Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2016) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

1st International Conference on the Material Point Method, MPM 2017

A numerical wave tank using a hybrid particle-mesh approach

Jakob Maljaars[a,*], Robert J. Labeur[a], Matthias Möller[b], Wim Uijttewaal[a]

[a] *Environmental Fluid Mechanics Section, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, 2628 CN Delft, The Netherlands*

[b] *Department of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*

Abstract

In this work the feasibility of a numerical wave tank using a hybrid particle-mesh method is investigated. Based on the Fluid Implicit Particle Method (FLIP) a generic formulation for the hybrid method is presented for incompressible multi-phase flows involving large density jumps and wave generating boundaries. The performance of the method is assessed for a standing wave, and the generation and propagation of a solitary wave over a flat and a sloping bed. These benchmark tests demonstrate that the method is a promising and attractive tool for simulating the nearshore propagation of waves.

Peer-review under responsibility of the organizing committee of the 1 st International Conference on the Material Point Method.

Keywords: Lagrangian-Eulerian; particle-mesh; free surface flows; numerical wave flume; Navier-Stokes.

1. Introduction

The numerical modeling of flow problems in environmental fluid mechanics has made huge progress over the past decades. While Eulerian mesh-based methods are widely used for solving the governing Navier-Stokes equations, Lagrangian particle-based methods such as the Smoothed Particle Hydrodynamics (SPH) method are becoming increasingly popular for problems involving violent free surface motion (see e.g. [1], [2]). While interface tracking in Eulerian methods requires additional effort, interfaces are trivially captured by Lagrangian particle-based methods. However, Lagrangian particle-based methods are known to be computationally much more demanding than Eulerian mesh-based methods, on top of that, enforcing the incompressibility constraint in particle-based methods is far from being trivial. Therefore, fluids are most often assumed to be weakly compressible in particle-based methods which, in combination with explicit time stepping, leads to restrictively small time steps when applied to (nearly) incompressible fluids.

This work outlines a hybrid Lagrangian-Eulerian (particle-mesh) method for simulating incompressible fluid flows by presenting a generic formulation for the particle-mesh interaction and a strategy for efficiently solving the (Navier-Stokes) equations on the background mesh. Building upon Fluid Implicit Particle (FLIP, [3]) and Particle in Cell (PIC, [4]) concepts, the method has the combined advantages of Eulerian mesh-based and Lagrangian particle-based

* Corresponding author.
E-mail address: j.m.maljaars@tudelft.nl

1877-7058
Peer-review under responsibility of the organizing committee of the 1 st International Conference on the Material Point Method.

methods. For the proposed particle-mesh interaction procedure, the method is more geared towards a mesh-based method than e.g. the material point method (MPM, [5]) and SPH.
More specifically, this work aims at developing a hybrid particle-mesh numerical wave tank, where both the air- and the water-phase are modeled by means of Lagrangian particles. Therefore, several issues to achieve this objective are addressed in this work: particularly, the implementation of wave-making boundary conditions and a method to deal with sharp discontinuities in material properties are presented.
The remainder of this work is structured as follows: in Section 2 the governing equations are presented, Section 3 discusses the model formulation. In Section 4 the model performance is assessed by running different benchmark tests. In Section 5 conclusions are drawn and an outlook for future research is given.

2. Governing equations

The motion of an incompressible fluid in a domain $\Omega \subset \mathbb{R}^d$, with d the spatial dimension, and time interval $I = (t^0, t^N)$ is described by the incompressible Navier-Stokes equations, comprising both the momentum balance and the incompressibility constraint. For Newtonian fluids these equations respectively read as follows:

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = \mathbf{f} - \nabla p + \nabla \cdot \boldsymbol{\sigma} \qquad \text{in} \quad \Omega \times I \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in} \quad \Omega \times I \tag{2}$$

in which $\mathbf{u}$ denotes the flow velocity, ρ the density, p the pressure, $\mathbf{f}$ the body force and the viscous stress tensor is given by $\boldsymbol{\sigma} = 2\mu\nabla^s\mathbf{u}$, with $\nabla^s(\cdot) = \frac{1}{2}\nabla(\cdot) + \frac{1}{2}\nabla(\cdot)^T$ being the symmetric gradient operator and μ the dynamic viscosity. In the framework of the hybrid particle-mesh methods, Eqs. (1)-(2) represent the equations to be solved at the background mesh.

3. Model formulation

A hybrid particle-mesh method typically consists of the following steps:

1. *Particle-mesh projection*; data from the scattered particles is projected onto the background mesh in order to reconstruct continuous density and velocity fields.
2. *Solving the mesh equations*; the continuum equations (Eqs. (1)-(2)), are solved at the Eulerian background mesh.
3. *Mesh-particle projection*; given the solution obtained in step 2, particle properties are updated. For immiscible, incompressible fluids this yields an update of the particle velocity (being representative for the change in particle momentum).
4. *Particle advection*; the particle position is updated by moving the particles with an advective velocity.

Step 2 and 4 are entirely performed at mesh and particle level respectively, while step 1 and 3 take care of the coupling between particles and the background mesh.

3.1. Particle projection to the mesh

Contrary to the common approach in the classical MPM [5], where (mass) integrals are replaced by summations over the discrete set of particles, a least square fitting is proposed for the projection of the particle properties to the mesh, following e.g. [6], [7]. Such an approach results in a smooth representation of the particle properties at the background mesh. Similar techniques have been developed in the MPM community in order to achieve this (see e.g. [8],[9]). An extensive discussion of the least square approach can be found in [6] and [10]. In essence, the continuous density and velocity fields ρ and $\mathbf{u}$ are approximated in finite dimensional function spaces $\mathcal{V}_h \subset L^2(\Omega)$ and $\boldsymbol{\mathcal{V}}_h \subset \mathbf{L}^2(\Omega)$:

$$\rho \approx \rho_h(\mathbf{x}) = \sum_{i=1}^{n_\rho} N_i(\mathbf{x})\rho_i \qquad \text{and} \qquad \mathbf{u} \approx \mathbf{u}_h(\mathbf{x}) = \sum_{i=1}^{n_u} N_i(\mathbf{x})\mathbf{u}_i \tag{3}$$

where $N_i(\mathbf{x})$ are the piecewise polynomial basis functions for the density field and the velocity field defined on the elements of the background mesh. Moreover, ρ_i and $\mathbf{u}_i$ denote the nodal degrees of freedom and the summation runs over the number of nodal density unknowns n_ρ and velocity unknowns n_u respectively.
Given Eq. (3), a least square projection of the particle properties onto the background mesh is obtained by solving:

$$\sum_p N_i(\mathbf{x}_p) N_j(\mathbf{x}_p) \rho_j = \sum_p N_i(\mathbf{x}_p) \rho_p \tag{4}$$

$$\sum_p N_i(\mathbf{x}_p) N_j(\mathbf{x}_p) \mathbf{u}_j = \sum_p N_i(\mathbf{x}_p) \mathbf{u}_p \tag{5}$$

for ρ_j and $\mathbf{u}_j$ the nodal density and velocity unknowns at the background mesh, respectively.
In these equations, ρ_p and $\mathbf{u}_p$ denote the particle density and - velocity of particle p. $N_{i,j}$ are nodal basis functions which are evaluated at the particle position $\mathbf{x}_p$, furthermore the summation runs over the set of particles in Ω.

3.2. Solving the mesh equations

The mesh equations (Eqs. (1)-(2)) are solved using the fractional-step method [11]. In this approach, the equations for the velocity and the pressure are solved sequentially and coupled using an intermediate velocity field. The resulting set of equations is discretized in space using the finite element method, where the velocity is approximated using Eq. (3) and the pressure p is approximated in the finite-dimensional function space $\mathcal{Q}_h \subset H^1(\Omega)$:

$$p \approx p_h(\mathbf{x}) = \sum_{i=1}^{n_\mathrm{p}} \bar{N}_i(\mathbf{x}) p_i \tag{6}$$

with $\bar{N}_i(\mathbf{x})$ the piecewise polyomial basis functions and p_i the nodal degrees of freedom for the pressure. Furthermore, the summation runs over the number of pressure nodal unknowns n_p.
The finite element discretization results in the following set of variational equations to be solved at the background mesh (an extensive derivation of below presented fractional step approach can be found in [10]):
Given $\mathbf{u}_h^n$, find the intermediate velocity $\mathbf{u}_h^* \in \boldsymbol{\mathcal{V}}_h$ such that:

$$\int_\Omega \mathbf{u}_h^* \cdot \mathbf{v}_h \mathrm{d}\Omega = \int_\Omega \mathbf{u}_h^n \cdot \mathbf{v}_h \mathrm{d}\Omega + \Delta t \int_\Omega \mathbf{g} \cdot \mathbf{v}_h \mathrm{d}\Omega \qquad \forall \mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h \tag{7}$$

Given the density field ρ_h^n and the intermediate velocity field $\mathbf{u}_h^*$, find the pressure $p_h^{n+1} \in \mathcal{Q}_h$ such that:

$$\Delta t \int_\Omega \frac{1}{\rho_h^n} \nabla p_h^{n+1} \cdot \nabla q_h \mathrm{d}\Omega = \int_\Omega \mathbf{u}_h^* \cdot \nabla q_h \mathrm{d}\Omega - \oint_{\partial\Omega} \mathbf{h} \cdot \mathbf{n} q_h \mathrm{d}\Gamma \qquad \forall q_h \in \mathcal{Q}_h \tag{8}$$

Given the updated pressure p_h^{n+1}: find $\mathbf{u}_h^{n+1} \in \boldsymbol{\mathcal{V}}_h$ such that:

$$\int_\Omega \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h \mathrm{d}\Omega = \int_\Omega \mathbf{u}_h^* \cdot \mathbf{v}_h \mathrm{d}\Omega - \Delta t \int_\Omega \frac{1}{\rho^n} \nabla p_h^{n+1} \cdot \mathbf{v}_h \mathrm{d}\Omega \qquad \forall \mathbf{v}_h \in \boldsymbol{\mathcal{V}}_h \tag{9}$$

In these equations, the time level is denoted by the superscript n with time interval Δt, and the gravitational acceleration $\mathbf{g}$, the boundary term involving the given quantity $\mathbf{h}$ is more extensively discussed in Section 3.5. The viscous term has been omitted although the method allows for including this term. Important to note is that both the velocity $\mathbf{u}_h^n$ and the density ρ_h^n are determined by means of the particle-mesh projection (Section 3.1), and hence, no information is stored at the mesh between consecutive time steps.
Given the saddle point nature of the incompressible Navier-Stokes equations, the velocity space $\boldsymbol{\mathcal{V}}_h$ and pressure space $\mathcal{Q}_h$ must satisfy the inf-sup condition [12] in order to avoid spurious pressure modes. In this work, the stable $\mathrm{P_0P_1}$-element is used, having piecewise constant velocities and piecewise linear pressure approximations. Although the method is not restricted to this particular element, it has the additional advantage that the particle-mesh mapping (Eqs. (4)-(5)) becomes a local operation since piecewise constant basis functions are used for this projection. For efficiently solving Eqs. (7)-(9), the FEniCS finite element package is used [13].

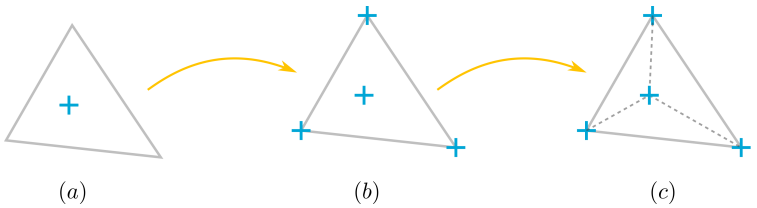

Fig. 1. Mapping procedure: (*a*) cell centered velocity field; (*b*) velocity mapped to vertices; (*c*) division into sub-elements around barycenter.

3.3. Mesh-particle mapping

Solving the set of discrete equations yields a velocity field on the background mesh at the new time step, which is used to update the velocity carried by the particles. This is done by adding the mesh velocity increment to the particles, which is a common approach in the FLIP method. It is however known that a pure FLIP velocity update can develop noisy particle velocity fluctuations [14]. These high-frequency fluctuations are absent in the PIC method, where particle velocities are overwritten by the mesh velocity in every time step. However, this method is known to be diffusive. In order to suppress the high-frequency fluctuations while avoiding the numerical diffusivity, a blended PIC/FLIP update following [14] is proposed. Thus, the particle velocity is updated as follows:

$$\mathbf{u}_p^{n+1} = \alpha \tilde{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^n) + (1-\alpha)\left\{\mathbf{u}_p^n + \Delta\tilde{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^n)\right\} \tag{10}$$

with the subscript p denoting the particle index and the mesh velocity increment given by $\Delta\tilde{\mathbf{u}}_h^{n+1} = \tilde{\mathbf{u}}_h^{n+1} - \tilde{\mathbf{u}}_h^n$, where $\tilde{\mathbf{u}}_h^n(\mathbf{x}_p^n) = \sum_i \tilde{N}_i(\mathbf{x}_p^n)\tilde{\mathbf{u}}_i^n$. Furthermore, the PIC fraction is denoted with $0 \leq \alpha \leq 1$.

Following [15] the interpolation functions $\tilde{N}_i(\mathbf{x})$ are the piecewise linear polynomials associated with the three sub-elements having the barycenter of the element as shared node, with $\tilde{\mathbf{u}}_i^n$ the corresponding nodal velocities, Figure 1.

3.4. Particle advection

At the end of every time step, the particles are advected by solving the ordinary differential equation for each particle:

$$\frac{\mathrm{d}\mathbf{x}_p(t)}{\mathrm{d}t} = \tilde{\mathbf{u}}_p(\mathbf{x}_p, t) \qquad \text{where} \qquad \tilde{\mathbf{u}}_p(\mathbf{x}_p, t) = \sum_i \tilde{N}_i(\mathbf{x}_p)\tilde{\mathbf{u}}_i(t) \tag{11}$$

using the same velocity interpolant $\tilde{N}_i(\mathbf{x})$ as in Section 3.3.

Since this step is responsible for the advection of material properties through the domain, the advection scheme should be accurate. As shown in [16] and [10], a forward Euler integration is not sufficient for this purpose. Instead, a three step Runga-Kutta scheme was used [17].

3.5. Boundary conditions

3.5.1. Wave making boundary

Wavemaking boundaries are of special interest in the context of a numerical wave flume. Recalling the weak form of the Pressure-Poisson equation (Eq. (8)) it is clear that the boundary term $\oint_{\Gamma_N} \mathbf{h} \cdot \mathbf{n} q \mathrm{d}\Gamma$ can be used to enforce the normal velocity, where Γ_N is the part of the domain on which a Neumann condition for the pressure is specified and the boundary velocity $\mathbf{h}$ can be specified according to a wavemaker theory. In order to avoid empty elements (i.e. elements not containing any particles), a moving mesh approach is used. Since no information is stored at the background mesh between consecutive time steps, moving the mesh does not alter the governing equations. At the top boundary of the numerical flume, the pressure is set to zero in order to simulate an open boundary. Correspondingly, the (air) particles are free to move out- and inward through this boundary during the computation.

3.5.2. Interface condition

In principle, the Lagrangian particles naturally track the air-water interface and no additional interface conditions have to be specified (provided surface tension is neglected). However, in two-phase flow problems, discontinuities in the pressure gradient occur due to the density jump at the interface. Ambiguous definition of the density in elements crossed by the interface, may result in spurious oscillations (see, e.g., [18]). To suppress these spurious velocities it is noted that in gravitational equilibrium the pressure gradient term exactly balances the gravitational acceleration, i.e. $\frac{1}{\rho}\nabla p = \mathbf{g}$. So although the pressure gradient itself and the density itself are discontinuous at the interface, the term $\frac{1}{\rho}\nabla p$ is continuous in gravitational equilibrium. Inspired by this observation, the pressure gradient term $\frac{1}{\rho}\nabla p$ in Eq. (9) is projected to a piecewise linear field. In other words, the task is to find $\mathbf{r} \in \mathcal{S}$ such that:

$$\int_\Omega \mathbf{r} \cdot \mathbf{s} \mathrm{d}\Omega = \int_\Omega \frac{1}{\rho} \nabla p \cdot \mathbf{s} \mathrm{d}\Omega \qquad \forall \mathbf{s} \in \mathcal{S} \tag{12}$$

in which the function space $\mathcal{S}_h \subset \mathbf{H}^1(\Omega)$. Consequently, the problem for calculating the end-of-time step velocity (Eq. (9)) becomes: find $\mathbf{u}_h^{n+1} \in \mathcal{V}_h$ such that:

$$\int_\Omega \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h \mathrm{d}\Omega = \int_\Omega \mathbf{u}_h^{*} \cdot \mathbf{v}_h \mathrm{d}\Omega - \Delta t \int_\Omega \mathbf{r}^{n+1} \cdot \mathbf{v}_h \mathrm{d}\Omega \qquad \forall \mathbf{v}_h \in \mathcal{V}_h \tag{13}$$

This approach indeed effectively suppresses the spurious velocities near the interface.

4. Numerical examples

4.1. Standing wave

As a first benchmark test, a standing wave in a closed basin is considered, prescribing the air-water density ratio between the two phases. This test provides useful insight in the amount of numerical diffusion and will indicate whether or not the scheme is able to deal with large density differences between the phases. Model results are compared with the analytical expression for the free surface elevation obtained from linear wave theory. Model settings are listed in Table 1. A small amount of numerical diffusivity is added by blending the FLIP update with a small PIC fraction (see Section 3.3). No detailed investigation of the critical time step was made, therefore a conservative value of $\Delta t = 0.001$ s was used (corresponding to a Courant number of approximately 0.02). It is however noted that this time step size is still typically $O(10^2)$ larger than in weakly compressible approximations (e.g. [19]).

Table 1. Model settings standing wave.

Model parameters	Ω	$[0, 20] \times [0, 10]\mathrm{m}^2$	Computational domain
	nx×ny	80×40	Regular triangular mesh (nx×ny×2 cells)
	Δx_p	0.02 m	Particle spacing
	α	0.02	PIC fraction
	d	5 m	Undisturbed water depth
	H	0.5 m	Wave amplitude
	Δt	0.001 s	Time step
	k	$2\pi/20$	Wave number ($2\pi/L$)
Computational aspects		500 000	Number of particles
		15 s	Simulated physical time
		176 min	Model run time

The simulated air-water interface and the resulting pressure fields for the water-phase are depicted in Figure 2 at two different time instances. The resulting pressure fields are perfectly smooth and no pathological locking effects

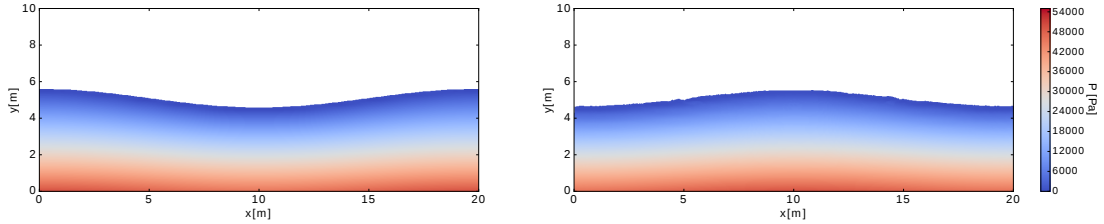

Fig. 2. Pressures in the water phase of a standing wave with $H/d = 0.1$ at $t = 0$ s (left panel) and $t = 5.5$ s (right panel).

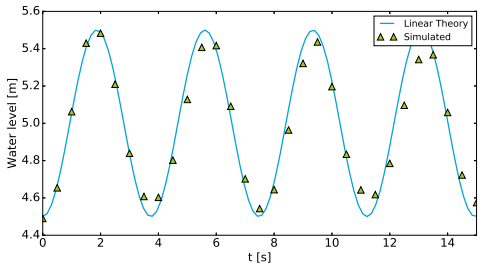

Fig. 3. Theoretical and simulated surface elevation at $x = 0.5L$.

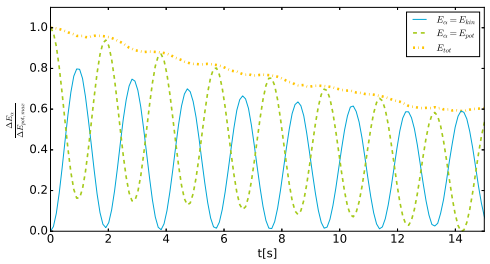

Fig. 4. Total, kinetic and potential energy in standing wave.

are observed. Furthermore, the air-water interface is relatively well-maintained over time, although small free surface disturbances are observed near the inflection points of the cosine wave (right panel of Figure 2).
The simulated amplitudes are compared with the analytical amplitudes at $x = 0.5L = 10$ m in Figure 3. A good correspondence between simulated results and analytical values is found, although a slight amplitude decay is observed. The diffusivity of the scheme is also observed when plotting the potential and the kinetic energy (non-dimensionalized by the potential energy budget), Figure 4. A decay in total particle energy of approximately 40% in four wave periods is observed. This is however smaller than the reported 69% in four wave periods for a comparable SPH simulation [20].

4.2. Propagation of a solitary wave

As a test case to assess the performance of the scheme for large amplitude problems, the generation and propagation of a solitary wave over a flat (Section 4.2.1) and over a sloping bed (Section 4.2.2) is considered. For the propagation over the flat bed, a comparison is made with approximate analytical expressions [19] and results obtained using the open-source DualSPHysics package [21]. Theory behind the forced generation of solitary waves by means of piston wave makers can be found in, e.g., [22], providing the non-homogeneous Neumann boundary condition for the numerical wave flume (Section 3.5).

Table 2. Model properties solitary wave generation.

Model parameters	Ω_0	$[0, 15] \times [0, 0.6]\text{m}^2$	Computational domain
	nx×ny	500×20	Regular triangular mesh (nx×ny×2 cells)
	Δx_p	0.005m	Particle spacing
	α	0.03	PIC fraction
	Δt	0.001 s	Time step
	d	0.3 m	Still water depth
Computational aspects		360000	Number of particles
		10 s	Simulated physical time
		176 min	Model run time

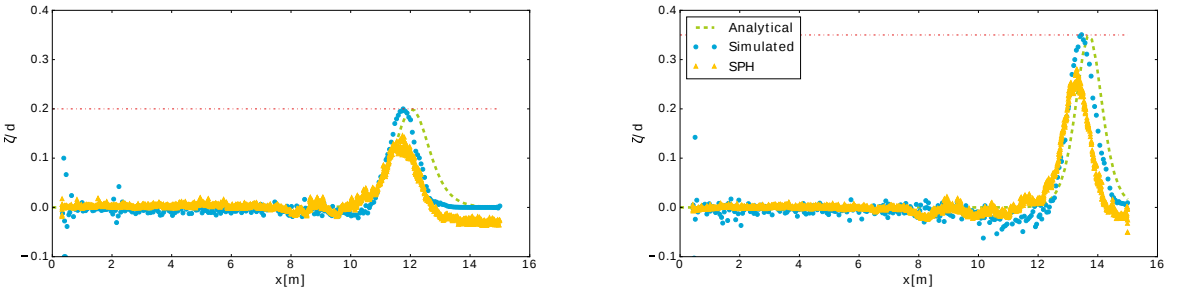

Fig. 5. Comparison between the analytical, the hybrid particle-mesh and the SPH solution for the generated solitary wave at $t = 8$ s. The left panel corresponds to the $H/d = 0.2$ case, the right panel corresponds to the $H/d = 0.35$ case.

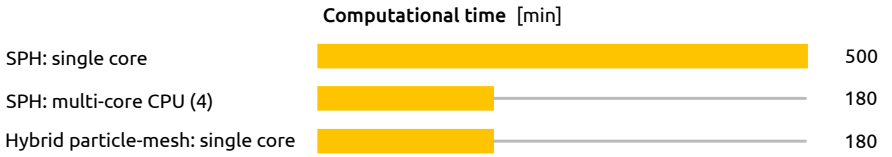

Fig. 6. Approximate computational times for the SPH model in various computing configurations and the developed hybrid particle-mesh method.

4.2.1. Solitary wave propagation over a flat bed

For the test case involving a flat bed, a solitary wave is generated in the domain initially defined by $\Omega = [0, 15] \times [0, 0.6]\text{m}^2$. A time step $\Delta t = 0.001$ s is used, corresponding to a Courant number of approximately 0.02. Other model parameters and computational aspects are listed in Table 2. The model is run for two different dimensionless amplitudes $H/d = 0.2$ and $H/d = 0.35$. Results for both runs are depicted in Figure 5 at $t = 8$ s. A good agreement is observed between simulated and analytical results, although model results are slightly lagging behind the analytical solution, which can be attributed to numerical diffusion. However, this numerical diffusion is much more pronounced for comparable SPH runs using DualSPHysics, Figure 5. In order to mimic the inviscid case in SPH, the artificial viscosity (representing the viscosity term and enhancing numerical stability) is set to 0. Clearly, results obtained with the SPH model (using the same particle spacing) are much more diffusive than results obtained with the hybrid particle-mesh method. In terms of computational times, the developed model is able to compete with the DualSPHysics implementation, as shown in Figure 6: running the hybrid particle-mesh simulation on a single core results in a computational time of approximately 180 min, compared to approximately 500 min for the SPH computation on a single core. The gain in computational time can be partly attributed to the less restrictive time step requirements in the incompressible hybrid particle-mesh method compared to the weakly-compressible SPH model.
Based on the comparison presented above, it can be concluded that the results obtained with the hybrid particle-mesh method are encouraging both in terms of accuracy and computational efficiency.

4.2.2. Solitary wave propagation over a sloping bed

A final example showing the capabilities of the model, is the breaking of a solitary wave on a submerged bar, placed in a domain with a horizontal and vertical base of 13 m × 1 m. An unstructured, boundary-fitted triangular mesh is used for this benchmark, containing approximately 25000 cells. Around 25 particles are initially placed in each cell. In the domain, a solitary wave is generated with a relative amplitude $H/d = 0.625$. A time step of 0.001 s is used.
The process of wave propagation and wave breaking is presented in Figure 7, qualitatively revealing the important nearshore wave processes: wave steepening (shoaling) around $t = 4.5$ s, wave plunging around $t = 5$ s, and wave breaking around $t = 5.5$ s. Although not validated quantitatively, the test reveals the capability of hybrid particle-mesh methods to simulate wave propagation up to and including wave breaking.

5. Conclusions and outlook

In this work, a hybrid particle-mesh approach for incompressible fluid flows was formulated and tested for various free surface benchmark tests. The method was shown to combine various advantages of an Eulerian mesh-based

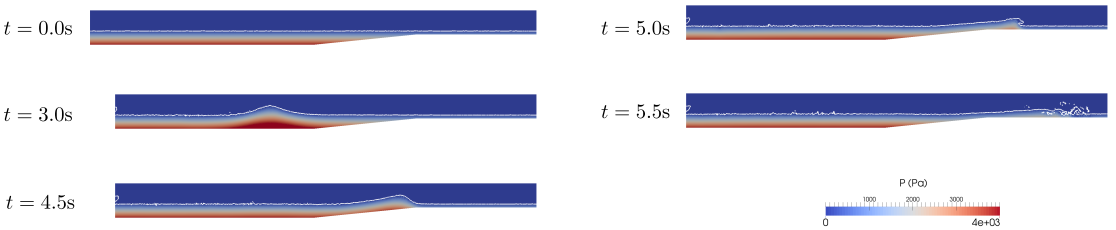

Fig. 7. Soliton propagation including breaking on a submerged bar. Color shading indicates the pressures at the background mesh.

approach (enforcing incompressibility, fast and efficient solvers) with the advantages of a pure Lagrangian particle-based approach (tracking interfaces of complex topography). Due to the implicit treatment of the pressure, smooth and accurate pressure fields are obtained. Additional advantage of this approach compared to the weakly compressible appoach is the mild time step requirement, thus rendering the method an attractive alternative to pure particle-based methods such as SPH when setting up a numerical wave tank.
Future work will focus on numerical aspects of the scheme such as convergence issues, the amount of particles required and, most importantly, the physical and theoretical underpinnings of the method.

Acknowledgements

The NWO (Netherlands Organisation for Scientific Research) is acknowledged for their support through the JMBC-EM Graduate Programme research grant.

References

[1] J. Monaghan, Simulating Free Surface Flows with SPH, Journal of Computational Physics 110 (1994) 399–406.
[2] A. Colagrossi, M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, Journal of Computational Physics 191 (2003) 448–475.
[3] J. Brackbill, H. Ruppel, FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions, Journal of Computational Physics 65 (1986) 314–343.
[4] M. Evans, F. Harlow, E. Bromberg, The particle-in-cell method for hydrodynamic calculations, Technical Report, Los Alamos Scientific Laboratory, 1957.
[5] D. Sulsky, Z. Chen, H. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1994) 179–196.
[6] S. Subramaniam, D. C. Haworth, A probability density function method for turbulent mixing and combustion on three-dimensional unstructured deforming meshes, International Journal of Engine Research 1 (2000) 171–190.
[7] E. Edwards, R. Bridson, Detailed water with coarse grids, ACM Transactions on Graphics 33 (2014) 1–9.
[8] M. Steffen, R. M. Kirby, M. Berzins, Analysis and reduction of quadrature errors in the material point method (MPM), International Journal for Numerical Methods in Engineering 76 (2008) 922–948.
[9] R. Tielen, High order material point method, Master's thesis, Delft University of Technology, 2016.
[10] J. Maljaars, A hybrid particle-mesh method for simulating free surface flows, Master's thesis, Delft University of Technology, 2016.
[11] A. J. Chorin, Numerical solution of the Navier-Stokes equations, Mathematics of Computation 22 (1968) 745–745.
[12] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, volume 15, Springer, 1991.
[13] A. Logg, K.-A. Mardal, G. N. Wells, Automated Solution of Differential Equations by the Finite Element Method, volume 84, Springer, 2012.
[14] R. Bridson, Fluid simulation for computer graphics, CRC Press, 2008.
[15] R. Ando, N. Thürey, C. Wojtan, Highly adaptive liquid simulations on tetrahedral meshes, ACM Transactions on Graphics 32 (2013) 1.
[16] B. A. Wols, CFD in drinking water treatment, Ph.D. thesis, Delft University of Technology, 2010.
[17] A. Ralston, Runge-Kutta Methods with minimum error bounds, Mathematics of Computation 16 (1962) 431–437.
[18] R. Wemmenhove, Numerical simulation of two-phase flow in offshore environments, Ph.D. thesis, Rijksuniveristeit Groningen, 2008.
[19] Z. Wieckowski, Enhancement of the material point method for fluid-structure interaction and erosion, Research project report, Deltares, 2013.
[20] L. de Wit, Smoothed Particle Hydrodynamics. A Study of the possibilities of SPH in hydraulic engineering, Master's thesis, Delft University of Technology, 2006.
[21] A. Crespo, J. Domínguez, B. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, O. García-Feal, DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH), Computer Physics Communications 187 (2015) 204–216.
[22] D. G. Goring, Tsunamis–the propagation of long waves onto a shelf, Ph.D. thesis, California Institute of Technology, 1978.