunami produced the flooding of the coast near Monai in Japan, and this phenomena was later reproduced by a physical model built in 1:400 scale with dimensions 5.448 x 3.402 m. In the real event the tsunami generated a very high run-up of 31.7 m (equal to 0.079 m in the scale model) which was recorded by debris and caused numerous fatalities. The incoming wave was induced in the wave tank by a mechanical paddle placed at x=0 and the water elevation was measured by three gauges at locations: (4.521,1.196) (4.521,1.696) and (4.521,2.196). The bathymetry and the gauges position are plotted in Figure 4.10, and the water surface elevation boundary condition is plotted in Figure 4.11. The registered datasets are available at the Third International Workshop on Long Wave Run-up Models (2004) website.

This test case has a number of difficult aspects: open and closed boundaries, irregular bathymetry, wetting and drying fronts and complex shape of the reflected waves due to the shape of the coast line and to the presence of the island close to the beach; due to these features many authors simulated this case by SWEs Eulerian numerical code (Nikolos and Delis [2009], Delis et al. [2008], LeVeque and George [2007]).

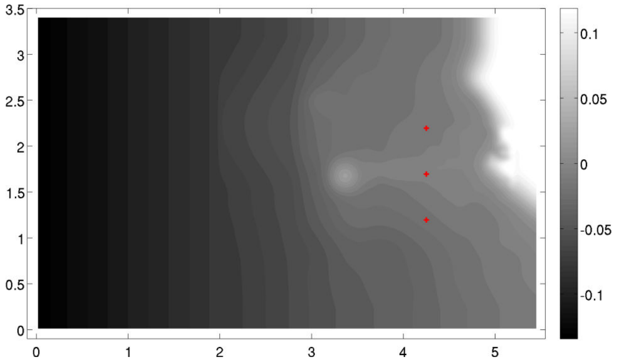


Figure 4.10. 1993 Okushiri tsunami: bathymetry of the domain and three gauges position

In this section a comparison between the experimental data and the numerical results is done in order to show the capability of the SPH-SWEs model to reproduce real flooding events. As suggested by Delis et al. [2008] the Manning coefficient was set equal to $0.025\mathrm{sm}^{-1/3}$. Two different simulations have been conducted: (i) in the first simulation 47,673 fluid particles are initially placed

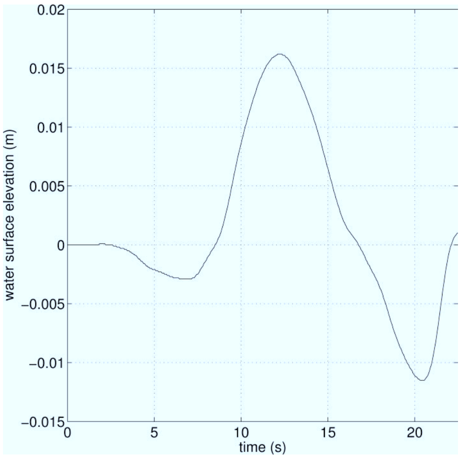


Figure 4.11. 1993 Okushiri tsunami: water level elevation imposed at x=0

in the domain over a Cartesian grid whereas, (ii) in the second simulation the same resolution is used but particle splitting procedure has been activated and close to the shoreline ($x > 2.50$m) the threshold area $\bar{A}$ is taken equal to $0.9A_0$ where A_0 is the area the particles at the beginning of the simulation. In this way the particles in this part of the domain are split in the first time step of the simulation according to the procedure described in chapter 3 and the number of fluid particles after the first time step becomes 186,081; in the remaining part of the domain $\bar{A}$ is taken equal to 5.0 in order to prevent any splitting. Therefore in this test case the particle splitting procedure is not used to avoid the lack of resolution as in the 1D-2D dam break (see section 3.4.2) but to increase the resolution in the part of the domain where the flow presents more complex behaviour.

In Figures 4.12 and 4.13 the maps and the 3-D views of the water depth are plotted at different time steps for the first simulation (with no splitting). In Figure 4.14 is plotted the comparison between numerical and experimental water elevation at the three gauges for the same simulation. Figures 4.15 - 4.16 show the water depth maps at different time steps for the second simulation (with splitting) whereas the comparison with the registered water level is plotted in Figures 4.14 and 4.17.

The arrival of the tsunami is anticipated by a draw-down that can be seen in the water depth maps of time steps 12 and 14 s and this causes a partial drying of the ridge behind the island. When the main tsunami wave is approaching the beach the diffracted waves meet together behind the island (at time steps 15 and

16 s). At time 17 the tsunami reaches the beach and the wetting - drying front moves forward over the initially dry land. At time steps 18 - 20 the shape of the reflected waves from the coast can be observed. Both the simulations without and with splitting are able to reproduce these characteristics of the phenomenon. In the simulation made by Nikolos and Delis [2009] the ridge behind the island is dry in the first time steps and the same area is wet in Figures 4.12, 4.13, 4.15, 4.16. This seems an important discrepancy between the two results but the water depth calculated by the SPH-SWEs numerical model is very shallow in that region and probably it is less than the threshold value that Nikolos and Delis [2009] used to localise the wet-dry front, another reason for this difference can be due to the postprocessing procedure (see 2.6) used in the SPH-SWEs code to generate the water depth maps.

Figure 4.18 shows the zoom of the particles position at the maximum run-up locations when the maximum run-up occurs (time 17s) with and without particle splitting. In the simulation with no splitting procedure the maximum run-up simulated is 0.07 m, whereas the registered value of 31.7 m in the real events corresponds to 0.08 m in the scale model. This underestimation is caused by the lack of resolution in the maximum run-up area. Conversely in the simulation with refinement activated the number of particles is clearly larger in the area where the maximum run-up occurs and the maximum run-up registered in the numerical models is very close to the registered value of 0.08 m.

4.6 Flood inundation in Thamesmead

In this section a real flood risk analysis problem is performed with the aim to show the capability of the SPH-SWEs to reproduce real scale events. The simulation is carried out for the 9 x 4 km Thamesmead site which is a low-lying area located in the estuary of the River Thames in United Kingdom protected from tidal flooding by a system of levees that separate it from the Thames riverbed. The area is densely populated therefore an hydraulic modelling is required to assess the nature of the flooding if a breaching of the flood defences occurs. A reference numerical simulations have been carried out by using the commercial software TUFLOW. Liang et al. [2008] compared the results of their Finite Volume code against this reference solution, in this section the same test case is simulated by the SPH-SWEs code and a comparison against the TUFLOW results is shown.

Figure 4.19 shows the bathymetry of the site: it is characterized by some low

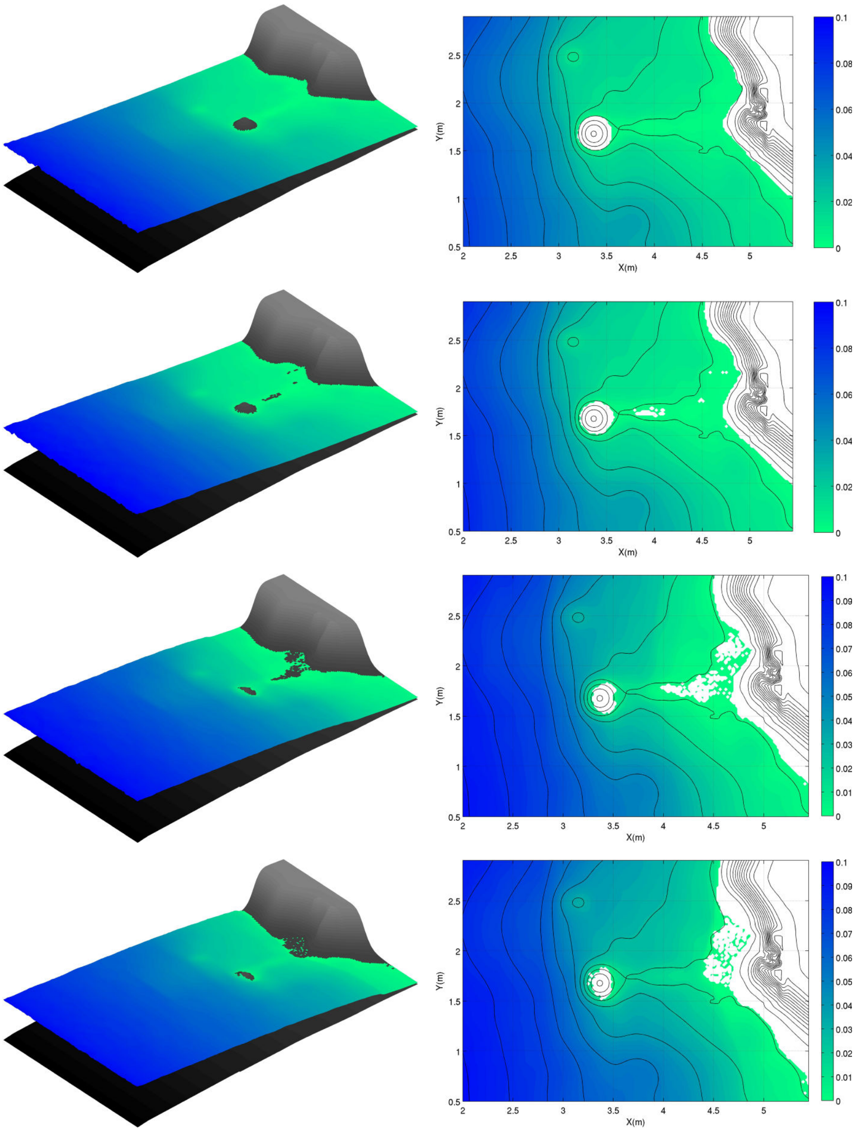


Figure 4.12. 1993 Okushiri tsunami simulation without splitting: 3D view (left) and contour (right) of water depth at times 10, 12, 14 and 15 s

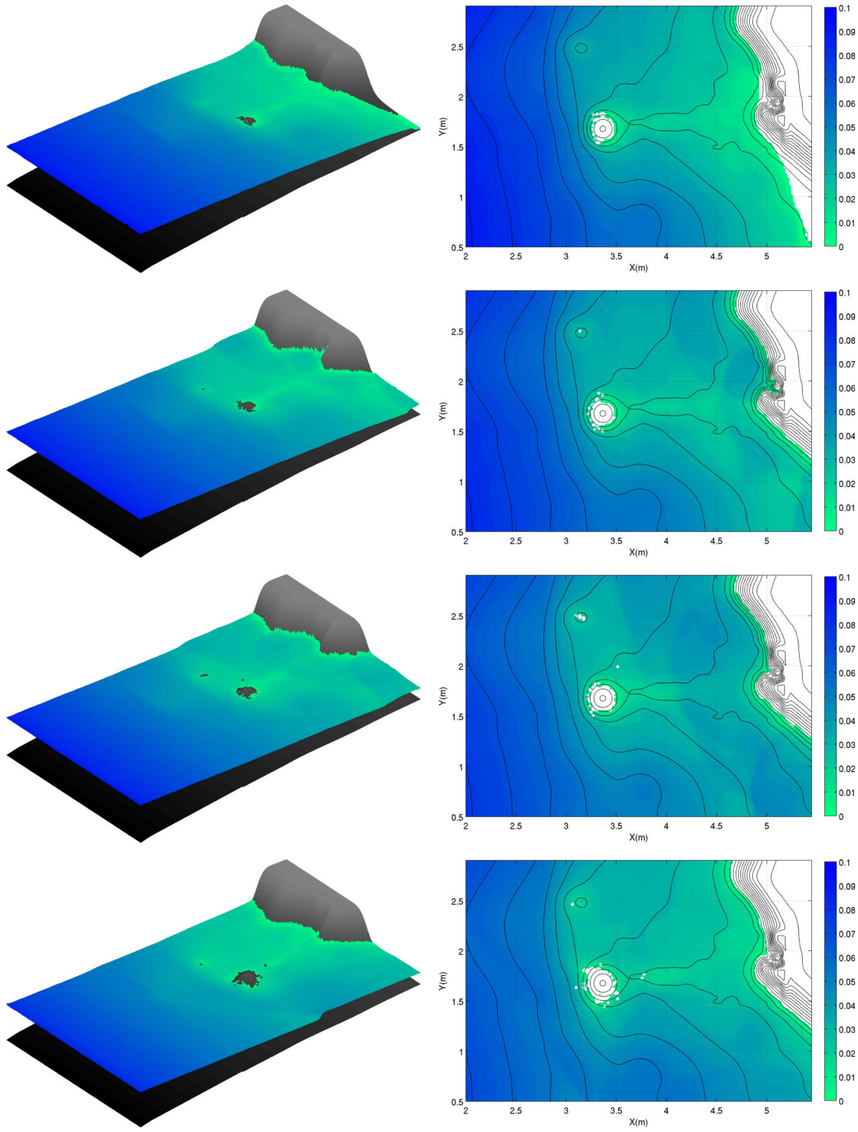


Figure 4.13. 1993 Okushiri tsunami simulation without splitting: 3D view (left) and contour (right) of water depth at times 16, 17, 18 and 20 s

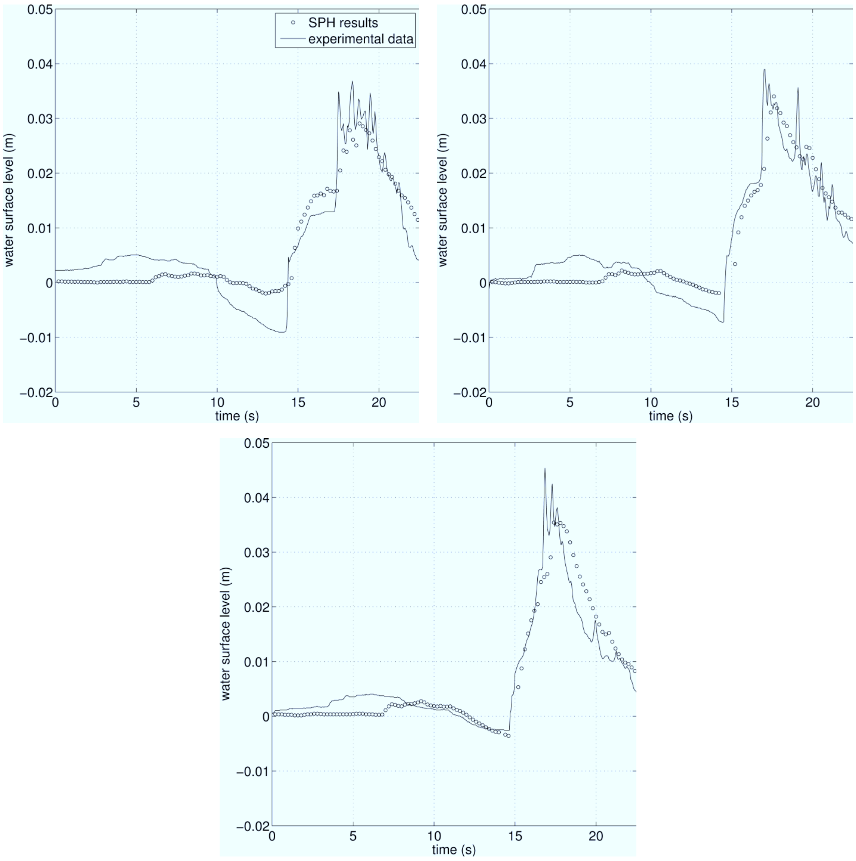


Figure 4.14. 1993 Okushiri tsunami simulation without splitting: surface elevation at three gauges

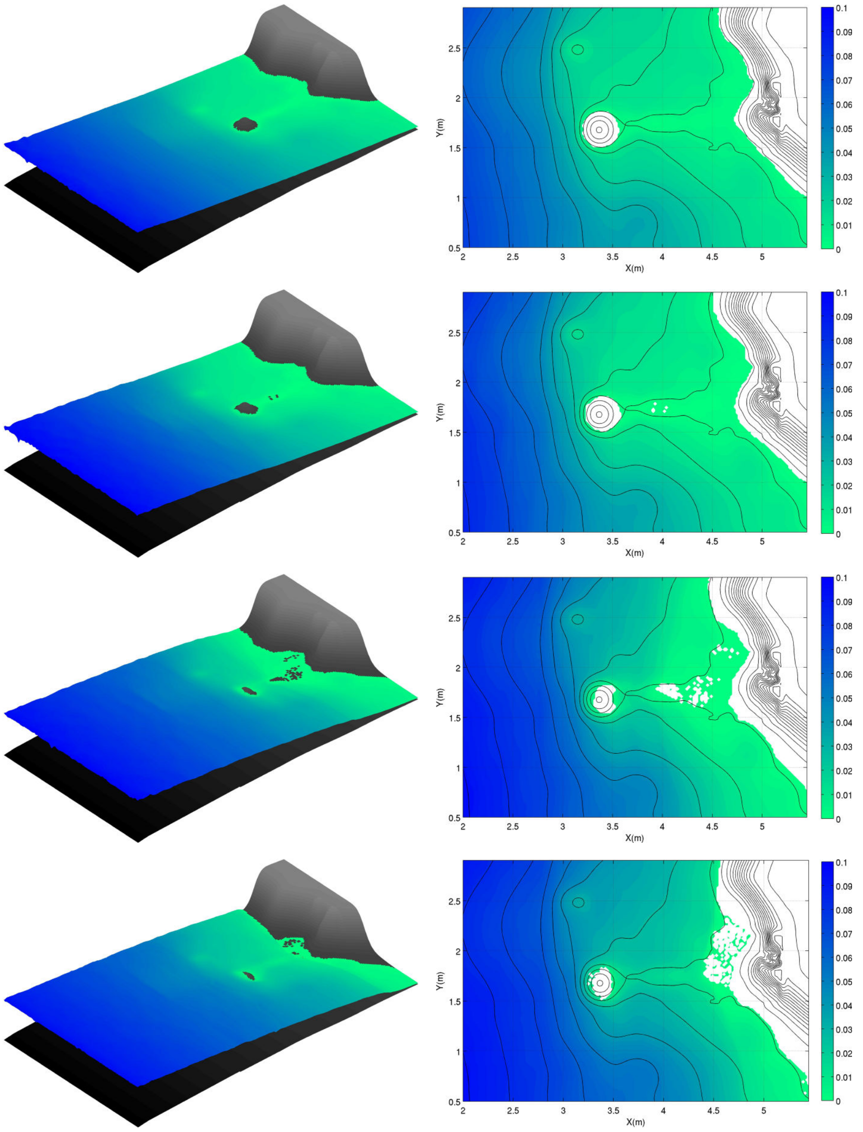


Figure 4.15. 1993 Okushiri tsunami simulation with splitting: 3D view (left) and contour (right) of water depth at times 10, 12, 14 and 15 s

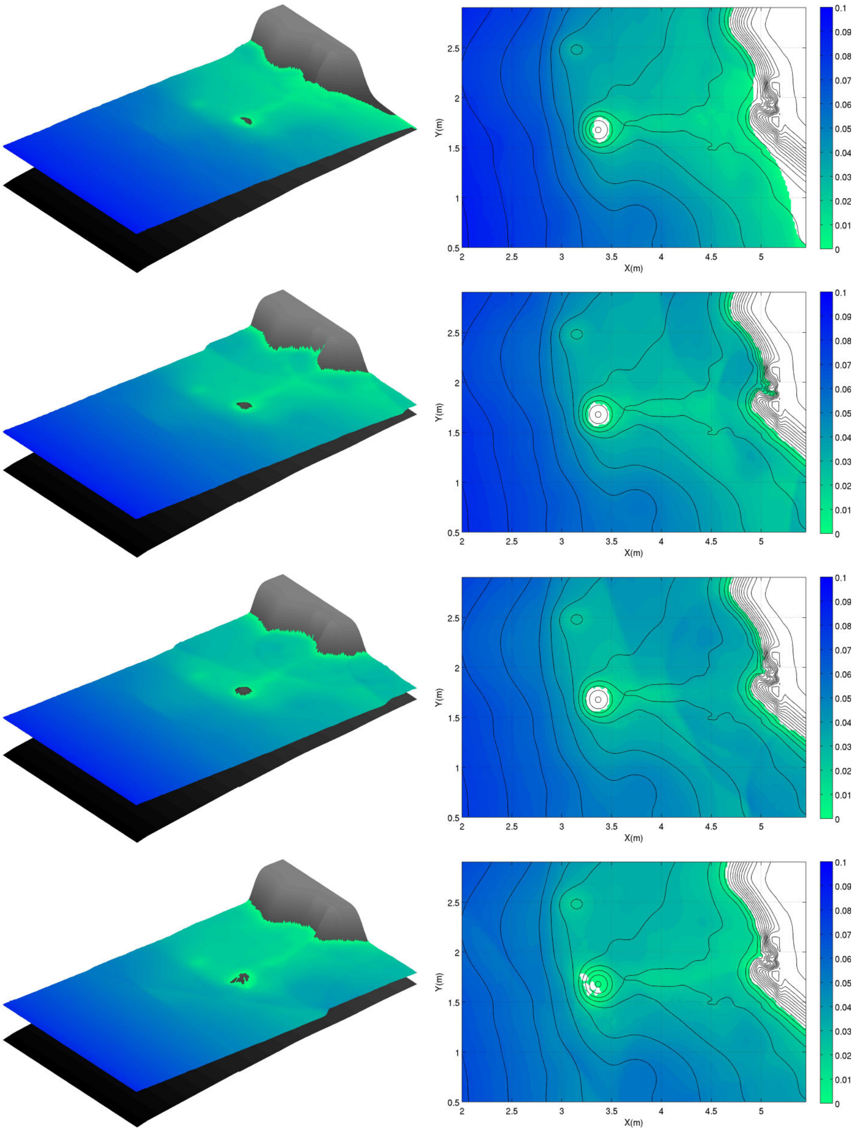


Figure 4.16. 1993 Okushiri tsunami simulation with splitting: 3D view (left) and contour (right) of water depth at times 16, 17, 18 and 19 s

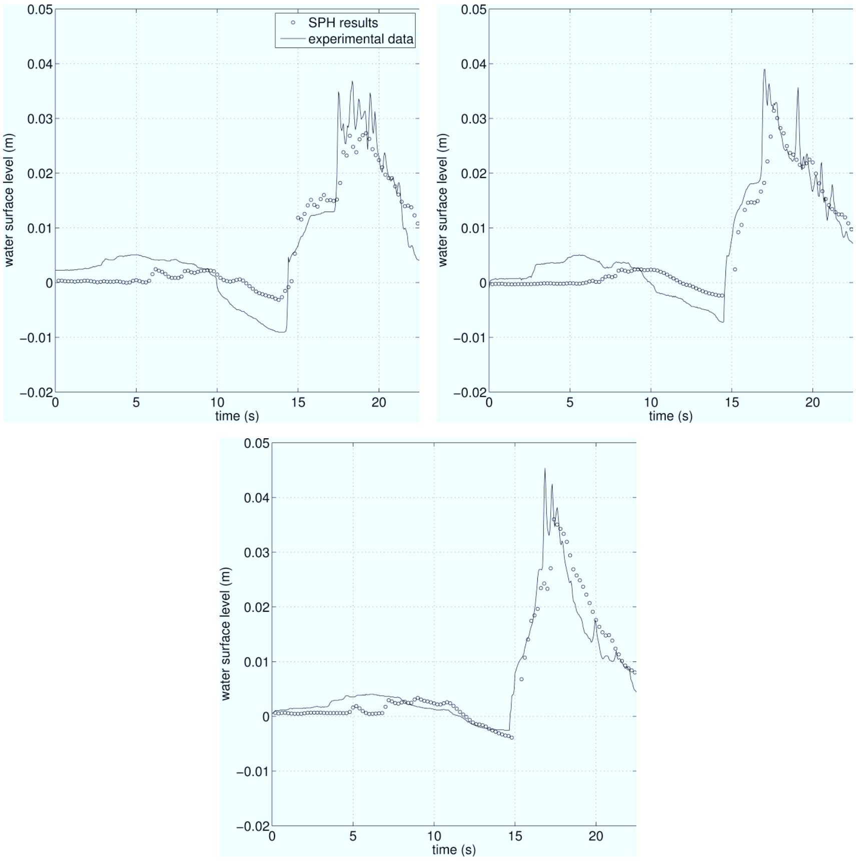
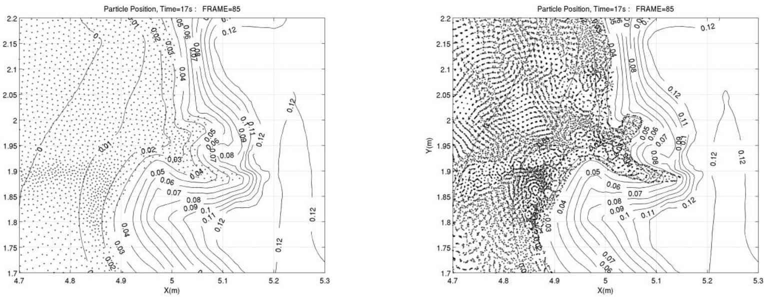


Figure 4.17. 1993 Okushiri tsunami simulation with splitting: surface elevation at three gauges



(a)

(b)

Figure 4.18. 1993 Okushiri tsunami simulation: zoom in the area with maximum water elevation at time 17 s without (a) and with particles splitting simulation (b) areas and by the presence of an railway embankment running from the south west to the centre of the domain.

In order to reproduce the results obtained by TUFLOW a breach 150 m long is considered; it starts at point (545785,181045) and ends at point with coordinates (545935,181045); along this breach the inflow discharge plotted in Figure 4.20 is imposed. This discharge is uniformly distributed along the breach using open boundary particles 15 m distant from each other. The flow never reaches the other boundaries because they are much higher than the floodplain, hence, no other boundary condition are imposed.

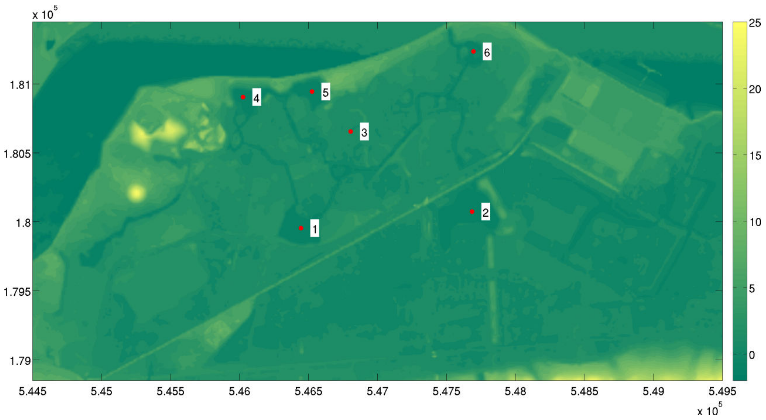


Figure 4.19. Thamesmead test case: bathymetry of the site and position of the 6 gauges, the color axis is expressed in meter

The Manning coefficient is taken equal to 0.035 sm$^{1/3}$ and the floodplain is considered dry at the begin of the simulation so no fluid particles are placed in the domain at the begin of the simulation. The bathymetry is discretized using 359,976 bottom particle placed over an uniform Cartesian grid with 10 m side

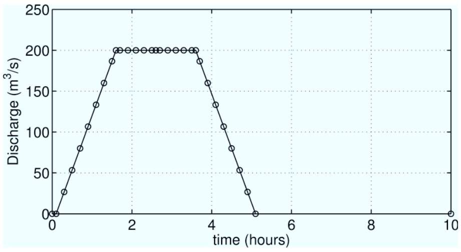


Figure 4.20. Thamesmead test case: discharge boundary condition imposed through the breach

length.

The results of the SPH simulation are compared with the one obtained by TUFLOW using a 10 m uniform grid. Figures 4.21 - 4.24 show the comparison of the water depth maps obtained by the two models at different times. Results of the SPH-SWEs numerical model agree with the TUFLOW simulations and the leading characteristics of the inundation phenomena caused by the complex bathymetry are properly described by the meshless Lagrangian model. The discrepancies in the position of the flood fronts in the last time steps (5 and 6 hours) are due to the fact that the water is very shallow in the region closed to the front therefore the particles density is low and this causes some problems in the post-processing algorithm used to plot the water depth maps (see section 2.6). Figure 4.25 shows the comparison of the water depth time history at locations plotted in Figure 4.19. The maximum differences of the water depth obtained by the two models are less than 0.5 m and the arrival time of the flooding obtained by the two models presents no substantial differences.

To the best author knowledge this is the first flooding simulation of a real scale events over a real initially dry bathymetry. Therefore the major purpose of this test case is to show that the SPH-SWEs model can be successfully applied to this kind of phenomena. Nevertheless some future analysis will concern the introduction of a dynamic particle splitting/coalescing in order to avoid the lack of resolution when the flooding of the floodplains occurs.

4.7 Concluding Remarks

In this chapter a method to insert Open Boundary condition in the SPH-SWEs numerical model has been presented. Boundary condition has been imposed using Open Boundary Particles placed in a buffer zone in order to prevent any kernel truncation in the fluid particles. The method has been tested against the analytical solutions available for the 1-D steady flow over a bump and accurate results were obtained for both supercritical and subcritical inflow/outflow boundary conditions. Moreover The 2-D Uniform flow in a sloping rectangular channel has been simulated in order to verify the effectiveness of the method when applied to a 2-D test case and reasonably accurate results were obtained. To the author's knowledge these are the firsts simulations where open boundaries are applied in a SPH-SWEs numerical model.

Finally the simulation of two real test cases has been made. The first one is the Okushiri tsunami where the results with and without particle splitting pro-

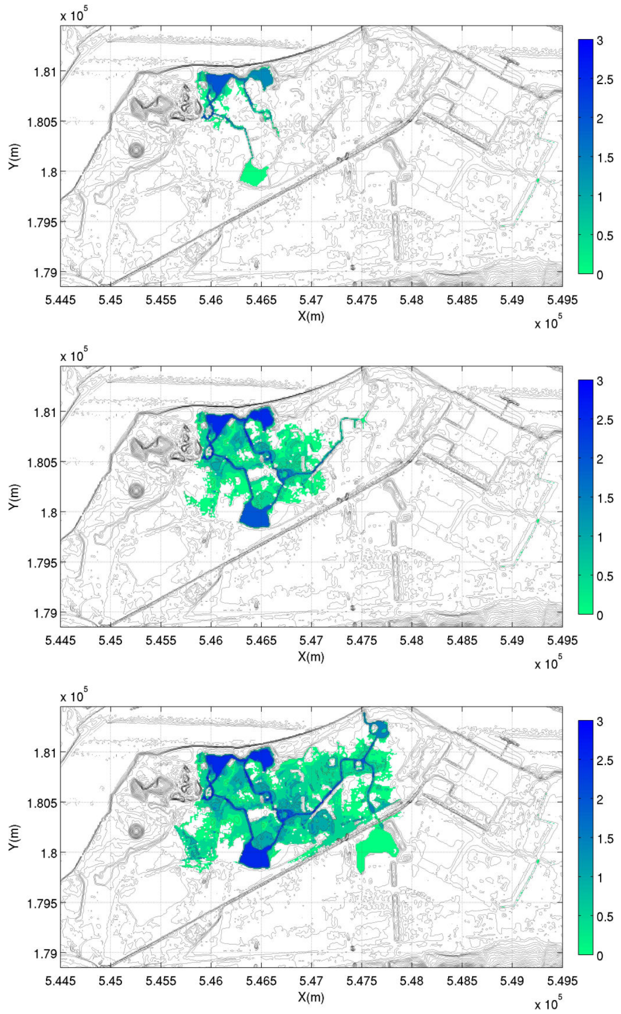


Figure 4.21. Thamesmead test case: contour plot of TUFLOW simulation at times 1, 2 and 3 hours

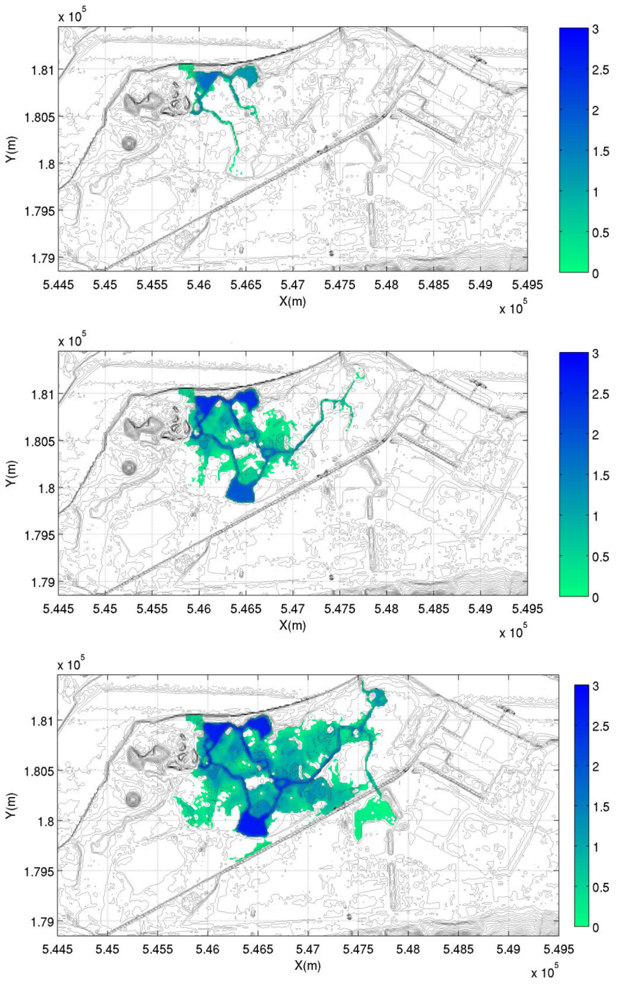

x 10^5
1.81
1.805
1.8
1.795
1.79
Y(m)
5.445
5.45
5.455
5.46
5.465
5.47
5.475
5.48
5.485
5.49
5.495
X(m)
x 10^5
3
2.5
2
1.5
1
0.5
0

Figure 4.22. Thamesmead test case: contour plot of SPH simulation at times 1, 2 and 3 hours

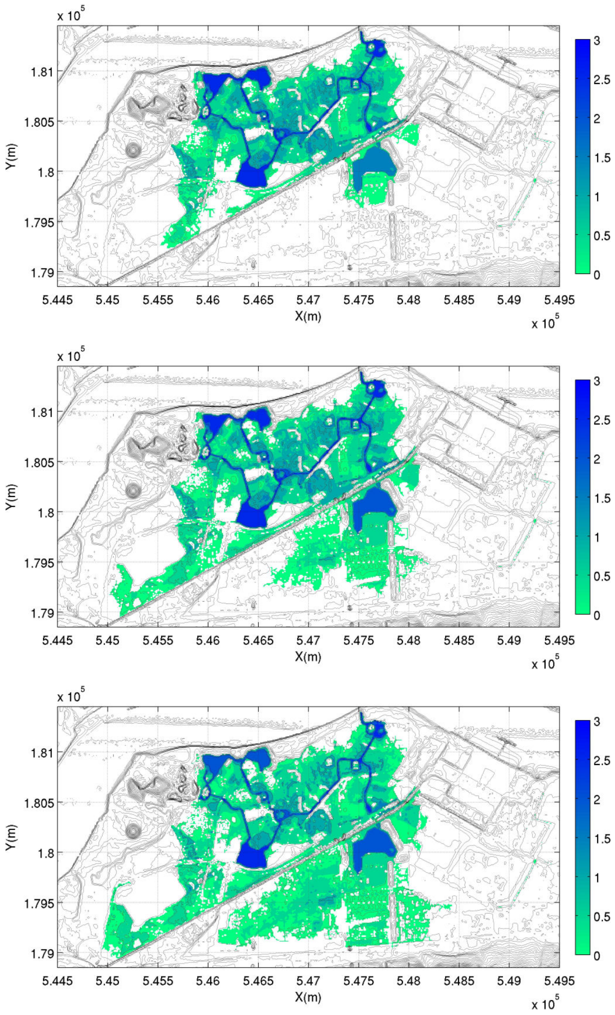

x 10^5
1.81
1.805
1.8
1.795
1.79
Y(m)
5.445
5.45
5.455
5.46
5.465
5.47
5.475
5.48
5.485
5.49
5.495
X(m)
x 10^5
3
2.5
2
1.5
1
0.5
0

Figure 4.23. Thamesmead test case: contour plot of TUFLOW simulation at times 4,5 and 6 hours

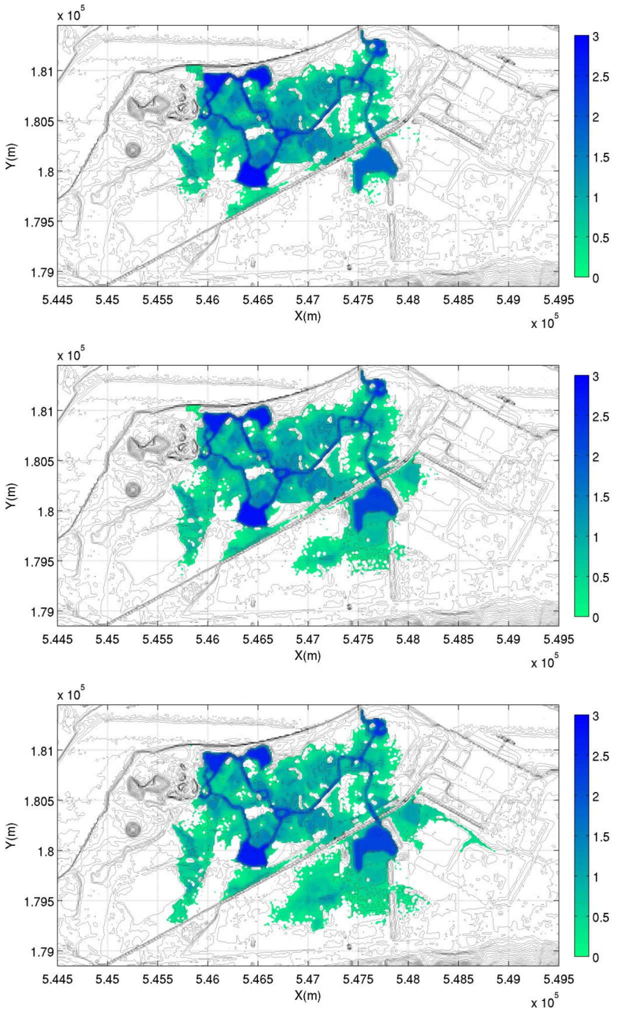

x 10^5
1.81
1.805
1.8
1.795
1.79
Y(m)
5.445
5.45
5.455
5.46
5.465
5.47
5.475
5.48
5.485
5.49
5.495
X(m)
x 10^5
3
2.5
2
1.5
1
0.5
0

Figure 4.24. Thamesmead test case: contour plot of SPH simulation at times 4, 5 and 6 hours

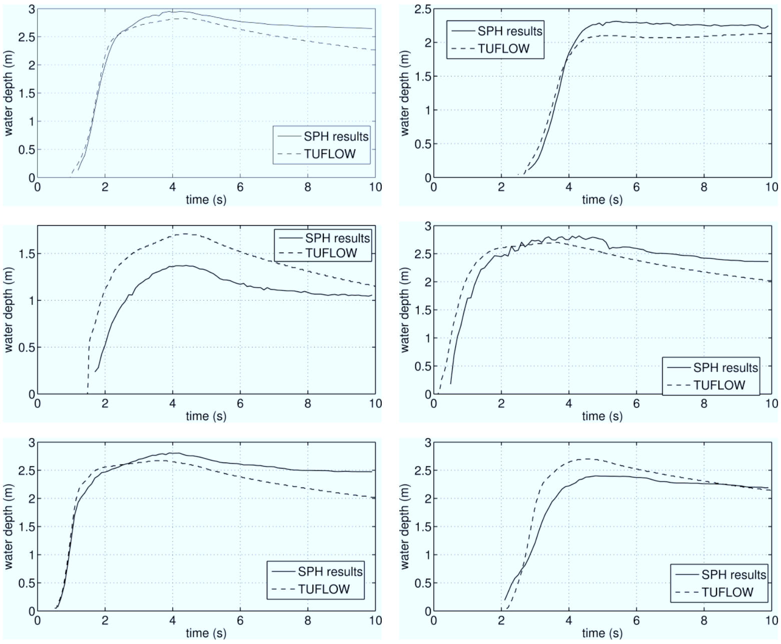


Figure 4.25. Thamesmead test case: water depth registered at different gauges

cedure has been adopted. A reasonable agreement with the registered levels is obtained, moreover the maximum run-up is well reproduced and the effectiveness of the splitting procedure in increasing the resolution in the portion of the domain closer to the beach is shown.

The model was later applied applied for a flood risk analysis, and the simulation of the breaching in a flood defence at Thamesmead location (United Kindom) has been made. The comparison of the results obtained against the TUFLOW simulation show that the SPH-SWEs numerical model can be able successfully applied in flood inundation simulations also in real scale studies.

Chapter 5

Finite Pointset Method for free-surface Flow

5.1 Finite pointset method and SPH

This final analysis chapter introduce a more recent meshless method, the Finite Pointset Method (FPM), that could well represent the future of particle methods and SPH. FPM is a Lagrangian meshless method for numerical integration of pure incompressible Navier-Stokes equations originally introduced by Tiwari and Kuhnert [2003]. This method is similar to incompressible SPH (Lee et al. [2008], Shao and Lo [2003], Hu and Adams [2007]) because each particle carries a vector of field. Information and physical quantities are approximated using particles in a circular neighbourhood.

There are also some fundamental differences between the two methods: FPM is based on a moving least squares approach, where particles are just interpolation points without any associated mass. Due to these key features, boundary conditions can be enforced analytically using boundary particles, and particles can be added and removed in order to preserve the stability of the solution. To date the FPM has been confined to single or two phase internal flow, in this chapter a free surface version of FPM is introduced: a novel, pure geometrical algorithm to detect the free surface boundary condition is proposed. The second major modification introduced is about the capability to reproduce the hydrostatic condition in time. This capability is one of the fundamental features of free-surface numerical models, however classical FPM is not able to reproduce that condition;

to overcome this problem a modification of the Incremental Pressure Projection Method is used to solve the Navier Stokes equations.

5.2 Incremental Pressure Projection Method

We consider the incompressible Navier-Stokes equations, written using the Lagrangian derivative

$$\nabla \cdot \mathbf{v} = 0 \tag{5.1}$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{f}_e + \Theta \tag{5.2}$$

where $\mathbf{v}$ is the velocity, p is the pressure, ρ is the density of the fluid, $\mathbf{f}_e$ is the vector of external forces and Θ is the viscous term equal to $\nu\nabla^2\mathbf{v}$ for a Newtonian fluid, where ν is the kinematic viscosity of the fluid.
Spatially discrete versions of the coupled Navier-Stokes equations are cumbersome to solve directly. Observing that the right-hand side of equation (5.2) is a Hodge decomposition, Chorin [1968] proposed the Projection Method (PM) in order to solve this system of partial differential equations. The key idea of the PM is to introduce a fractional step procedure: an intermediate velocity $\mathbf{v}^*$ is computed neglecting the gradient pressure in equation 5.2, then, projecting the final velocity onto a divergence free space, the pressure p is calculated solving a Laplace equation.

If the classical Projection Method is applied in a bounded domain Ω, then an issue arises: boundary conditions applied calculating $\mathbf{v}^*$ should be consistent with the final velocity although the final pressure p^{n+1} is not known, this problem is analysed for Eulerian models by many authors (see for example Bell et al. [1989] or Kim and Moin [1985]). The Incremental Pressure Projection Method (IPPM) was introduced by Brown et al. [2001]; the version presented herein is an adaption for Lagrangian projection methods.
The following discrete implicit formulation of (5.2) is initially considered

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} + \frac{1}{\rho}\nabla p^{n+1} = 0.5\left(\Theta^n + \Theta^{n+1}\right) + \mathbf{f}_e; \tag{5.3}$$

with boundary condition $\mathbf{v} = \mathbf{v}_b$ on $\partial\Omega$.
In IPPM Equations (5.1) and (5.3) are solved by means of Hodge decomposition

as in Chorin's original method. In the first step a predicted velocity is computed:

$$\begin{aligned} \tfrac{\mathbf{v}^*-\mathbf{v}^n}{\Delta t} + \tfrac{1}{\rho}\nabla q = 0.5\left(\Theta^n + \Theta^{n+1}\right) + \mathbf{f}_e \\ B(\mathbf{v}^*) = 0 \;\partial\Omega \end{aligned} \tag{5.4}$$

This equation differs from the one used in classical PM because the viscous term is now discretized implicitly, and an approximation q of the pressure is added. A boundary condition $B(\mathbf{v}^*) = 0$ for $\mathbf{v}^*$ is also introduced.

The second step is the projection of $\mathbf{v}^*$ in divergence free space:

$$\mathbf{v}^* = \mathbf{v}^{n+1} + \Delta t \frac{1}{\rho} \nabla \phi^{n+1} \tag{5.5}$$

where ϕ is not the pressure, as in PM, but a correction to the approximation q. Equation (5.5) is solved using boundary conditions consistent with $B(\mathbf{v}^*) = 0$ and $\mathbf{v}^{n+1} = \mathbf{v}_b^{n+1}$ on $\partial\Omega$.

Substituting (5.5) into (5.1) leads to the following Laplace equation:

$$\nabla \cdot \frac{1}{\rho}\left(\nabla \phi^{n+1}\right) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t} \tag{5.6}$$

The third and last step of IPPM is the pressure update:

$$p^{n+1} = q + L(\phi^{n+1}) \tag{5.7}$$

where the function L represents the dependence of p^{n+1} on ϕ^{n+1}. Once the time step is completed, the predicted velocity $\mathbf{v}^*$ is not used again at that time or any later time.

In order to define the IPPM, the pressure approximation q, the boundary condition $B(\mathbf{v}^*) = 0$ and the function $L(\phi^{n+1})$ have to be defined. An obvious choice for the pressure approximation q into (5.4) is the pressure at the previous time step, so: $q = p^n$. Assuming that q is a good approximation for p^{n+1}, then the field of predicted velocity will not differ too much from the final one and $\mathbf{v}^* = \mathbf{v}^{n+1}$ on $\partial\Omega$ is the boundary condition for $\mathbf{v}^*$.

Substituting (5.5) into (5.4) and comparing to (5.3) for a Newtonian fluid we obtain the following formulation for L

$$L(\phi^{n+1}) = \phi^{n+1} - \frac{\nu \Delta t}{2} \nabla^2 \phi^{n+1} \tag{5.8}$$

5.3 FPM implementation

5.3.1 Moving Least Squares approximation

Let a scalar function be defined as $f : \Omega \rightarrow \mathbb{R}$ and $i = 1, \cdots, N$ a set of discrete points in the domain Ω where the function f is known. Moving Least Squares is a numerical method to approximate the value of the function $f(\mathbf{x})$ and its derivatives in a generic point $\mathbf{x}$ using the point clouds around $\mathbf{x}$ (see Dilts [1999]). The point cloud of $\mathbf{x}$ is defined as a set of points $\{x_1, x_2, \cdots x_n\}$ that satisfy the following property: $\|\mathbf{x}-\mathbf{x}_i\| \leq h(\mathbf{x})$ where $h(\mathbf{x})$ is similar to the smoothing length in SPH method (Monaghan [1992]).
In order to compute the approximation of $f(\mathbf{x})$ and its derivatives n Taylor expansion of $f(\mathbf{x}_i)$ around $\mathbf{x}$ are written:

$$\begin{aligned} f\left(\mathbf{x}_i\right) = f\left(\mathbf{x}\right) + \sum_{k=1}^{3} \left(\frac{\partial f}{\partial x_k}\right)_{\mathbf{x}_i} \left(x_{k,i} - x_k\right) \\ + \sum_{k,l=1}^{3} \left(\frac{\partial^2 f}{\partial x_k x_l}\right)_{\mathbf{x}_i} \left(x_{k,i} - x_k\right)\left(x_{l,i} - x_l\right) + e_i; \\ \forall\, i = 1, 2, \cdots, n \end{aligned} \tag{5.9}$$

where e_i is the error in the Taylor series at the points $\mathbf{x}_i$ and the unknowns are $f\left(\mathbf{x}\right)$ and their derivatives. These unknowns are computed by minimizing the error e_i for $i = 1, 2, ..., n$. The system of equations can be written in two dimensions and in matrix form as:

$$\mathbf{e} = M\mathbf{a} - \mathbf{b} \tag{5.10}$$

where:

$$M = \begin{bmatrix} 1 & dx_1 & dy_1 & 1/2dx_1^2 & dx_1dy_1 & 1/2dy_1^2 \\ 1 & dx_2 & dy_2 & 1/2dx_2^2 & dx_2dy_2 & 1/2dy_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & dx_n & dy_n & 1/2dx_n^2 & dx_ndy_n & 1/2dy_n^2 \end{bmatrix}$$

$$\mathbf{a} = \left[f, \left(\frac{\partial f}{\partial x}\right)_{\mathbf{x}_i}, \left(\frac{\partial f}{\partial y}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 f}{\partial x^2}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 f}{\partial xy}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 f}{\partial y^2}\right)_{\mathbf{x}_i} \right]^T$$

$$\mathbf{b} = [f_1, f_2, \cdots, f_n]^T$$

$$\mathbf{e} = [e_1, e_2, \cdots, e_n]^T$$

The unknown vector $\mathbf{a}$ is then obtained by minimizing the quadratic form J defined as follows:

$$J = \sum_{i=1}^{n} \omega_i e_i^2 \tag{5.11}$$

where ω_i is the weight of the ith error, and can be expressed using any bell-shaped function. In this work we used a Gaussian function:

$$\omega_i = \begin{cases} exp\left(-\alpha \frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{h^2}\right) & if\ R^2 \leq 1 \\ 0 & if\ R^2 > 1 \end{cases} \tag{5.12}$$

where $R = \parallel \mathbf{x}_i - \mathbf{x} \parallel /h$ and α is a coefficient taken equal to 6.25 (Tiwari and Kuhnert [2007]).

The value of h is not fixed (see paragraph 5.3.4), but the initial value h_0 is taken as $h_0 = 3.2\ dx$ where dx is the initial particle spacing.

The minimization of J leads to the following 6 × 6 linear system of equations in 2D (and 10 × 10 in 3D)

$$\left(M^T W M\right) \mathbf{a} = \left(M^T W\right) \mathbf{b} \tag{5.13}$$

where $W = diag\left(\omega_1, \omega_2, \cdots, \omega_n\right)$. By solving this system of equations is possible to obtain the unknown vector $\mathbf{a}$ and its derivatives anywhere in the domain.

In contrast to the SPH intepolation method used in the first part of this work, this technique requires the inversion of relatively large matrices for each particles and thus it is more time consuming.

5.3.2 Elliptic equations

The Moving Least Squares method outlined in the previous paragraph can also be used for solving an elliptical equation written in generic form as:

$$A\psi + \mathbf{B} \cdot \nabla \psi + C \Delta \psi = f \tag{5.14}$$

Where A, $\mathbf{B}$, C and $f = f(\mathbf{x})$ are given (for $A = 0$ and $\mathbf{B} = 0$ we recover the classical Poisson equations) and ψ is the unknown. We solve this equation with

Dirichlet $\psi = g$ or Neumann boundary condition:

$$\frac{\partial \psi}{\partial \mathbf{n}} = g \ on \ \partial\Omega \tag{5.15}$$

where $\mathbf{n}$ is the versor normal to the boundary.
Adding eqs (5.14) and (5.15) to the Taylor series of the neighbouring points (see eqs 5.9) leads to the following linear system:

$$\mathbf{e}^* = M^*\mathbf{a} - \mathbf{b}^* \tag{5.16}$$

where:

$$M^* = \begin{bmatrix} 1 & dx_1 & dy_1 & 1/2dx_1^2 & dx_1dy_1 & 1/2dy_1^2 \\ 1 & dx_2 & dy_2 & 1/2dx_2^2 & dx_2dy_2 & 1/2dy_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & dx_n & dy_n & 1/2dx_n^2 & dx_ndy_n & 1/2dy_n^2 \\ A & B_1 & B_2 & C & 0 & C \\ 0 & n_x & n_y & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{a} = \left[\psi, \left(\frac{\partial \psi}{\partial x}\right)_{\mathbf{x}_i}, \left(\frac{\partial \psi}{\partial y}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 \psi}{\partial x^2}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 \psi}{\partial xy}\right)_{\mathbf{x}_i}, \left(\frac{\partial^2 \psi}{\partial y^2}\right)_{\mathbf{x}_i}\right]^T$$

$$\mathbf{b}^* = [\psi_1, \psi_2, \cdots, \psi_n, \phi, g]^T$$

$$\mathbf{e}^* = [e_1, e_2, \cdots, e_n, e_{n+1}, e_{n+2}]^T$$

The unknowns $\mathbf{a}$ are computed by minimizing the quadratic form J as shown in paragraph (5.3.1). This leads to the following equations:

$$\mathbf{a}^* = \left(M^{*T}W^*M^*\right)^{-1}\left(M^{*T}W^*\right)\mathbf{b}^* \tag{5.17}$$

where $W = diag\,(\omega_1, \omega_2, \cdots, \omega_n, 1, 1)$ because we assign weight 1 to Equations (5.14) and (5.15).

The first line of Equation(5.17) is:

$$\begin{gathered}
\psi = \beta_1 \left(\sum_{i=1}^{n} \omega_i \psi_i + A\phi \right) \\
+\beta_2 \left(\sum_{i=1}^{n} \omega_i dx_i \psi_i + B_1\phi + n_1 g \right) \\
+\beta_3 \left(\sum_{i=1}^{n} \omega_i dy_i \psi_i + B_2\phi + n_2 g \right) \\
+\beta_4 \left(0.5 \sum_{i=1}^{n} \omega_i dx_i^2 \psi_i + C\phi \right) \\
+\beta_5 \left(\sum_{i=1}^{n} \omega_i \psi_i dx_i dy_i \right) \\
+\beta_6 \left(0.5 \sum_{i=1}^{n} \omega_i \psi_i dy_i^2 \psi_i + C\phi \right)
\end{gathered} \tag{5.18}$$

where $(\beta_1, \beta_2, \cdots, \beta_6)$ is the first raw of matrix $(M^{*T} W^{*})\mathbf{b}$.
If (5.9), (5.14) and (5.15) are written for each one of the $j = 1, \cdots, N$ discrete points used for the discretization of the domain, the summation in (5.18) can be arranged as follow

$$\begin{aligned}
&\psi_j - \sum_{i=1}^{n(j)} \omega_{ji} \big(\beta_1 + \beta_2 dx_{ij} + \beta_3 dy_{ij} + 0.5\beta_4 dx_{ij}^2 + \\
&\quad \beta_5 dx_{ij} dy_{ij} + 0.5\beta_6 dy_{ij}^2 \big) \psi_i = (\beta_1 A + \beta_2 B_1 + \\
&\quad \beta_3 B_2 + \beta_4 C + +\beta_6 C)\, \phi + (\beta_2 n_x + \beta_3 n_y +)\, g
\end{aligned} \tag{5.19}$$

where $n(j)$ is the number of neighbours of particle j. Equation (5.19) can be written for every particle of the domain, finally the discrete form of (5.14) is obtained and it is a sparse $N \times N$ linear system.
This linear system is solved using the iterative method BiCGStab (van der Vorst [1992]).

5.3.3 Particle management

One of the features of any meshless Lagrangian method is the irregular distribution of the particles in the domain. This occurs because particles are moving during the simulation; in some regions particles could be too close to each other and in some other part of the domain the number of particles could be too scarce and it is well known that this can cause numerical instabilities (Fang and Parriaux [2008]).

In the FPM particles do not carry any mass, so particle management can be done more easily than in SPH, but this requires an efficient detection procedures for nearby particles and holes, and correct interpolation of the field data when particles are deleted and inserted.

Two particles i and j are too close if:

$$\| x_i - x_j \| < k_c h_0 \tag{5.20}$$

where h_0 is the initial smoothing length of the kernel function and k_c is a coefficient significantly less than 1 (in our simulations $k_c = 0.2$). If the condition reported in Equation(5.20) holds, then the two particles are removed and a new one is inserted in a position $\mathbf{x}_{new} = 0.5(\mathbf{x}_i + \mathbf{x}_j)$.

Finding holes in the domain is a more difficult task because no information is provided where no particles exist. In order to find regions where the particles are too sparse a Delaunay triangulation is performed using all the points of the domain. If the area of a triangle is bigger than a prefixed value A_{max} then a point is inserted in the centre of mass of the triangle. Pressure, velocity, and any other physical quantities of the new particle are then obtained by means of MLS spatial interpolation. In our simulation we take $A_{max} = 0.76dx^2$ where $dx = 3.2/h_0$ is the intial particle spacing.

5.3.4 Variable smoothing length

In order to maintain approximately constant the number of neighbours n_i of each particle i, the smoothing length h_i is considered variable in space and time. According to Benz [1990] the following relation holds:

$$h_i = h_0 \left(\frac{n_0}{n_i} \right)^{1/d_m} \tag{5.21}$$

where d_m is the number of spatial dimensions (2 in 2-D and 3 in 3-D), h_0 and n_0 are the initial smoothing length and initial number of neighbours for a particle i. Equation (5.21) is implicit because the number of neighbours n_i depends on h_i. A Newton-Raphson iterative procedure is used for updating the smoothing length similar to the one presented in chapter 2:

Defining the residual as $R(h_i) = n_0 - n_i(h_i)$ the Newton-Raphson formula for updating h_i is:

$$h_i^{k+1} = h_i^k - \frac{R(h_i^k)}{\left[\frac{dR}{dh}\right]_i^k} \tag{5.22}$$

where the superscript k indicates the iteration number.
Using Equation(5.21) for computing dR/dh we can obtain the following formula for updating the smoothing length:

$$h_i^{k+1} = h_i^k \left(1 - \frac{n_i^k - n_0}{2n_i^k}\right); \tag{5.23}$$

This iteration is stopped when $\|n_i^{k+1} - n_i^k\| < n_g$ where n_g is an integer number much smaller then n_0.

5.3.5 Free-surface boundary condition

In order to impose a Dirichlet boundary condition on the free surface when the Poisson Equation (5.6) is solved, we have to detect particles that are on the free surface.
In the incompressible SPH formulation the free surface is detected by checking either the density (Shao and Lo [2003]) or the divergence of particles' position. These techniques are based on a kernel interpolation and they are not able to detect all the free surface particles (see Lee et al. [2008]). In FPM these approaches cannot be used because the kernel truncation near the surface is no longer present, therefore a pure geometrical approach similar to the one used in the Particle Finite Element Method (Oñate et al. [2006]) is adopted.

As shown in the example plotted in Figure 5.1 each couple of neighbouring points in the domain is considered and the equation of the two circles of radius h_0 from two neighbouring points is calculated; if at least one of them is empty then the two points are on the free surface.

5.3.6 Time step

Since the numerical scheme is essentially explicit, some restrictions are necessary for the stability, in particular the time step dt should satisfy a Courant-Friedrichs-

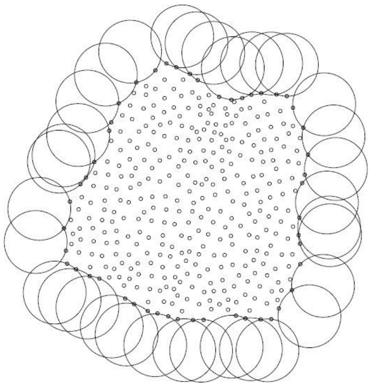


Figure 5.1. example of free surface detection

Lewy condition:

$$dt \leq \min_{\forall_i} \left[0.15 \frac{h_i}{\|\mathbf{v}_i\|} \right] \tag{5.24}$$

a constraint due to viscous diffusion:

$$dt \leq \min_{\forall_i} 0.125 \frac{h_i^2}{\nu} \tag{5.25}$$

and an additional constraint due to body force:

$$dt \leq \min_{\forall_i} 0.04 \sqrt{\frac{h_i}{g}} \tag{5.26}$$

5.4 Test Cases

5.4.1 Stationary fluid in a box

One of the key features of free-surface flow numerical models is the capability to reproduce hydrostatic pressure distribution and to maintain it in time. Because of the difficulties in enforcing boundary conditions in meshless SPH-like models this is a challenging test case.

The test case consists of a box with 3 sides 1 m long filled with still water and immersed in the gravity field, the kinematic viscosity ν is set equal to 0.001 m^2/s. The test case is performed using both the Projection Method and the Incremental Pressure Projection Method. Figure 5.2 shows the particles position obtained after 1 s of simulation, and Figure 5.3 shows the pressure obtained at the same time step at $x = 0.5$ compared with the analytical solution. Only the IPPM is able to reproduce the hydrostatic pressure with a satisfactory approximation and to keep the particles at rest during the simulation.

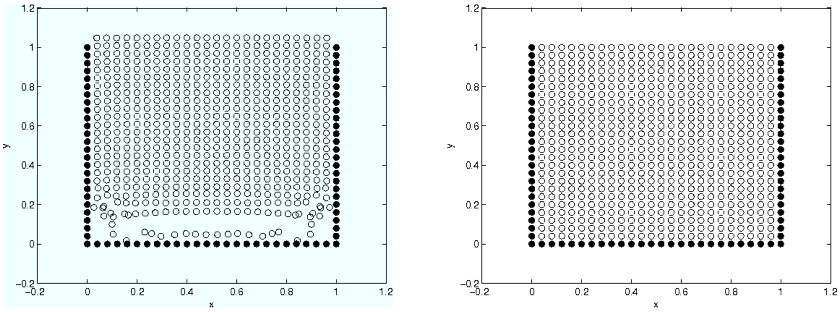


Figure 5.2. Stationary fluid in a box: particle displacement with PM (left) and IPPM (right) after 1 s of simulation

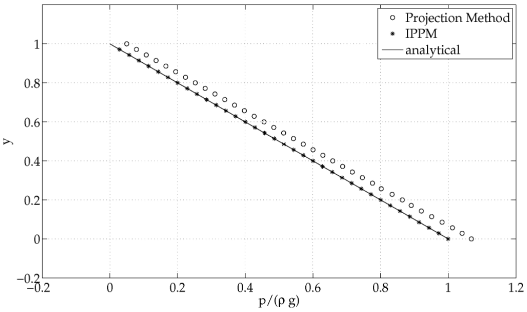


Figure 5.3. Stationary fluid in a box: comparison of adimensional pressure obtained with PM and IPPM method after 1s of simulation at the vertical section x=0.5 m

5.4.2 Taylor-Green flow

In order to assess the convergence rate of the numerical model a Taylor-Green flow with a Reynold number $Re = 1000$ is simulated considering three differents particle resolutions: 20×20, 40×40 and 80×80. The flow is defined by the following analytical solution (Chorin [1968]):

$$v_x(x,y,t) = -u_0 cos\,(\omega x)\, sin\,(\omega y)\, e^{-\frac{8\pi^2\nu}{L^2}t}$$

$$v_y(x,y,t) = u_0 sin\,(\omega x)\,cos\,(\omega y)\,e^{-\frac{8\pi^2\nu}{L^2}t}$$

$$p(x,y,t) = -\frac{1}{4}\,[cos\,(\omega x) + cos\,(\omega y)]\,e^{-\frac{8\pi^2\nu}{L^2}t}$$

where $\omega = \frac{2\pi}{L}$, u_0 is the initial reference velocity and L is the dimension of the periodic domain considered.

In classical Lagrangian meshless method such as SPH, satisfactory results for Taylor-Green test case are difficult to obtain because the particles are compressed at one direction, and stretched at the other an this causes instability (Xu et al. [2009]). In FPM this problem is overcome thanks to the particles management procedure: the number of particles is almost constant in every part of the domain (see for example figure 5.4). Figures (5.5) and (5.6) show that numerical results of the velocities v_x, v_y are almost identical to the analytical solutions.

In order to assess the convergence rate of the method a non-dimensional L_2 norm is calculated for every particle resolutions:

$$L_2(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\frac{f_n^i - f_a^i}{f_0}\right]}$$

where f_a and f_n are the analytical and numerical solutions, f_0 is a reference value equal to u_0 for the velocities and to 1 Pa for the pressure. In Figure (5.7) the non-dimensional L_2 norm of velocities and pressure is plotted at time 2 s; the algorithm has a convergence rate which is more than first order and less than second one.

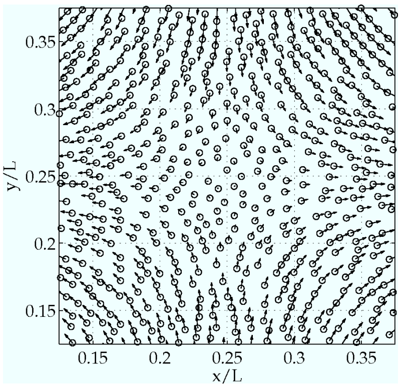


Figure 5.4. Taylor-Green flow with Re=1000: particles position and velocity field for a particle resolution of 80 × 80 at time 2 s

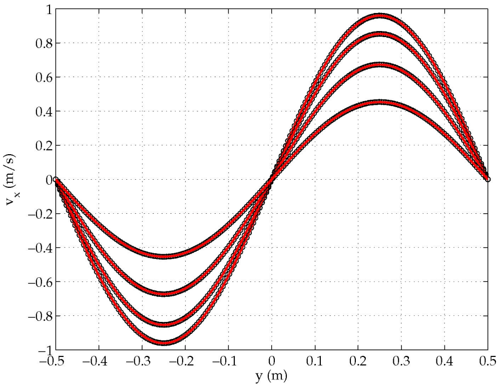


Figure 5.5. Taylor-Green flow with Re=1000: comparison of analytical (red lines) and numerical (dots) v_x at $y = 0$ for a particle resolution of 80 × 80 at times 0.5,2,5,10 s

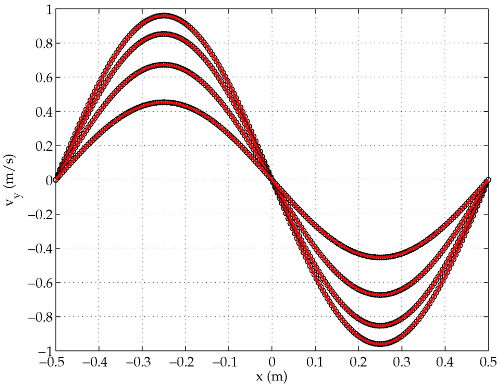


Figure 5.6. Taylor-Green flow with Re=1000: comparison of analytical (red lines) and numerical (dots) v_y at $x = 0$ for a particle resolution of 80 × 80 at times 0.5,2,5,10 s

5.4.3 Evolution of intially circular water bubble

In order to test the capability to reproduce free-surface flow the FPM is tested against the reference solution of the evolution of an initially circular water bubble (Monaghan [1994], Bonet and Lok [1999], and Ferrari et al. [2009]). At the initial time particles are disposed in a circle of radius R with the following velocity and

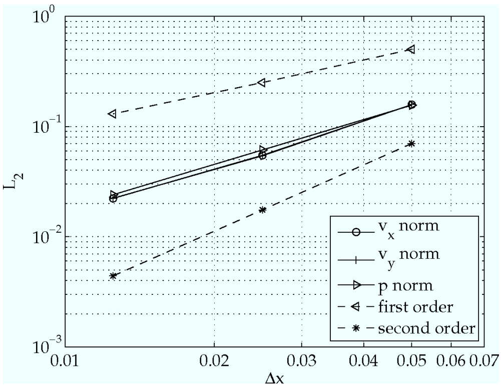


Figure 5.7. Taylor-Green flow with Re=1000: variation of L_2 error with particle spacing at time 2s

pressure field

$$\begin{aligned} v_x(x,y) &= -A_0 x \quad v_y(x,y) = A_0 y \\ p(x,y) &= 0.5\rho A_0^2 \left[R^2 - (x^2+y^2)\right] \end{aligned} \tag{5.27}$$

where ρ is the fluid density, A_0^{-1} is a reference time, and the origin of the reference system is in the centre of the circle. No gravity force and no viscosity is considered.

Because of the incompressibility constraint in this test case the fluid domain remain elliptical during the motion (Colagrossi and Landrini [2003]). It is possible to demonstrate that, under this assumption and considering the initial conditions of (5.27), solving Navier-Stokes equation for a inviscid fluid ($\nu = 0$) is equivalent to solving the following Ordinary Differential Equation:

$$\frac{d^2A}{dt^2} - 4\frac{dA}{dt} + 2A^4 = 0 \tag{5.28}$$

with initial condition $A(0) = A_0$ and $dA/dt = 0$. Defining a and b as the two semi-axes of the ellipse, A is

$$A = -\frac{1}{a}\frac{da}{dt} = \frac{1}{b}\frac{db}{dt}$$

Equation (5.28) is solved by means of a fourth order Runge-Kutta method. Once known $A(t)$, $a(t)$ and $b(t)$ the reference solutions for the pressure and velocity fields are

$$v_x(x, y, t) = -A(t)x \quad v_y(x, y, t) = A(t)y$$

$$p(x, y, t) = 0.5\rho \left[\frac{dA}{dt}(x^2 + y^2) - A^2(x^2 + y^2) \right. \\ \left. -a^2\left(\frac{dA}{dt} - A^2\right) \right] \tag{5.29}$$

The drop evolution has been simulated by FPM up to non-dimensional time $t = 2/A_0$ in order to test the capability of the numerical model to reproduce free-surface flow with a large deformation. The domain is discretized using 2907 particles at the beginning of the simulation. Figure 5.8 shows the comparison between the numerical and reference solutions of ellipse evolution at different times, in the same figure the pressure field is also plotted. In Figures 5.9 and 5.10 the pressure and the velocity along the axis of the ellipse is plotted against the reference solution. The pressure field and the free surface positions at different times are in agreement with the reference solution. Contrary to what was reported in Colagrossi and Landrini [2003] no spurious pressure oscillation is observed: this is due to the pure incompressible approach used in FPM.

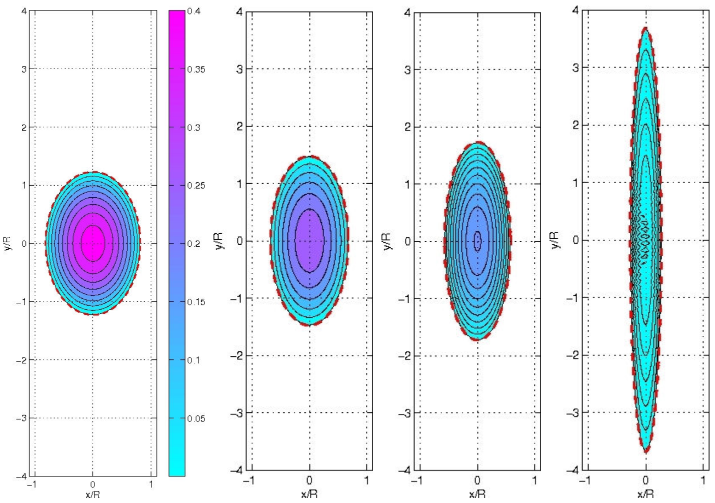


Figure 5.8. Evolution of an initially circular fluid patch: FPM solution at times $tA_0 = 0.2, 0.4, 0.6, 2.0$ s. The colour contours represent the non dimensional pressure field $p/\rho A_0^2 R^2$, the dashed red line is the ellipse calculated by the reference solution

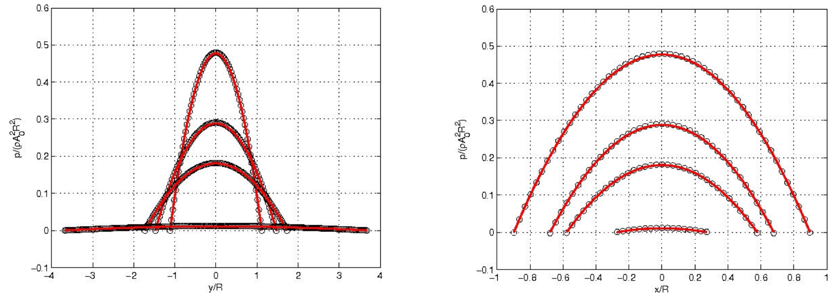


Figure 5.9. Non-dimensional pressure $p/(\rho A_0^2 R^2)$ of evolution of an initially circular fluid: patch: comparison between reference solution (dashed line) and FPM (dots) at x=0 (above) and y=0 (below)

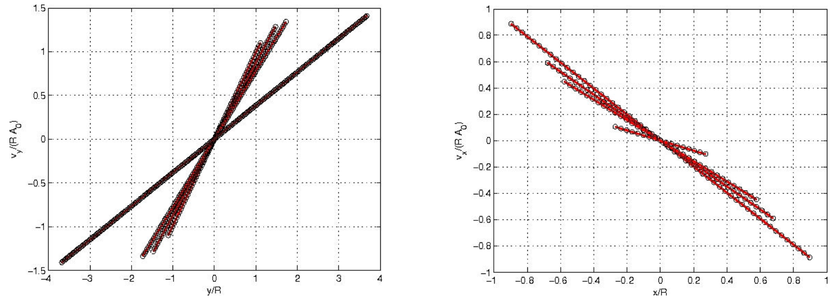


Figure 5.10. Non-dimensional velocities of evolution of an initially circular fluid patch: comparison between reference solution (dashed line) and FPM (dots); $v_y/(A_0 R)$ at x=0 (above) and $v_x/(A_0 R)$ at y=0 (below)

5.4.4 Initial stages of a dam break flow

The numerical results of the initial stages of a 2D dam break flow are compared with a reference solution. We consider a rectangular box 0.40 m long, where a gate is positioned at x=0.25 m separating water at different level is instantly removed at the initial time. Depth ratios of 0.1 and 0.45 are investigated for a larger depth of 0.10 m in the left part of the box. The reference solution is obtained by means of an highly accurate, non-linear, potential-flow model (Stansby et al. [1998]). Figures 5.11 and 5.12 show the comparison of the free surface between the FPM and the reference solutions at different times for the two depth ratios considered. The FPM is able to reproduce the mushroom-like jet and the free surface is in good agreement with the reference solution.

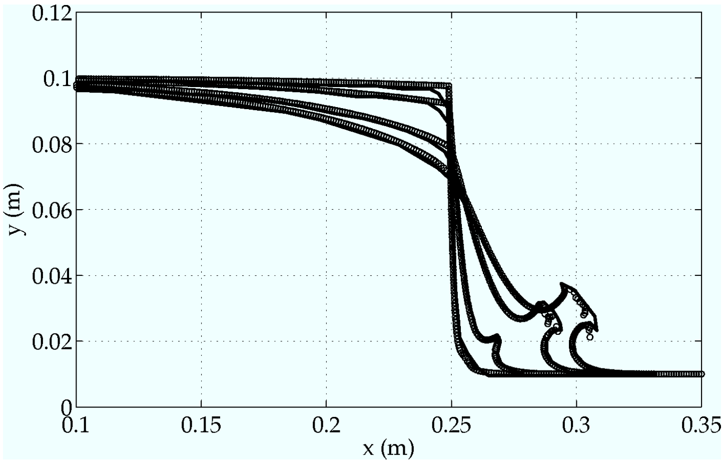


Figure 5.11. Initial stages of a dam break flow with initial water depth ratio of 0.1: profiles of free surface elevation obtained at successive times of 0.024, 0.04, 0.066 and 0.08 s. The FPM (dots) and the reference solution (continuous line) are plotted

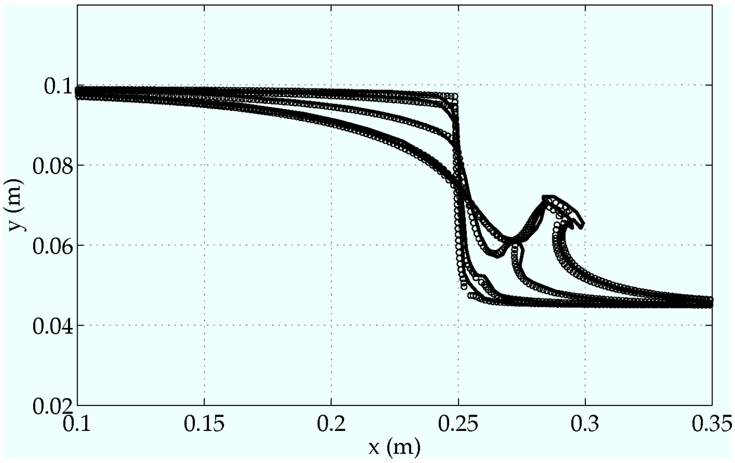


Figure 5.12. Initial stages of a dam break flow with initial water depth ratio of 0.45: profiles of free surface elevation obtained at successive times of 0.02, 0.03, 0.052 and 0.076 and 0.08 s. The FPM (dots) and the reference solution (continuous line) are plotted

5.5 Concluding Remarks

In this chapter a Finite PointSet Method for the numerical simulation of viscous incompressible flow in presence of a free surface has been developed. The analytical enforcement of boundary conditions and the particle management procedure are the most attractive features of FPM. In order to insert a free-surface boundary condition some modifications to the classical form of FPM are proposed: a new technique for free surface detection based on a pure geometrical approach is introduced, and a modified form of Chorin's projection method called Incremental Pressure Projection Method is used with the aim of preserving the hydrostatic condition in time. The algorithm to insert and remove particles together with variable smoothing length have led to good agreement of numerical results with analytical or reference solutions of flow with large distortions, as the evolution of initially circular water bubble and the Taylor-Green. Finally the initial stages of a 2D dam break flows have been simulated.

Chapter 6

Conclusions

This dissertation involved the numerical simulations of rapidly varying open channel flows by Lagrangian meshless numerical methods.

In the first part of this work a Smoothed Particle Hydrodynamics numerical scheme for the discretization of the Shallow Water equation has been developed. This method presents some attractive features in comparison with classical Eulerian methods because no mesh and no special treatment of wet/dry interface are needed. Conversely, one of the main limitations of the numerical scheme is the lack of resolution in zones with a reduced water depth; this has been overcome in this work by introducing a particle splitting procedure: if one particle has an area which is more than a fixed value it is divided into seven daughter particles. The masses, velocities and water depth of daughter particles are assigned by conserving both the mass and momentum. A new procedure has been proposed that significantly reduces the error due to particle splitting when using variable smoothing lengths.

The capability of the method to deal with shock waves has been improved by removing artificial viscosity and considering particle interactions as a Riemann problem. Moreover non-upwind MUSCL reconstruction procedures have been introduced with the aim of obtaining a good representation of rarefaction waves.

In order to extend the method to real case problems, other improvements have been made: closed boundary conditions have been simulated by the Virtual Boundary Particle method, modified to restore almost zero consistency, even in presence of highly variable smoothing length and the slope source term is calculated by means of a SPH interpolation method which can be applied for any bathymetry; finally the possibility to use open boundary conditions have been inserted.

The code has been tested against different 1-D and 2-D dam break problems showing satisfactory results. Thacker's 2-D analytical solution has been also simulated showing the capability of the method to reproduce accurate results with moving 2-D wetting and drying boundaries on non-flat topographies. With the aim of validating the open boundaries algorithm, the 1-D flow over a bump has been reproduced, imposing both subcritical and supercritical inflow/outflow conditions. The last test case simulated is the CADAM where a dam break flow occurs along an initially dry channel; the step located at the inflow and the bend positioned along the channel create a quite complicated geometry representative of a real bathymetry; nevertheless, the numerical model is able to simulate the phenomenon in a satisfactory way, both with and without the refinement procedure activated.

Finally, the simulation of two real cases has been made: the first one is the Okushiri tsunami and the second one is the flooding due to a levee breaching at Thamesmead location (United Kindom). To the author's knowledge these are the first simulations of two real events made by an SPH-SWEs models. The results show that this kind of scheme can be successfully adopted in flooding events simulations.

In the last part of this work the hypothesis of Shallow Water is removed and a meshless, Lagrangian model for the integration of Navier-Stokes equations has been developed. The numerical scheme adopted is the Finite Pointset Method (FPM): its most attractive features are the analytical enforcement of boundary conditions and the particle managment procedure. In order to insert a free-surface boundary condition, some modifications to the classical form of FPM are proposed: a new technique for free surface detection, based on a pure geometrical approach is introduced, and a modified form of Chorin's projection method called Incremental Pressure Projection Method is used with the aim of preserving the hydrostatic condition in time. The algorithm to insert and remove particles together with variable smoothing length have led to good agreement of numerical results with analytical or reference solutions of flow with large distortions, as the evolution of initially circular water bubble and the Taylor-Green. Finally the initial stages of a 2D dam break flows have been simulated.

Bibliography

R. Ata and A. Soulaïmani. A stabilized sph method for inviscid shallow water flows. *International Journal for Numerical Methods in Fluids*, 47(2):139–159, 2004.

F. Aureli, A. Maranzoni, and P. Mignosa. Two dimensional modeling of rapidly varying flows by finite volume schemes. In *Proc. of the second international conference of fluvial hydraulics*, pages 837–849, Napoli, Italy, 2004.

F. Aureli, A. Maranzoni, P. Mignosa, and C. Ziveri. A weighted surface-depth gradient method for the numerical integration of the 2d shallow water equations with topography. *Advances in Water Resources*, 31(7):962 – 974, 2008.

D. S. Balsara. Von Neumann stability analysis of Smoothed Particle Hydrodynamics - Suggestions for optimal algorithms. *Journal of Computational Physics*, 121:357–372, 1995.

D. S. Balsara. Higher order godunov schemes for isothermal hydrodynamics. *The Astrophysical Journal*, 420:197–212, 1994.

Mihai Basa, Nathan J. Quinlan, and Martin Lastiwka. Robustness and accuracy of sph formulations for viscous flow. *International Journal for Numerical Methods in Fluids*, 2008. doi: 10.1002/fld.1927.

J. B. Bell, P. Colella, and H. M. Glaz. A second order projection method for the incompressible NavierStokes equations. *Journal of Computational Physics*, 85: 257–283, 1989.

W. Benz. Smooth particle hydrodynamics - a review. In Robert J. Buchler, editor, *Proceedings of the NATO Advanced Research Workshop on The Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects.* Kluwer Academic Publishers, 1990.

A. Bermúdez and M.E. Vázquez. Upwind methods for hyperbolic conservation laws with source terms. *Comput Fluids*, 8:1049–1071, 1994.

J. Bonet and T. S. L. Lok. Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in applied mechanics and engineering*, 180:97–115, 1999.

David L. Brown, Ricardo Cortez, and Michael L. Minion. Accurate projection methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, 168(2):464 – 499, 2001.

S.-H. Cha and A. P. Whitworth. Implementations and tests of godunov-type particle hydrodynamics. *Monthly Notice of the Royal Astronomical Society*, 340(1):73–90, 2003.

A.J. Chorin. Numerical solution of navier-stokes equations. *Mathematics of Computations*, 22:745–762, 1968.

P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148:227–264, 1999.

A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, 191: 448–475, 2003.

A.J.C. Crespo, M. Gómez-Gesteria, and R.A. Dalrymple. 3d sph simulation of large waves mitigation with a dike. *Journal of Hydraulic Research*, 45(6): 631–642, 2007.

M. de Leffe, D. Le Touzé, and B. Alessandrini. Coastal flow simulations using an sph formulation modelling the non-linear shallow water equations. In *Proc. 3th ERCOFTAC SPHERIC workshop on SPH applications*, pages 48–54, Lausanne, Switzerland, 2008.

M. de Leffe, D. Le Touzé, and B. Alessandrini. Normal flux method at the boundary for sph. In *Proc. IVth ERCOFTAC SPHERIC workshop*, pages 150–157, Nantes, France, 2009.

A. I. Delis, M. Kazolea, and N. A. Kampanis. robust high-resolution finite volume scheme for the simulation of long waves over complex domains. *International Journal for Numerical Methods in Fluids*, 56(4):419–452, 2008.

Gary A. Dilts. Moving-least-squares-particle hydrodynamics - i. consistency and stability. *International Journal for Numerical Methods in Engineering*, 44(8): 1115–1155, 1999.

Michael G. Edwards. The dominant wave-capturing flux: A finite-volume scheme without decomposition for systems of hyperbolic conservation laws. *Journal of Computational Physics*, 218(1):275 – 294, 2006.

J. Fang and A. Parriaux. A regularized lagrangian finite point method for the simulation of incompressible viscous flows. *Journal of Computational Physics*, 227(20):8894 – 8908, 2008.

J. Feldman and J. Bonet. International journal for numerical methods in engineering. *International Journal for Numerical Methods in Fluids*, 72(3):295–324, 2007.

A. Ferrari, M. Dumbser, E. F. Toro, and A. Armanini. A new 3d parallel sph scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.

Masayuki Fujihara and Alistair G. L. Borthwick. Godunov-type solution of curvilinear shallow-water equations. *Journal of Hydraulic Engineering*, 126(11): 827–836, 2000.

P. García-Navarro and M.E. Vázquez-Cendón. On numerical treatment of the source terms in the shallow water equations. *Journal of Computational Physics*, 29:951–979, 2000.

M. B. Giles. Nonreflecting boundary conditions for Euler equation calculations. *AIAA Journal*, 28:2050–2058, December 1990.

R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamic: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

N. Goutal and F. Maurel. In *Proceedings of the 2nd Workshop on Dam-Break Wave Simulation HE 43/97/016/B, Department Laboratoire National d'Hydraulique, Groupe Hydraulique Fluviale Electricité de France, France*, Commission Europenne, Bruxelles, 1997.

L. Hernquist and N. Katz. Treesph- a unification of sph with the hierarchical tree method. *The Astrophysical Journal*, 70:419–446, 1989.

S.E. Hieber and P. Koumoutsakos. An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers. *Journal of Computational Physics*, 227(19):8636 – 8654, 2008.

X.Y. Hu and N.A. Adams. An incompressible multi-phase sph method. *Journal of Computational Physics*, 227(1):264 – 278, 2007.

M.E. Hubbard and P. García-Navarro. Flux difference splitting and the balancing of source terms and flux gradients. *Journal of Computational Physics*, 165(1): 89–125, 2000.

S.-I. Inutsuka. Reformulation of smoothed particle hydrodynamics with riemann solver. *Journal of Computational Physics*, 179:238–267, 2002.

Y. Jang, J.-C . Marongiu, and E. Parkinson. Analysis of sph and mesh based simulations using point based post processing tool. In *Proc. 3th ERCOFTAC SPHERIC workshop on SPH applications*, pages 198–202, Lausanne, Switzerland, 2008.

J. Kim and P. Moin. Application of fractional-step method to incompressible navier-stokes equations. *Journal of Computational Physics*, 59:308–323, 1985.

S. Kitsionas and A. P. Whitworth. Smoothed particle hydrodynamics with particle splitting, applied to self-gravitating collapse. *Monthly Notices of the Royal Astronomical Society*, 330(129), 2002.

S. Kulasegaram, J. Bonet, R. W. Lewis, and M. Profit. A variational formulation based contact algorithm for rigid boundaries in two-dimensional sph applications. *Computational Mechanics*, 33(4):316–325, 2004.

M. Lastiwka and N. Quinland. Permeable and non-reflecting boundary conditions in sph. *International Journal for Numerical Methods in Fluids*, 61(7):709–724, 2008.

M. Lastiwka, N. Quinland, and Basa M. Adaptive particle distribution for smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 47(10–11):1403–1409, 2005.

E.-S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method. *Journal of Computational Physics*, 227(18): 8417 – 8436, 2008.

R. J. LeVeque and D. L. George. High-resolution finite volume methods for the shallow water equations with bathymetry and dry states. In H. Yeh, P.L. Lui, and C.E. Synolakis, editors, *Advanced Numerical Models for simulating Tsunami Waves and Runup*, Advances in Coastal and Ocean Engineering. Word Scientific Publishing Co., Nantes, France, 2007.

Shaofan Li and Wing K. Liu. *Smoothed Particle Hydrodynamics*. World Scientific Publishing, Singapore, 2003.

Qiuhua Liang, Guozhi Du, Jim W. Hall, and Alistair G. L. Borthwick. Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *Journal of Hydraulic Engineering*, 134(11):1603–1610, 2008.

Y. Loukili and A Soulaïmani. Numerical tracking of shallow water waves by the unstructured finite volume waf approximation. *International Journal for Computational Methods in Engineering Science and Mechanics*, 8(2):75–88, 2007.

Leon B. Lucy. A numerical approach to testing the fission hypothesis. *The Astronomical Journal*, 82(12):1013–1924, 1977.

J.B. Marion and S.T. Thornton. *Classical Dynamics fo particles and systems*. Harcourt Brace Jovanovich Inc., 1988.

J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759, 2005.

J. J. Monaghan. An introduction to sph. *Computer Physics Communications*, 48:89–96, 1988.

J. J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–406, 1994.

J. J. Monaghan. Smoothed particle hydrodynamics. *Annual review of Astronomy and Astrophysics*, 30:543–574, 1992.

J. J. Monaghan, A. Kos, and N. Issa. Fluid motion generated by impact. *Journal of waterway, port, coastal and ocean engineering*, 129:250–259, 2003.

R. P. Nelson and J. C. B. Papaloizou. Variable smoothing lengths and energy conservation in smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 270(1), 1994.

I.K. Nikolos and A.I. Delis. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. *Computer Methods in Applied Mechanics and Engineering*, 198(47-48):3723 – 3750, 2009.

E. Oñate, M.A. Celigueta, and S.R. Idelsohn. Modeling bed erosion in free surface flows by the particle finite element method. *Acta Geotechnica*, 1:237–252, 2006.

G. Ramos-Becerra, C. Moulinec, D. R. Emerson, and X. J Gu. Inlet-outlet boundary conditions and truly incompressible sph. In *Proc. 4th ERCOFTAC SPHERIC workshop*, pages 226–231, Nantes, France, 2009.

P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in applied mechanics and engineering*, 139:375–408, 1996.

A. Ritter. Die fortplanzung der wasserwellen. *Zeitschrift des Vereines Deutscher Ingenieure*, 36:947–954, 1892.

M. Rodriguez-Paz and J. Bonet. A corrected smooth particle hydrodynamics formulation of the shallow-water equations. *Computers & Structures*, 17–18: 1396–1410, 2005.

B.D. Rogers, M. Fujihara, and A. G. L. Borthwick. Adptive q-tree godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids*, 35(3):247–280, 2001.

B.D. Rogers, A. G. L. Borthwick, and P. H. Taylor. Mathematical balancing of flux gradient and source terms prior to using roes approximate riemann solver. *Journal of Computational Physics*, 192(2):422–451, 2003.

Hans F. Schwaiger. An implicit corrected sph formulation for thermal diffusion with linear free surface boundary conditions. *International Journal for Numerical Methods in Engineering*, 75(6):647–671, 2008.

Songdong Shao and Edmond Y. M. Lo. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. *Advances in Water Resources*, 26(7):787 – 800, 2003.

S. Soares Frazão, X. Sillen, and Y. Zech. Dam-break flow through sharp bends physical model and 2d boltzmann model validation. In *Proc. of the 6. Summary and conclusions CADAM meeting, Wallingford, United Kingdom, 2 and 3 March 1998*, pages 151–169, Commission Europenne, Bruxelles, 1998.

V. Springel and L. Hernquist. Cosmological smoothed particle hydrodynamics simulations: the entropy equation. *Monthly Notices of the Royal Astronomical Society*, 333:649–664, 2002.

P. K. Stansby, A. Chegini, and T. C. D. Barnes. The initial stages of dam-break flow. *Journal of Fluid Mechanics*, 374(1):407–424, 1998.

W. C. Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, pages 499–508, 1981.

S Tiwari and J. Kuhnert. Finite pointset method based on the projection method for simulations of the incompressible navier-stokes equations. *Lecture Notes in Computational Science and Engineering*, 26:373–388, 2003.

S Tiwari and J. Kuhnert. Modeling of two-phase flows with surface tension by finite pointset method fpm least-squares approximation. *Journal of Computational and Applied Mathematics*, 203:376–386, 2007.

E. F. Toro. Direct riemann solvers for the time-dependent euler equations. *Shock Waves*, 5(1–2):75–80, 1995.

E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics.* Springer, Berlin., 1997.

E.F. Toro. *Shock Capturing Methods for Free Surface Shallow Water Flows.* Wiley: New York, 1999.

H. A. van der Vorst. Bi-cgstab: a fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631–644, 1992.

M. E. Vázquez-Cendón. An extension of the q-scheme of van leer for the shallow water equations using unstructured meshes. In *Proc. of the 6. Summary and conclusions CADAM meeting, Wallingford, United Kingdom, 2 and 3 March 1998*, pages 479–487, Commission Europenne, Bruxelles, 1998.

M.E. Vázquez-Cendón. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics*, 148(2):497–526, 1999.

J. P. Vila. On particle weighted methods and Smoothed Particle Hydrodynamics. *Mathematical models and Methods in Applied Sciences*, 2000.

J. P. Vila. Sph renormalized hybrid methods for conservation laws: Applications to free surface flows. *Lecture Notes in Computational Science and Engineering - Meshfree Methods for Partial Differential Equations II*, 43:207–229, 2005.

Rui Xu, Peter Stansby, and Dominique Laurence. Accuracy and stability in incompressible sph (isph) based on the projection method and a new approach. *Journal of Computational Physics*, 228(18):6703 – 6725, 2009.

J. G. Zhou, D. M. Causon, C. G. Mingham, and D. M. Ingram. The surface gradient method for the treatment of source terms in the shallow-water equations. *Journal of Computational Physics*, 168(1):1 – 25, 2001a.

J.G. Zhou, D.M. Causon, C.G. Mingham, and D.M. Ingram. The surface gradient method for the treatment of source terms in the shallow-water equations. *Journal of Computational Physics*, 168(1):1–25, 2001b.

Jian G. Zhou, Derek M. Causon, Clive G. Mingham, and David M. Ingram. Numerical prediction of dam-break flows in general geometries with complex bed topography. *Journal of Hydraulic Engineering*, 130(4):332–340, 2004.