2D Wave simulation using Moving Particle Implicit method

Kartik NA20B053

1 Introduction

Understanding and accurately predicting the dynamics of water surfaces are essential for designing structures, optimizing coastal defences, and ensuring the safety of maritime activities. In this project, we delve into the realm of numerical methods for fluid simulation, focusing on the Moving Particle Semi-Implicit (MPS) method—a mesh-free approach renowned for its versatility in capturing complex fluid behaviours.

The MPS method is chosen for its ability to model free surfaces and fluid interactions without the constraints of a fixed grid. It employs particles to represent the fluid, enabling the simulation of dynamic free surfaces and fluid flows with greater flexibility than traditional grid-based methods.

2 Governing Equations

Governing equations of viscous fluid including continuity and momentum equations which can be represented as follows:

$$\frac{1}{\rho}\frac{D\rho}{Dt} + \nabla u = 0$$

$$\frac{Du}{Dt} = -\frac{1}{\rho}\nabla p + \nu_t \nabla^2 u + f$$

where u = velocity vector, t= time, = fluid density, P = pressure, vt = turbulent eddy viscosity and f = vector of gravitational acceleration, impact force of particles and surface tension.

In the MPS method, the pressure term can be considered as a dependent variable which contains the static and dynamic parts together. Thus, by solving Poisson equation, the pressure term can be calculated for all particles. Pressure Poisson Equation can be written as follows:

$$\frac{\Delta\rho}{\Delta t^2} = \nabla^2 P$$

3 MPS formulation

In the MPS method the equations of continuity and momentum are converted to interaction equations of particles using different operators. All interactions between particles are limited to a specific distance known as efficient radius (r_e). The weighing of different neighboring particles within efficient radius on the desired particle is calculated based on Kernel functions. In present project, following kernel function is used:

$$n_i = \sum_{j \neq i} w(||r_j^* - r_i^*||)$$

$$\text{w(r)} = \begin{cases} 0 & r_e \leq r \\ \frac{r_e}{r} - 1 & 0 \leq r < r_e \end{cases}$$

where r = distance between two particles, r_e = efficient radius and w = Kernel function.

In MPS, the gradient vector is modelled using the weight function to obtain a gradient vector of the particle i as:

$$< \nabla\phi >_i = \frac{d}{n^0} \Sigma_{j \neq i} [\frac{\phi_j - \hat{\phi}_i}{|r_j - r_i|}(r_j - r_i) w(|r_j - r_i|)]$$

where d is the number of space dimension and $\hat{\phi}_i$ is taken as the minimum value of j within the interaction ratio. This is good for numerical stability as forces between particles are always repulsive as j ˆ i is positive.

Continuing, the Laplacian operator is then defined as:

$$< \nabla^2\phi >_i = \frac{2d}{n^0\lambda} \Sigma_{j \neq i} [(\phi_j - \phi_i) w(|r_j - r_i|)]$$

where is a parameter to ensure that the variance increase is equal to the analytical solution ;

$$\lambda = \frac{\Sigma_{j \neq i} |r_j - r_i|^2 w(|r_j - r_i|)}{\Sigma_{j \neq i} w(|r_j - r_i|)}$$

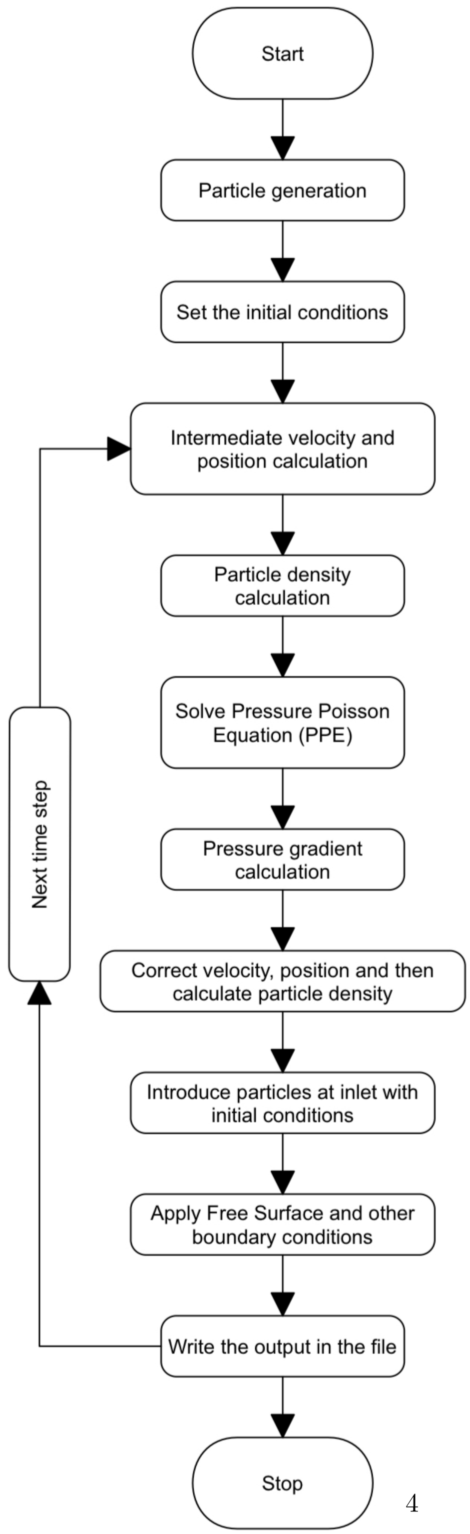
So the equation obtained for the Laplacian operation is used to model and discretise not only the Laplacian of the velocities from the Navier stokes equation but also the Poisson Pressure Equation. The Poisson Pressure Equation with the density approximated by the particle number density is:

$$< \nabla^2 P_i^{k+1} > = \frac{\rho}{\Delta t^2} \frac{n_i^* - n^0}{n^0}$$

where n_i^* is the particle number density at the intermediate position r_i^*, after the Laplacian of the velocity and gravity are computed. The right hand side term of the equation is modeled according to the equation obtained for laplacian operator which generates a system of equations which becomes a matrix. This matrix is sparse and is solved using gaussian elimination in the program. Its derivation is discussed ahead in the section of algorithm description.

4 Program

The inputs include the wave parameters wave height (H) and wavelength (L) of the wave to be generated are set by the user. Also the total time of simulation and time step (dt). The size of the domain, particles and parameters like viscosity (ν), radius of influence (r_e), initial particle density (n^0) and free surface condition parameter (β) can be set to desired value by user in the program.


Start
Particle generation
Set the initial conditions
Intermediate velocity and position calculation
Particle density calculation
Solve Pressure Poisson Equation (PPE)
Pressure gradient calculation
Correct velocity, position and then calculate particle density
Introduce particles at inlet with initial conditions
Apply Free Surface and other boundary conditions
Write the output in the file
Stop
Next time step

5 Algorithm Description

5.1 Particle Generation

Particles are systematically generated in such a way that an uniform distribution exists over the whole domain including fluid and solid. Hence the initial particle number density no is constant as is taken as a reference during the computation. This particle generation over all the domain is done once at the beginning of the simulation.

An abstract data structure is used to store field variables values of each particles and all domain particles are stored in form of a vector since its easy to add and remove elements from vector unlike an array.

5.2 Initial setup conditions

The fluid velocity is initialised with the free horizontal speed flow U_0 and pressure P_0 for the wave profile at t=0. The vertical velocity is set to zero. The Korteweg–de Vries (KdV) equation is used to model the solitary waves initially.

Wave elevation

$$\eta = Hsech^2[\tfrac{2*\pi}{L}(x - ct)]$$

where $c = \sqrt{gd}(1 + \frac{H}{2d})$

Horizontal velocity

$$U_0 = \tfrac{\eta}{d}\sqrt{gd}$$

Pressure

$$P_0 = p_0 + \tfrac{1}{\rho}c^2[sech(\tfrac{2*\pi*x}{L})]^2$$

where p_0 is the atmospheric pressure.

5.3 Intermediate position and velocity calculation

The intermediate velocity is calculated with the viscous term and the external force as,

$$u_i^* = u_i^k + \Delta t.(\nu\nabla^2 u_i^k + g)$$

The intermediate position is then calculated for that i^{th} particle as,

$$r^* = r^k + \Delta t.u^*$$

5.4 Pressure Poisson Equation solution

The Poisson Pressure equation is implicitly solved using the original equation obtained after applying the Laplacian formula as discussed earlier. The equation we get is,

$$\frac{2d}{\lambda n^0}\Sigma_{j\neq i}(P_j^{k+1} - P_i^{k+1})w(|r_j - r_i|) = \frac{\rho}{\Delta t^2}\frac{n_i^* - n^0}{n^0}$$

As the system of equations generate the following sparse matrix it is solved using Gauss elimination method.

$$\begin{bmatrix} -\Sigma_{j\neq 1}a_{1j} & a_{12} & a_{13} & .... & a_{1N} \\ a_{21} & -\Sigma_{j\neq 2}a_{2j} & a_{23} & .... & a_{2N} \\ a_{31} & a_{32} & -\Sigma_{j\neq 3}a_{3j} & .... & a_{2N} \\ . & . & . & .... & . \\ . & . & . & .... & . \\ . & . & . & .... & . \\ . & . & . & .... & . \\ . & . & . & .... & . \\ . & . & . & .... & . \\ a_{N1} & a_{N2} & a_{N3} & .... & -\Sigma_{j\neq N}a_{Nj} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ . \\ . \\ . \\ . \\ . \\ . \\ P_N \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ . \\ . \\ . \\ . \\ . \\ . \\ C_N \end{bmatrix}$$

where $a_{ij} = w(|r_j - r_i|)$, $C_i = \frac{\lambda n^0}{2d}\frac{\rho}{\Delta t^2}\frac{n_i^* - n^0}{n^0}$ and N is the size of the vector.

The particle density is calculated for i^{th} particle as,

$$n_i = \Sigma_{j\neq i}w(|r_j - r_i|)$$

5.5 Pressure Gradient calculation

The Gradient pressure is calculated using the equation we got for the gradient operator.

5.6 Velocity correction and Velocity & Position update

The velocity correction is calculated with the remainder part of the Navier Stokes' equation which includes the gradient of pressure as

$$u' = -\frac{\Delta t}{\rho}\nabla P^{k+1}$$

Then the new position and velocity of the fluid particles are updated with the above correction as

$$u^{k+1} = u^* + u'$$
$$r^{k+1} = r^k + \Delta t.u^{k+1}$$

5.7 Imposing boundary conditions

If the particle number density of a particle i n_i, is less than a constant β times the initial n^o, then the particle is identified as one on the free surface.

$$n_i < \beta . n^0$$

Pressure zero is given to those particles as Dirichlet boundary condition. For the present study the parameter β was taken as 0.97 which is taken from the reference listed at the end of report.

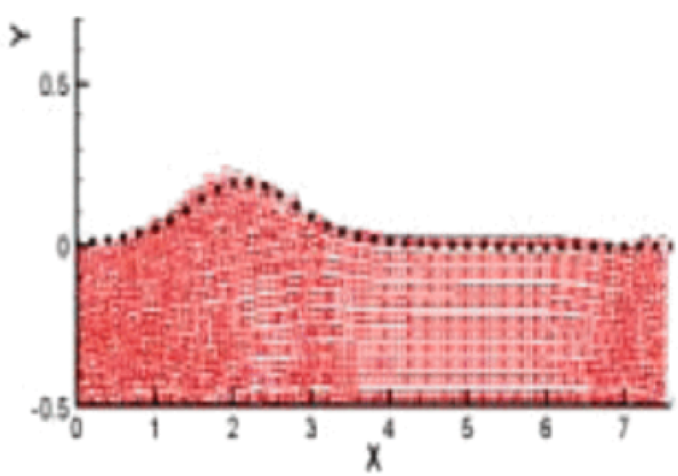
At the inlet, particles are introduced to the domain at upstream velocity and at the outlet particles are removed from the domain. At the bottom there are two layers of particles that moves at free stream velocity. The steps three to nine are repeated until the total time of simulation is reached.

6 Model evaluation on Solitary Wave simulation in a straight channel

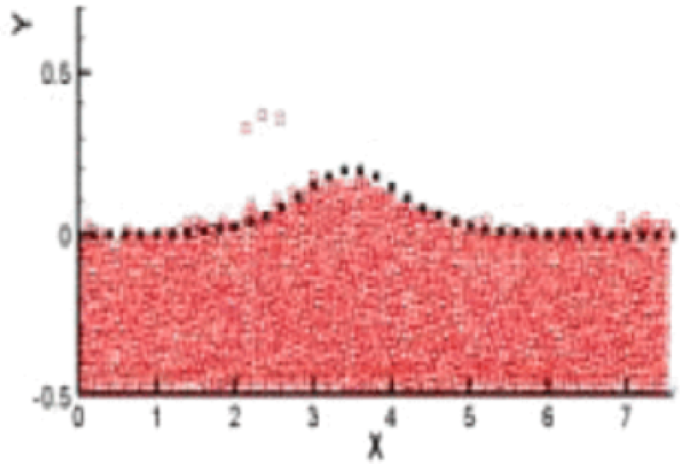
In order to study it, the vertical two-dimensional model is used. MPS model predicts wave profile changes, especially the pressure variation. The source code file "MPS-SW.cpp" is attched along with this file for reference. A 7.5 m channel with 0.5 m initial water depth has been considered.

Particle diameter is considered 0.045 meter and radius of influence of 0.09m. So totally 2496 particles are used in this calculation. The simulation is ran for a time period of 3s with a time step of 0.01s.

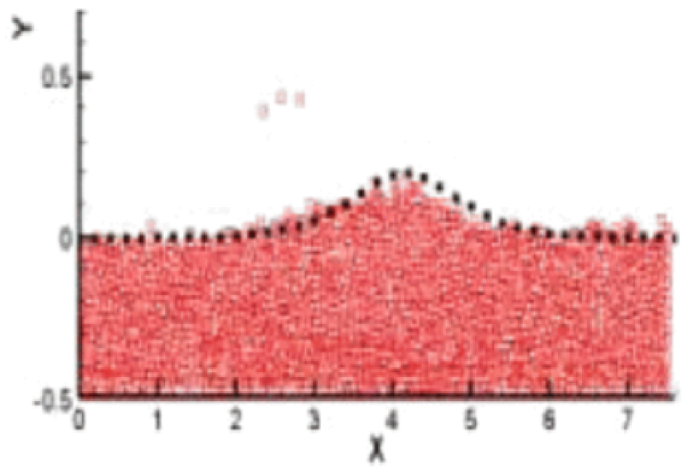
The wave profiles at t = 0.24s, 0.75s and 1.0s are



Wave profile at t = 0.24s



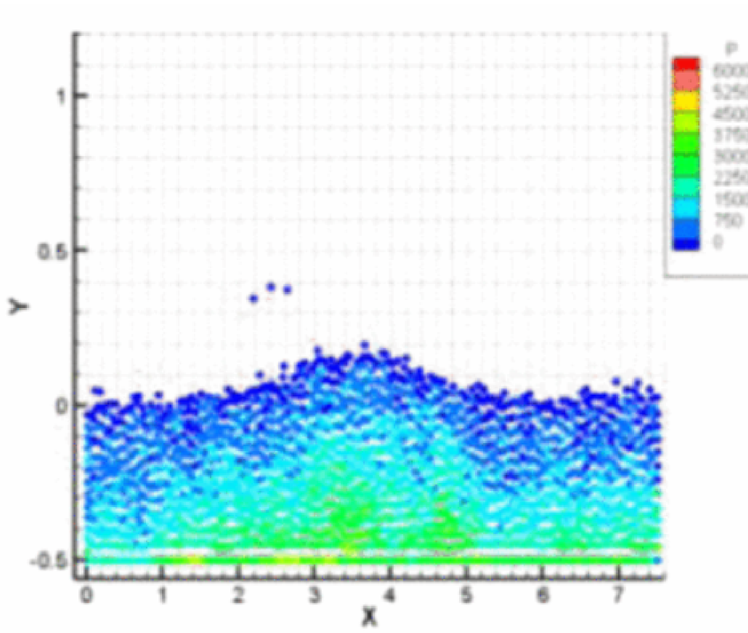
Wave profile at t = 0.75s



Wave profile at t = 1.0s

The water particles (with ID = 1) and the free surface particles from the result are plotted here to capture the wave profile at that instants.

The pressure gradient profile at t=0.81s is plotted for water particles from the result file.



Pressure profile at t = 0.81s

7 Conclusion

- The projection method has been deployed and discretization of N-S equations is completed in two half steps.
- The neighbors are searched each time by looping through the entire vector. This technique is computationally time consuming and works for small scale problems. A neighbour search using Linked Lists or Tree data structure can be implemented to improve efficiency.
- Coupling with methods for force calculation can help to simulate patched bodies to study its effects.
- The RHS source term of the Pressure Poisson Equation can be updated using the modifications suggested in the reference paper to improve pressure gradient approximation.

8 Reference

- Implementation of the Moving Particle Semi-implicit method to predict the drag resistance coefficient on 2D PhD thesis by Carlos Andrés Pérez Gutiérrez Facultad de Ingeniería Universidad Eafit Medellín, June 2016