J. Marine. Sci. Appl. (2008) 7:184-189
DOI: 10.1007/s11804-008-7047-3

Numerical simulation of viscous liquid sloshing by moving-particle semi-implicit method

PAN Xu-jie*（潘徐杰）, **ZHANG Huai-xin**（张怀新）, **LU Yun-tao**（卢云涛）

School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiaotong University, Shanghai 200030, China

Abstract: A meshless numerical simulation method, the moving-particle semi-implicit method (MPS) is presented in this paper to study the sloshing phenomenon in ocean and naval engineering. As a meshless method, MPS uses particles to replace the mesh in traditional methods, the governing equations are discretized by virtue of the relationship of particles, and the Poisson equation of pressure is solved by incomplete Cholesky conjugate gradient method (ICCG), the free surface is tracked by the change of numerical density. A numerical experiment of viscous liquid sloshing tank was presented and compared with the result got by the difference method with the VOF, and an additional modification step was added to make the simulation more stable. The results show that the MPS method is suitable for the simulation of viscous liquid sloshing, with the advantage in arranging the particles easily, especially on some complex curved surface.

Keywords: sloshing; liquid tank; viscous fluid; incompressible fluid; free surface; meshless; moving-particle semi-implicit method (MPS)
CLC number: O357.1 **Document code:** A **Article ID:** 1671-9433(2008)03-0184-06

1 Introduction

Liquid sloshing is the to-and-fro motion of a partially filled tank. With the ship's movement (rolling, pitching and so on), the ballast tank, oil tank of the ship will show sloshing phenomenon. The liquid sloshing loads action on the wall of tank, and that can cause local structural damage, some can influence the stability or maneuverability of the ship[1-2].

Recently, with the development of the LNG, LPG and VLCC, except the ballast tank and oil tank, more tanks of ship need to be considered, so more attention are paid to the sloshing of the liquid tanks in ship. In the last century, many researches used mesh to study the liquid sloshing, resolving the free surface with Marker and Cell (MAC) or Volume of Fluid (VOF), others used experimental methods by virtue of the potential flow theory for the liquid sloshing. Numerical simulation plays an important role in the liquid sloshing study.

In recent years, the viscous CFD technology has been developed, and many meshless methods have been put forward, such as smoothed particle hydrodynamics (SPH), Moving-Particle Semi-implicit Method (MPS), and so on. These meshless methods provided new tools for studying the hydrodynamic phenomena, which have the advantage of arranging the particles on some complex curved surface more easily than mesh methods.

MPS is a new fully Lagrangian meshless method[3], which substitutes particle for mesh in traditional methods, so MPS is called particle method too. Some ocean engineering problems have been studied by MPS method in Japan[4-6]. In this paper, MPS is used to simulate the liquid sloshing and demonstrate the possibility that MPS is suitable for simulating the liquid sloshing.

2 MPS method

The continuity and the Navier-Stokes equations are used as governing equations for incompressible flow:
Continuity equation:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla \boldsymbol{U} \,. \tag{1}$$

Navier-Stokes equation:

$$\frac{\mathrm{D}\boldsymbol{U}}{\mathrm{D}t} = -\frac{1}{\rho}\nabla \boldsymbol{P} + \nu\nabla^2 \boldsymbol{U} + \boldsymbol{F} \,. \tag{2}$$

Received date: 2007-05-31.
Foundation item: Supported by the National Natural Science Foundation under Grant No.50579035.
***Corresponding author Email:** panxujie@126.com

In Eqs.(1) ~ (2), ρ is the density of fluid, t the time, $\boldsymbol{U}$ the velocity vector, $\boldsymbol{P}$ the pressure and $\boldsymbol{F}$ the acceleration of body forces.

2.1 Kernel function

In MPS method, the particles' relationships substitute for the differential operators in traditional methods. The kernel function is used for a particle's interaction with others. This concept has been used in SPH, another meshless method. Study shows that the kernel function plays a very important role in simulations. In this paper, 5 different functions are employed for simulations of sloshing (see Fig.1 and Table 1).

With these different functions, particles only interact with a finite number of neighboring particles. The range of interaction is determined by parameter r_e.

Table 1 Different kernel functions

Kernel function	KF1	KF2	KF3	KF4	KF5
Kernel function formulation	$w(r)=\begin{cases}\frac{r_e}{r}-1, \\ 0\le r<r_e; \\ 0, \\ r_e\le r.\end{cases}$	$w(r)=\begin{cases}\exp[-(\frac{r}{\alpha r_e})^2], \\ 0\le r\le r_e; \\ 0, \\ r_e<r.\end{cases}$	$w(r)=\begin{cases}1-6(\frac{r}{r_e})^2+8(\frac{r}{r_e})^3-3(\frac{r}{r_e})^4, \\ 0\le r\le r_e; \\ 0, \\ r_e<r.\end{cases}$	$w(r)=\begin{cases}\frac{2}{3}-4(\frac{r}{r_e})^2+4(\frac{r}{r_e})^3, \\ 0\le r\le \frac{r_e}{2}; \\ \frac{4}{3}-4(\frac{r}{r_e})+4(\frac{r}{r_e})^2-(\frac{4}{3})(\frac{r}{r_e})^3, \\ \frac{r_e}{2}<r\le r_e; \\ 0, \\ r_e<r.\end{cases}$	$w(r)=\begin{cases}-(2\frac{r}{r_e})^2+2, \\ 0\le \frac{r}{r_e}\le \frac{1}{2}; \\ [2(\frac{r}{r_e})-2]^2, \\ \frac{1}{2}<r\le 1; \\ 0, \\ r_e<r.\end{cases}$
Reference	[7]	[7]	[7]	[7]	[3]

The function $w(r)$ depends on the distance r between the two particles. This function only works when distance r is less than r_e, the distance is closer, and the function $w(r)$ is biger, which guarantees that two particles cannot hit each other.

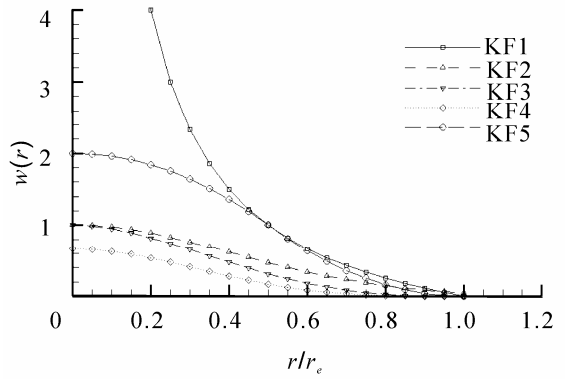


Fig.1 5 different kernel functions

2.2 Particle number density

Particle number density is defined as

$$n_i=\sum_{j\neq i} w(|\boldsymbol{r}_i-\boldsymbol{r}_j|)\,. \tag{3}$$

It is the sum of kernel functions of one particle i that works with other particles within the range of particle i, the contribution from i itself is not considered. Vectors $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the coordinates of particles i and j, respectively.

2.3 Modeling gradient and laplacian

MPS uses the gradient model and Laplacian model to replace terms of governing equation. These two models only work for one particle in its range r_e, which is the same range in kernel functions.

A gradient vector between two particles i and j possessing scalar quantities f_i and f_j at coordinates $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ is given as $(f_j - f_i)(\boldsymbol{r}_j - \boldsymbol{r}_i)/|\boldsymbol{r}_j - \boldsymbol{r}_i|^2$. For one particle i, it needs to calculate the gradient vectors with any combination of two particles which are in the range r_e of particle i. The gradient vectors between particle i and its neighboring particles j are weighted by the kernel function and added to obtain the gradient vector of particle i:

$$<\nabla f>_i=\frac{d}{n_0}\sum_{j\neq i}\left[\frac{f_i-f_j}{|\boldsymbol{r}_j-\boldsymbol{r}_i|^2}(\boldsymbol{r}_j-\boldsymbol{r}_i)\,w(|\boldsymbol{r}_j-\boldsymbol{r}_i|)\right], \tag{4}$$

where the number d of space dimensions is 2 for the 2-D space and 3 for the 3-D space. The gradient model is applied in MPS for the pressure gradient term, and n_0 is the initial particle number density which can be got by Eq.(3) in the initial step of the simulation, vectors $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are the coordinates of particles i and j respectively.

The Laplacian model $\nabla^2 f$ can be got by diffusion equation, which simulates the change of the quantities f_i and f_j between particles i and j, the kernel function should be considered between the two particles:

$$<\nabla^2 f>_i = \frac{d}{\lambda n_0} \sum_{j \neq i} \left[\left(f_j - f_i \right) w\left(\left| \boldsymbol{r}_j - \boldsymbol{r}_i \right| \right) \right]. \quad (5)$$

λ in Eq.(5) can be got as follows:

$$\lambda = \frac{\sum_{j \neq i} w\left(\left| \boldsymbol{r}_j - \boldsymbol{r}_i \right| \right) \left| \boldsymbol{r}_j - \boldsymbol{r}_i \right|^2}{w\left(\left| \boldsymbol{r}_j - \boldsymbol{r}_i \right| \right)}. \quad (6)$$

Other parameters are similar to Eq.(4).

2.4 Modeling of incompressibility

The continuity equation of governing equation requires that the density of fluid should be constant, MPS keeps n_0 as a constant to ensure the fluid's incompressibility. In the simulation, if the number density n^* is not n_0, it needs to be implicitly corrected to n_0 by

$$n^* + n' = n_0, \quad (7)$$

where n' is the correction value related to the velocity correction value u':

$$\frac{1}{\Delta t} \frac{n'}{n_0} = -\nabla \cdot u'. \quad (8)$$

$\triangle t$ is the increment of time, and the velocity correction value can be derived from the implicit pressure gradient as follows:

$$u' = -\frac{\Delta t}{\rho} \nabla P^{n+1} \quad (9)$$

With Eqs.(7) ~ (9), a pressure Poisson equation is obtained as

$$<\nabla^2 P^{n+1}> = -\frac{\rho}{\Delta t^2} \frac{<n^*>_i - n_0}{n_0}. \quad (10)$$

The pressure Poisson equation can be solved by super over relax method (SOR) or incomplete Cholesky conjugate gradient method (ICCG).

2.5 Modeling of free surface

In a full Lagrangian motion of particles, free surface is clear to be traced by:

$$<n^*>_i < \beta n_0. \quad (11)$$

n_0 is the initial particle number density calculated in the initial step of the simulation, $<n^*>_i$ is the number density of particle i in any simulation step, β can be chosen between 0.8 and 0.99.

3 Algorithm of simulation

In each step of simulation, the source terms are explicitly calculated and the temporal velocities $\triangle \boldsymbol{u}_i^*$ are got, then use the temporal velocities to calculate the motion of particles, and temporal coordinates $\boldsymbol{r}_i^*$ are obtained. And then, the number density of each particle is changed. Use the number density and initial number density to set the pressure Poisson equation, then solve the pressure Poisson equation, and then get the pressure of the next step of simulation. Finally, the new velocities and coordinates of the next step of simulation are obtained by adding the pressure gradient terms with the next step pressure values, particularly,

1) Calculate the initial velocities u^0 and coordinates r^0 of each particle;

2) In one increment of time $\triangle t$, calculate the increment of velocities $\triangle \boldsymbol{u}_i^*$ by

$$\Delta \boldsymbol{u}_i^* = \Delta t \left(\nu \nabla^2 \boldsymbol{u}_i^n + \boldsymbol{F} \right); \quad (12)$$

3) Obtain the temporal velocities $\boldsymbol{u}_i^*$ and $\boldsymbol{r}_i^*$ coordinates by $\triangle \boldsymbol{u}_i^*$:

$$\boldsymbol{u}_i^* = \boldsymbol{u}_i^n + \Delta \boldsymbol{u}_i^*, \quad (13)$$

$$\boldsymbol{r}_i^* = \boldsymbol{r}_i^n + \Delta \boldsymbol{u}_i^* \cdot \Delta t; \quad (14)$$

4) Obtain the new number density of each particle $<n^*>_i$ by Eq.(3) and set the pressure Poisson Eq.(10);

5) With the next step's pressure of each particle, calculate the correction velocities:

$$\boldsymbol{u}_i' = -\frac{\Delta t}{\rho} \nabla P_i^{n+1}; \quad (15)$$

6) With the correction velocities, obtain the new velocities $\boldsymbol{u}_i^{n+1}$ and coordinates $\boldsymbol{r}_i^{n+1}$ of the next step of simulation:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}_i^* + \boldsymbol{u}_i'; \quad (16)$$

$$\boldsymbol{r}^{n+1} = \boldsymbol{r}_i^* + \boldsymbol{u}_i' \cdot \Delta t. \quad (17)$$

4 Numerical results of sloshing

The sloshing of a partially filled tank is presented in this paper. The geometry is depicted in Fig.2. The tank was surging to-and-fro with the frequency of 1.0 Hz, which was close to the first order linear frequency of an ideal fluid (1.008 Hz), the amplitude of the surging

was 0.005 m. Result of this sloshing was compared with the result of the difference method with the VOF[8].

The increment of time $\triangle t$ is a dynamic state in this paper, the fastest particle in each simulation step is chosen, and the velocity of this particle is used in Courant's law for calculation of $\triangle t$:

$$\Delta t = \alpha \frac{l_0}{\boldsymbol{v}_{\max}}. \tag{18}$$

In Eq.(18), l_0 is the distance of particles, $\boldsymbol{v}_{\max}$ is the velocity of the fastest particle, α in generalized case is 0.1.

An open tank was used for sloshing at first, but it was found that at about 2.0 s of the simulation, one particle spattered out of the tank, with the gravity action, this particle became so fast that the increment of time $\triangle t$ became very little. So a close tank was chosen for sloshing in this paper (see Fig.2).

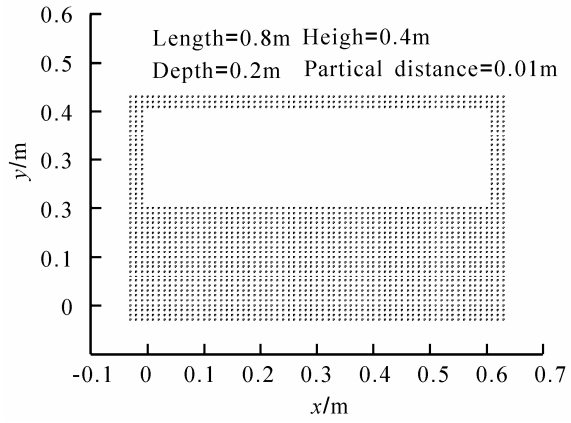
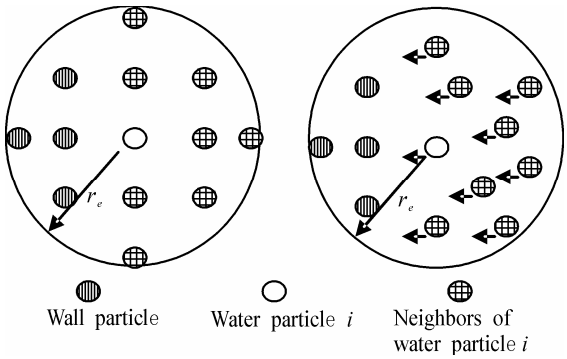


Fig.2 Initial distribution of particles and scales of the tank

At some steps of the simulation, a few of particles were too close and even passed through the wall of tank after the first correction of coordinates in Eq.(14), so the force given by kernel function was very big, and pushed the particle too far, while it could not happen in the real situation. It occurred because after the second correction of the last time step, the number density of each particle went back to the initial number density n_0, but it couldn't guarantee the distance of each particle with its neighbors going back to l_0, so l_i^{n} may be less than $0.1l_0$, that caused the pass-through phenomenon in the first correction after Eq.(14) (see Fig.3).



(a) Particle i and neighbors in initial step (b) Particle i and neighbors in some time step

Fig.3 Reason of the pass-through phenomenon

Because of that, an additional modification is added after the correction of coordinates, by choosing a distance parameter. If two particles are close enough, pull them off by this additional modification (see Fig.4).

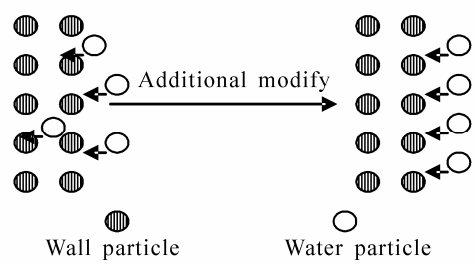


Fig.4 Additional modification step

Although the additional modification step has been taken, some kernel functions still cannot work in simulations, but in other functions, this additional modification step works well.

Five different kernel functions in Table 1 were used for simulation, and in kernel function 2, 3.0 was chosen for α. The results show that kernel functions 2 and 5 aren't suitable for simulation of sloshing with 0.005 m surging and 1.0 Hz frequency (see Fig.5). The other three kernel functions can be used in the simulation.

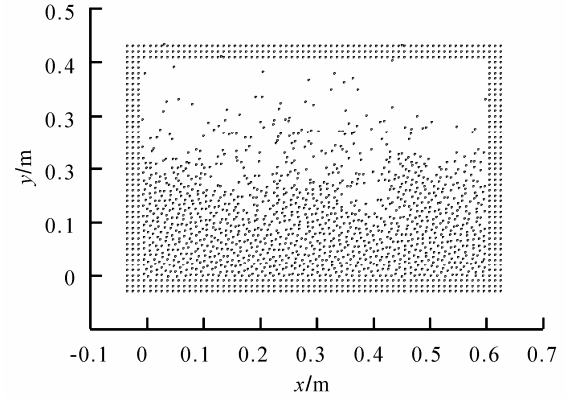


Fig.5 One example of failure

The simulation shows that results of kernel functions 1, 2 and 4 are different in motions on each side of wall, and the results show that the motion of kernel function 3 is the most close to the result of VOF (see Fig.6), so kernel function 3 is the most suitable for simulation of sloshing with 0.005 m surging and 1.0 Hz frequency.

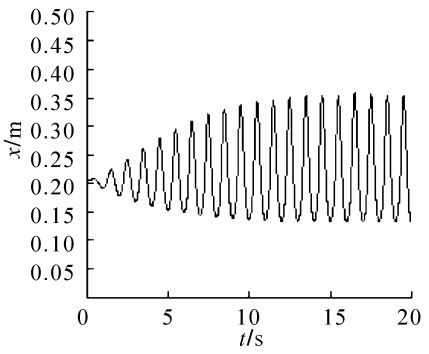
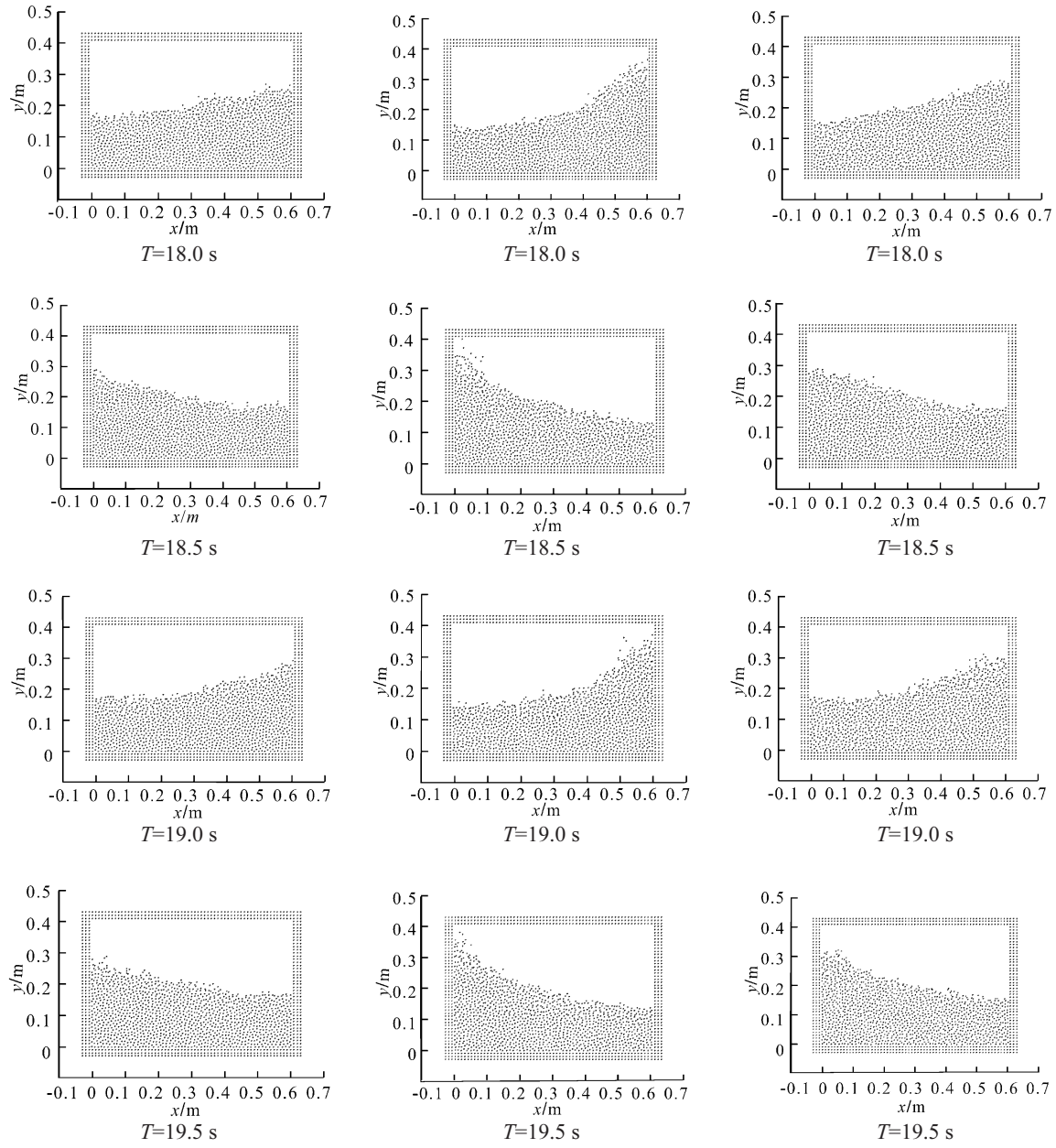


Fig.6 The surface of water on left wall in VOF



(a) Kernel Function 1 (b) Kernel Function 3 (c) Kernel Function 4

Fig.7 Sloshing simulation using different kernel functions

Comparing the result with the difference method with VOF[8], it shows the velocities and coordinates simulated by MPS with kernel function 3 are very close to VOF (see Figs.6~7), and the pressure is acceptable too (see Figs.8~9). In pressure part, there are lots of fluctuation because the random of the

particles' motion and every motion can influence the pressure of corresponding particle.

As a full Lagrangian meshless method, MPS can trace the free surface easily and show the water condition in sloshing more easily, which are advantageous than some Eulerian numerical methods, but it's difficult to trace the data of one fixed position, because of the continues moving particles.

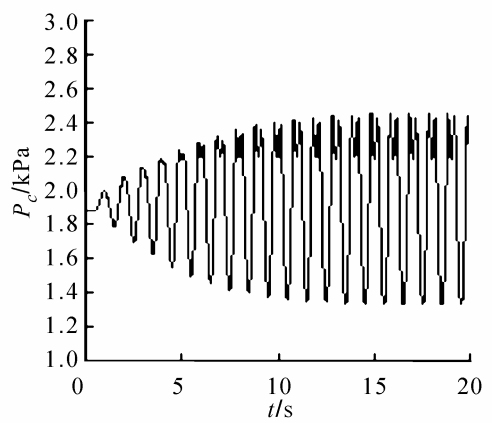


Fig.8 Pressure of left bottom (MPS, Kernel function3)

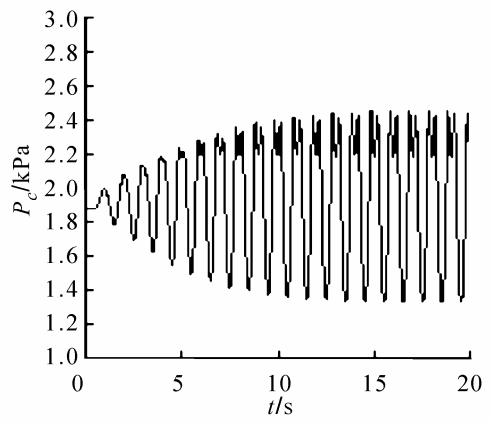


Fig.9 Pressure of left bottom (VOF)

5 Conclusions

In this paper, a simulation of liquid sloshing is put forward by MPS and 5 different kernel functions. It is found that kernel function 3 is most suitable for simulation. Result shows that MPS can trace the free surface easily and show the fluid condition in each step. An additional modification step is added to make the simulation more stable.

After comparing with other results, it shows the result of MPS is acceptable, and MPS is suitable for simulating the liquid sloshing. This paper just presents an initial study for the liquid sloshing using MPS, and more complicated sloshing simulations need to be studied based on MPS.

References

[1] VALENTINE D T, FRANDSEN J B. Numerical investigation of two-dimensional sloshing: Nonlinear internal waves[J]. Journal of Offshore Mechanics and Arctic Engineering, 2005, 127(4): 300-305.

[2] LEE S J, KIM M H, LEE D H, KIM J W , KIM Y H. The effects of LNG-tank sloshing on the global motions of LNG carriers[J]. Ocean Engineering, 2007, 34(1): 10-20.

[3] KOSHIZUKA S, OKA Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid[J]. Nuclear Science and Engineering, 1996, 123(3): 421-434.

[4] SUEYOSHI M, NAITO S. A study of nonlinear fluid phenomena with particle method (Part1)[J]. Journal of Kansai Society Naval Architects, 2001, 236: 191-198(in Japanese).

[5] SUEYOSHI M, NAITO S. A study of nonlinear fluid phenomena with particle method (Part2)[J]. Journal of Kansai Society Naval Architects, 2002, 237: 181-186(in Japanese).

[6] SUEYOSHI M, NAITO S. A study of nonlinear fluid phenomena with particle method (Part3)[J]. Journal of Kansai Society Naval Architects, 2003, 239: 81-86(in Japanese)

[7] ATAIE-ASHTIANI B, FARHADI L. A stable moving-particle semi-implicit method for free surface flows[J]. Fluid Dynamics, 2006, 38(4): 241-256.

[8] ZHU R Q. Time domain simulation of liquid sloshing and its interaction with flexible structure[D]. Wuxi: China Ship Scientific Research Center, 2002(in Chinese).

PAN Xu-jie was born in 1980. He is a Ph.D candidate at Shanghai Jiaotong University. His current research interests include computational fluid dynamics.

ZHANG Huai-xin was born in 1961. She is a professor at Shanghai Jiaotong University. Her current research interests include computational fluid dynamics.

LU Yun-Tao was born in 1982. He received his master degree at Shanghai Jiaotong University in 2008.