NUCLEAR SCIENCE AND ENGINEERING: **123**, 421–434 (1996)

Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid

S. Koshizuka and Y. Oka

University of Tokyo, Nuclear Engineering Research Laboratory, Faculty of Engineering
2-22 Shirane, Shirakata, Tokai-mura, Naka-gun, Ibaraki 319-11, Japan

Received September 15, 1995
Accepted February 16, 1996

Abstract—*A moving-particle semi-implicit (MPS) method for simulating fragmentation of incompressible fluids is presented. The motion of each particle is calculated through interactions with neighboring particles covered with the kernel function. Deterministic particle interaction models representing gradient, Laplacian, and free surfaces are proposed. Fluid density is implicitly required to be constant as the incompressibility condition, while the other terms are explicitly calculated. The Poisson equation of pressure is solved by the incomplete Cholesky conjugate gradient method. Collapse of a water column is calculated using MPS. The effect of parameters in the models is investigated in test calculations. Good agreement with an experiment is obtained even if fragmentation and coalescence of the fluid take place.*

I. INTRODUCTION

Computer simulation has been required to analyze increasingly complex geometry and physics problems. However, in the field of nuclear engineering, we have many thermal-hydraulic problems that are still difficult to analyze via computer with either the finite volume or the finite element method, including water-steam two-phase flows, where the shape of the interface changes continuously; vapor explosions, where water, steam, and liquid metal should be simultaneously analyzed; and fluid/structure interactions, where large deformation of both fluids and structures should be considered.

Computer simulation using particles has the capability to analyze more complex geometry and physics than grid methods. Particularly, topological deformation of the fluid can be analyzed by particles, while it is impossible to fit and move a grid continuously in such domains. Another advantage is that convection is directly calculated by the motion of particles without numerical diffusion. Thus, interfaces are kept clear. In addition, grid generation is not necessary. Recently it seems that increasingly complex domains are analyzed with grid methods, which requires increasingly complex grids to be generated. Although we need to set initial configurations of particles in particle methods, this is easier than in grid generation because we do not need to set up topological relations among the particles. It is also easier to add or remove particles in the midst of the calculation.

Particle methods can be roughly classified into two types: (a) those based on probabilistic models, such as the molecular dynamics,[1] direct simulation Monte Carlo,[2] and lattice gas automaton,[3] and (b) those based on deterministic models, such as the particle-and-force method,[4] the smoothed particle hydrodynamics[5–10] (SPH), a gridless Euler/Navier-Stokes solution algorithm,[11] the element-free Galerkin method,[12–14] and the discrete element method.[15] The first set of methods represents macroscopic properties as statistical behaviors of microscopic particles, so that a huge number of particles should be traced for a long period to obtain accurate average values. The second set of methods requires much shorter computation time and smaller storage. SPH has been used in astrophysics to determine the fluid dynamics of interstellar gas, which is regarded as a compressible, inviscid fluid.

Grid-particle hybrid methods, such as marker-and-cell,[16] particle-in-cell,[17] and fluid-implicit-particle,[18,19] partially use particles to trace interfaces or to calculate convection terms without numerical diffusion. In these methods, successful techniques of grid methods can be

integrated, although they still suffer from the limitation of grid methods.

A deterministic particle method for incompressible flows, which appear in many industrial applications, was proposed.[20-22] All terms in the Navier-Stokes equation were represented by particle interaction models. Accuracy of these models was investigated in test calculations; for example, a parabolic profile of the flow velocity was obtained in a Poiseuille flow calculation. Using this method, collapse of a water column was calculated. However, particles near the free surface were too widely dispersed, and comparison with the experimental result was not good. Numerical stability of the incompressibility model was sensitive to a correction parameter used in the method, and the calculation often failed due to numerical instability. Computation time was much longer than that of usual grid methods.

In this paper, a modified particle method, called the moving-particle semi-implicit (MPS) method, is presented. Due to the modified kernel function and incompressibility model in MPS, numerical stability and computation speed are markedly improved. The main parameters used in MPS are investigated with test calculations. Collapse of a water column is calculated again with the modified method with selected parameters.

Collapse of a water column has been used to verify the calculation of free surfaces.[23,24] Motion of the leading edge has been compared with that of an experiment.[25] The free surface is kept smooth in this experiment since there are no obstacles or vertical walls on which the collapsing water can impinge. In the experiments of Koshizuka, Oka, and Tamako,[21,22] a vertical wall was added to create fragmentation and coalescence of the fluid. This is used for comparison with the current calculation. In Ref. 26, collapse of a cylindrical water column with a wall and an obstacle is calculated using grid methods. The calculation results are compared with experimental ones. Although fragmentation of the fluid is calculated, the free surface is not clear. This must be due to the numerical diffusion, which is inevitable in the convection calculation of the fluid fraction on the stationary grid.

II. GOVERNING EQUATIONS

Governing equations for incompressible flows are the continuity and the Navier-Stokes equations as follows:

$$\frac{\partial \rho}{\partial t} = 0 \tag{1}$$

and

$$\frac{D\boldsymbol{u}}{Dt} = -\frac{1}{\rho}\nabla P + \boldsymbol{f} \ . \tag{2}$$

The continuity equation is written with density, while velocity divergence is usually used in grid methods. The left side of the Navier-Stokes equation [Eq. (2)] denotes the Lagrangian differentiation including convection terms. This is directly calculated by tracing particle motion. The right side consists of pressure gradient and external force terms. All terms expressed by differential operators should be replaced by particle interactions.

III. PARTICLE INTERACTION MODELS

III.A. Kernel

A particle interacts with others in its vicinity covered with kernel function $w(r)$, where r is the distance between two particles. This concept has been used in the SPH method. In this study, the following function is employed (Fig. 1):

$$w(r) = \begin{cases} \dfrac{r_e}{r} - 1 & (0 \le r < r_e) \\ 0 & (r_e \le r) \end{cases} \ . \tag{3}$$

Since the area that is covered by this kernel function is bounded, a particle interacts with a finite number of neighboring particles. The radius of the interaction area is determined by parameter r_e. Compared with a kernel function covering an infinite area, such as a Gaussian function, the current function needs less memory and computation time.

The kernel function is infinity at $r = 0$. This is good for numerical stability in the model of incompressibility. On the other hand, the old kernel function used in Refs. 20, 21, and 22 gives a finite value at $r = 0$. Comparison between two functions is shown later with test calculations.

III.B. Particle Number Density

Particle number density at coordinate $\boldsymbol{r}_i$ where particle i is located is defined by

$$\langle n \rangle_i = \sum_{j \ne i} w(|\boldsymbol{r}_j - \boldsymbol{r}_i|) \ . \tag{4}$$

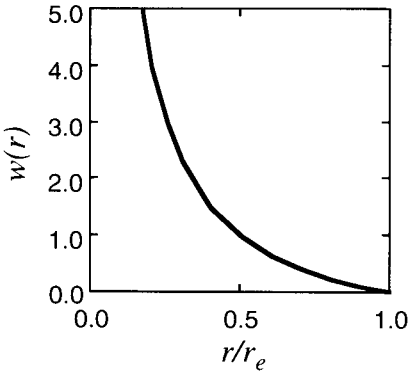


Fig. 1. Kernel function.

In this equation, the contribution from particle i itself is not considered. When the number of particles in a unit volume is denoted by $\langle N\rangle_i$, the relation between $\langle n\rangle_i$ and $\langle N\rangle_i$ is written as

$$\langle N\rangle_i = \frac{\langle n\rangle_i}{\int_V w(r)\,dv} \,. \tag{5}$$

The denominator of Eq. (5) is the integral of the kernel in the whole region, excluding a central part occupied by particle i. Assuming that the particles have the same mass m, we can see that the fluid density is proportional to the particle number density:

$$\langle \rho\rangle_i = m\langle N\rangle_i = \frac{m\langle n\rangle_i}{\int_V w(r)\,dv} \,. \tag{6}$$

Thus, the continuity equation is satisfied if the particle number density is constant. This constant value is denoted by n^0.

III.C. Modeling of Gradient

A gradient vector between two particles i and j possessing scalar quantities ϕ_i and ϕ_j at coordinates $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ is simply defined by $(\phi_j-\phi_i)(\boldsymbol{r}_j-\boldsymbol{r}_i)/|\boldsymbol{r}_j-\boldsymbol{r}_i|^2$. We can evaluate a gradient vector with any combination of two particles. The gradient vectors between particle i and its neighboring particles j are weighted with the kernel function and averaged to obtain a gradient vector at particle i:

$$\langle \nabla\phi\rangle_i = \frac{d}{n^0}\sum_{j\neq i}\left[\frac{\phi_j-\phi_i}{|\boldsymbol{r}_j-\boldsymbol{r}_i|^2}(\boldsymbol{r}_j-\boldsymbol{r}_i)w(|\boldsymbol{r}_j-\boldsymbol{r}_i|)\right] , \tag{7}$$

where d is the number of space dimensions. This model is applied to the pressure gradient term in MPS.

When particles gather, pressure increases and repulsive forces work among the particles to satisfy the continuity equation. Equation (7) gives a larger force to a shorter distance between two particles. This is a good property to avoid the clustering of particles.

The current model is not sensitive to absolute pressure. This is consistent with the property of incompressible fluids, which depends on the relative pressure distribution.

The following equation is a good approximation if particle i is located at the center of its neighboring particles j:

$$0 = \sum_{j\neq i}\left[\frac{(\boldsymbol{r}_j-\boldsymbol{r}_i)}{|\boldsymbol{r}_j-\boldsymbol{r}_i|^2}w(|\boldsymbol{r}_j-\boldsymbol{r}_i|)\right] . \tag{8}$$

Thus, Eq. (7) can be rearranged as follows:

$$\langle \nabla\phi\rangle_i = \frac{d}{n^0}\sum_{j\neq i}\left[\frac{\phi_j-\phi_i'}{|\boldsymbol{r}_j-\boldsymbol{r}_i|^2}(\boldsymbol{r}_j-\boldsymbol{r}_i)w(|\boldsymbol{r}_j-\boldsymbol{r}_i|)\right] , \tag{9}$$

where any constant value is allowed to ϕ_i'. In the code, Eq. (9) is used instead of Eq. (7) as the model of gradient, and ϕ_i' is calculated from

$$\phi_i' = \min(\phi_j) \,, \tag{10}$$

for any j satisfying

$$w(|\boldsymbol{r}_j-\boldsymbol{r}_i|)\neq 0 \,.$$

III.D. Modeling of Laplacian

A time-dependent diffusion problem with respect to ϕ is represented by Laplacian as

$$\frac{d\phi}{dt} = \nu\nabla^2\phi \,, \tag{11}$$

where ν is the diffusion coefficient. The variance of the distribution of ϕ increases by $2d\nu\Delta t$ during time step Δt, where d is the number of space dimensions. In the current model, part of quantity ϕ_i of particle i is distributed to the neighboring particles according to the kernel function such that the variance increase is equal to $2d\nu\Delta t$. Thus, the quantity transferred from particle i to j is

$$\Delta\phi_{i\to j} = \frac{2d\nu\Delta t}{n^0\lambda}\phi_i w(|\boldsymbol{r}_j-\boldsymbol{r}_i|) \,, \tag{12}$$

where

$$\lambda = \frac{\int_V w(r)r^2\,dv}{\int_V w(r)\,dv} \,. \tag{13}$$

As far as linear diffusion is concerned, the quantity transfer can be superposed. Eventually, Laplacian is represented by

$$\underset{\varepsilon}{\langle \nabla^2\phi\rangle_i} = \frac{2d}{n^0\lambda}\sum_{j\neq i}(\phi_j-\phi_i)w(|\boldsymbol{r}_j-\boldsymbol{r}_i|) \,. \tag{14}$$

The current model of Laplacian is conservative since the quantity lost by particle i is just obtained by particles j.

III.E. Modeling of Incompressibility

The continuity equation requires that the fluid density should be constant. This is equivalent to the particle number density being constant, n^0. When the particle number density n^* is not n^0, it is implicitly corrected to n^0 by

$$n^* + n' = n^0 \,, \tag{15}$$

where n' is the correction value. This is related to the velocity correction value $\boldsymbol{u}'$ through the mass conservation equation:

$$\frac{1}{\Delta t}\frac{n'}{n^0} = -\nabla\cdot\boldsymbol{u}' \,. \tag{16}$$

The velocity correction value is derived from the implicit pressure gradient term as

$$\boldsymbol{u}' = -\frac{\Delta t}{\rho} \nabla P^{n+1} , \tag{17}$$

With Eqs. (15), (16), and (17), a Poisson equation of pressure is obtained:

$$\langle \nabla^2 P^{n+1} \rangle_i = -\frac{\rho}{\Delta t^2} \frac{\langle n^* \rangle_i - n^0}{n^0} . \tag{18}$$

The right side is represented by the deviation of the particle number density from the constant value, while it is usually velocity divergence in grid methods. The left side of Eq. (18) is discretized by the Laplacian model [Eq. (14)]. Finally we have simultaneous equations expressed by a linear symmetric matrix. These are solved by the incomplete Cholesky conjugate gradient (ICCG) method.[27] The pressure gradient terms are calculated from the gradient model [Eq. (9)], where scalar ϕ is substituted by P^{n+1}.

The computation speed of the current model is much faster than the previous model used in Refs. 20, 21, and 22 since a matrix equation is constructed, and it is solved by ICCG. In addition, the calculation is more stable because of the robustness of ICCG, even if the particle configuration is strongly distorted.

Equation (18) is obtained through the manipulation of a differential equation so that the discretization form of Eq. (18) is inconsistent with discretization forms of the original equations [Eqs. (15), (16), and (17)]. In other words, the modified particle number density, which is evaluated from the modified coordinates of particles, is not equal to n^0. However, this inconsistency is not serious since the error of the particle number density is not accumulated in each time step and since the mass conservation is strictly guaranteed by keeping the number of particles.

The overall algorithm is described in Fig. 2. In each time step, source terms are explicitly calculated and temporal velocities $\boldsymbol{u}_i^*$ are obtained. Next, motion of particles is calculated, and temporal coordinates $\boldsymbol{r}_i^*$ are obtained. This corresponds to the convection term. The Poisson equation of pressure is solved with the source term representing the deviation of the particle number density, and then new-time pressure values are obtained. Finally we have new-time velocities and coordinates by adding the pressure gradient terms with the new-time pressure values. This algorithm is basically semi-implicit time marching, which has been widely used in grid methods.

III.F. Modeling of Free Surfaces

Free surfaces are always clear as a result of the fully Lagrangian motion of particles. A particle whose particle number density satisfies

$$\langle n \rangle_i^* < \beta n^0 \tag{19}$$

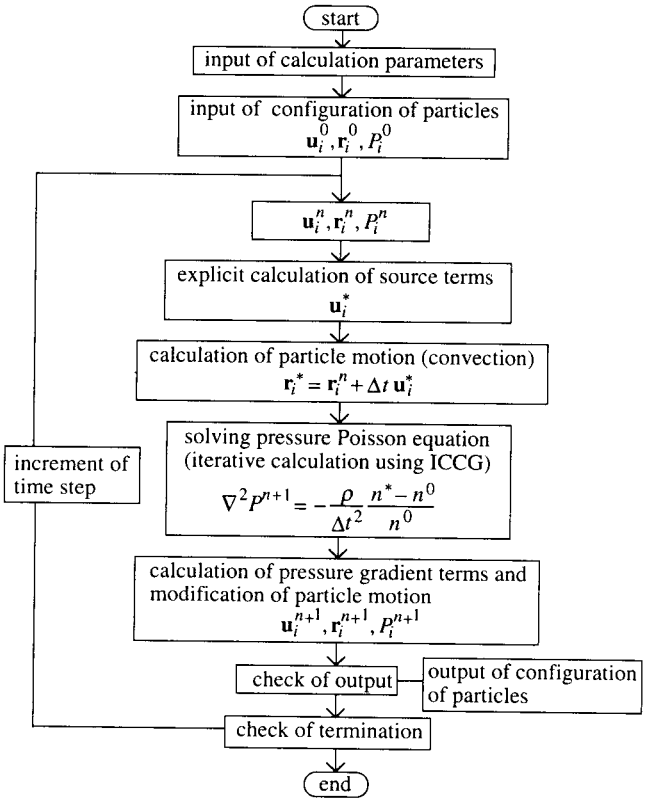


Fig. 2. Calculation algorithm of MPS.

is regarded as the free surface, where β is a parameter below 1.0. Pressure zero is given to this particle as the boundary condition when the pressure Poisson equation is solved. The effect of parameter β is described later.

IV. COLLAPSE OF A WATER COLUMN

IV.A. Geometry

The collapse of a water column is calculated, and the geometry is depicted in Fig. 3. An experiment was carried out in this geometry.[21,22] The water column is initially located on the left vertical wall. In the experiment, the initial water column is supported by a removable board, which is quickly slid up to start the collapse. Collapsing water impinges on the right vertical wall, which creates fragmentation and coalescence of the fluid. The problem without this right vertical wall has been used as a verification problem of the codes for free surfaces.[23,24] In this case, the shape of the free surface is kept smooth so that grid methods can be applied.

IV.B. Calculation Parameters

Parameters used in the current particle interaction models are investigated with test calculations of the collapse of a water column.

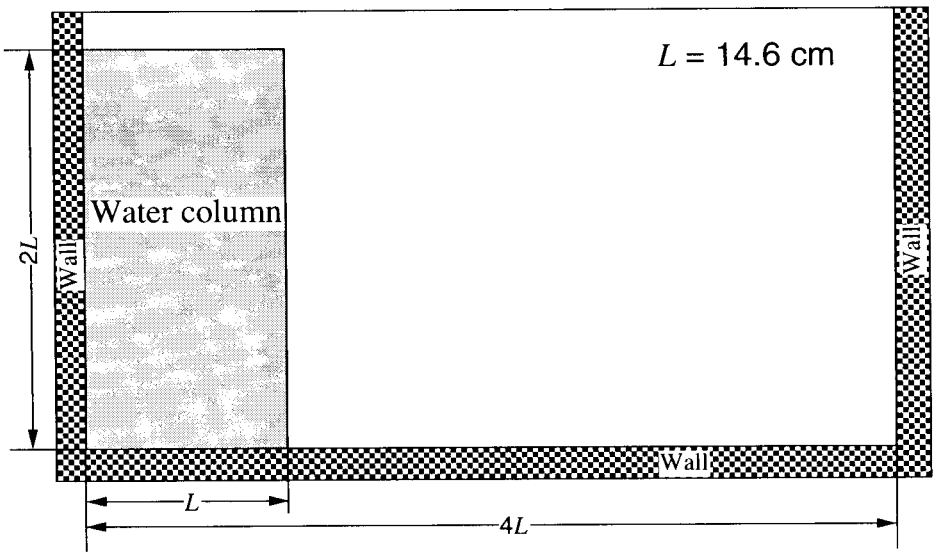


Fig. 3. Geometry of collapse of a water column.

The free surface parameter β is used to judge whether the particle is on the free surface or not [Eq. (19)]. Figure 4 shows the number of particles regarded as the free surface. The trajectories are almost the same from $\beta =$ 0.8 to 0.99, although they are shifted lower in parallel when the parameter is smaller. The calculation result of $\beta = 0.7$ shows a different trajectory from 0.0 s, and the calculation is abnormally terminated at 0.6 s. We can conclude that the free surface parameter is not effective to the calculation result if the calculation proceeds stably. The parameter can be chosen from 0.8 to 0.99. In this study, 0.97 is selected.

The kernel size is represented by parameter r_e in Eq. (3). A larger kernel size leads to more particles for interactions. The kernel function is used for calculating the particle number density, the pressure gradient

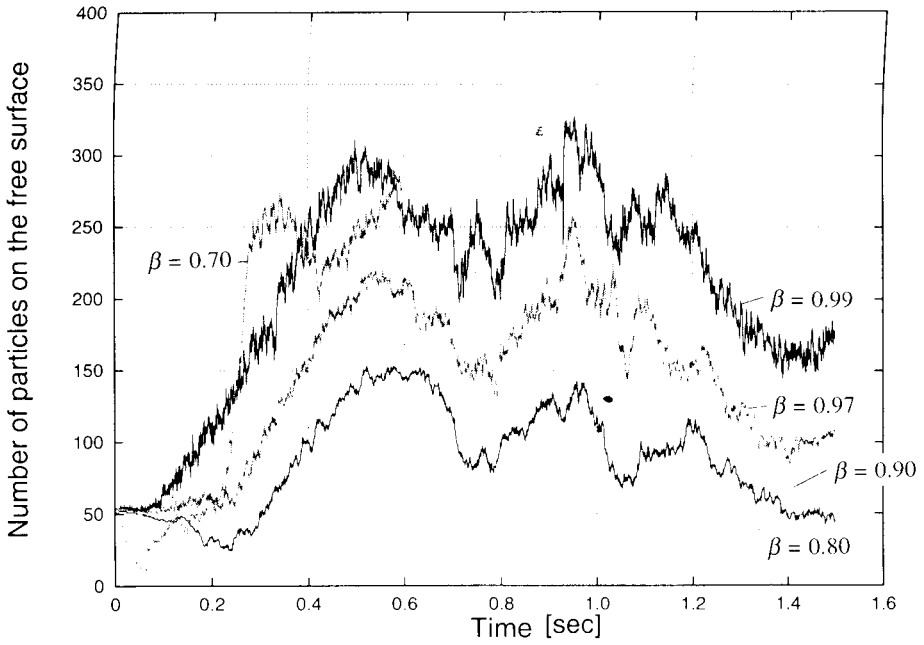


Fig. 4. Effect of free surface parameter.

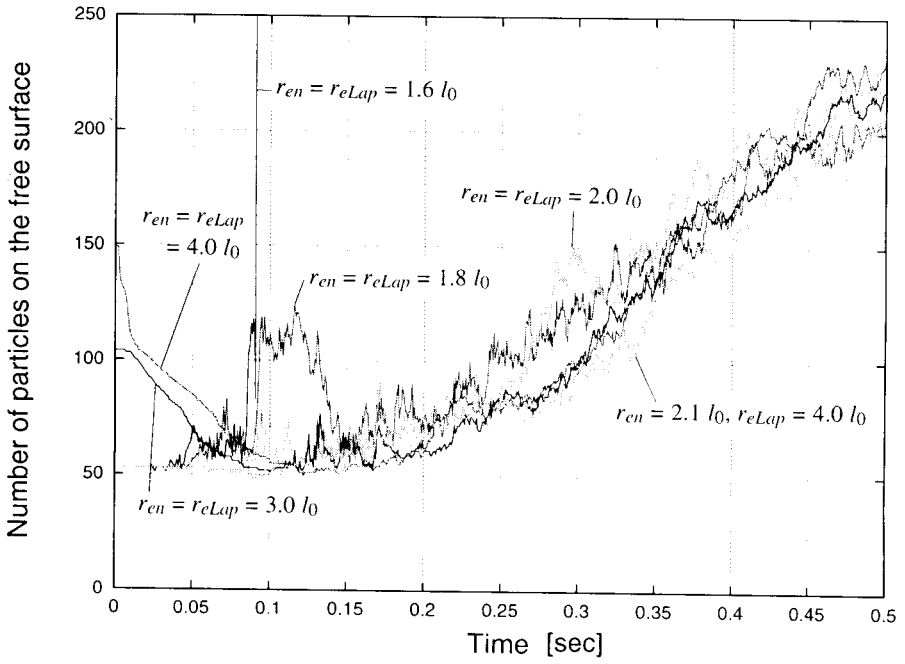


Fig. 5. Effect of kernel size used in Eq. (3)

term, and Laplacian of pressure. It is not necessary to employ a common kernel size for the three calculations. In Fig. 5, the effect of the kernel size is shown, where r_{en} is used in the particle number density and the pressure gradient term and r_{eLap} is used in the Laplacian of pressure. The calculations are successful when $r_{en} = r_{eLap} \geq 1.8l_0$, where l_0 is the distance between neighboring particles in the initial configuration. A numerical explosion occurs when $r_{en} = r_{eLap} = 1.6l_0$. If r_e is too small, the number of neighboring particles for interactions is too small for accurate and stable calculations. It is reasonable that there is a minimum limit of r_e for numerical stability. When $r_{en} = r_{eLap} = 1.8l_0$, the number of particles on the free surface is abnormally high between 0.1 and 0.15 s. This means that numerical instability occurs but is suppressed.

We can see strange trajectories from 0.0 to 0.1 s at $r_{en} = r_{eLap} = 3.0$ and $4.0l_0$. The initial numbers are larger than those of other cases and decrease until 0.1 s. The initial configuration of particles is like a square grid (see Fig. 10a-2). Peripheral particles are regarded as the free surface since fewer neighboring particles are located where the kernel function lies. When the kernel size is small, lower than $3.0l_0$ in this case, particles on the first line from the periphery of the water column are judged as the free surface. When the kernel size is larger, more particles in the inner region satisfy the free surface condition. Thus, the initial numbers are large. Figure 6 shows the calculated configuration of particles at 0.1 s with $r_{en} = r_{eLap} = 4.0l_0$. We can see that the particles gather near the free surface. As a result, the particle number densities of the inner particles are enhanced, and they are not judged as the free surface any longer. This leads to a decrease of the number. Therefore, the kernel size should be $<3.0l_0$, otherwise particles will gather near the free surface. On the other hand, it is shown that the discretization of Laplacian is more accurate when the kernel size r_{eLap} is larger.[21,22] To satisfy these contrary requirements, different values are employed for r_{en} and r_{eLap}: $r_{en} = 2.1l_0$ and $r_{eLap} = 4.0l_0$ in this study.

The average CPU time needed in one time iteration is plotted in Fig. 7. The calculation is carried out with Indigo#2 from Silicon Graphics, Inc. The most time-consuming routine is the ICCG solver of the Poisson equation of pressure. This routine consists of generation of the matrix, the incomplete Cholesky decomposition,

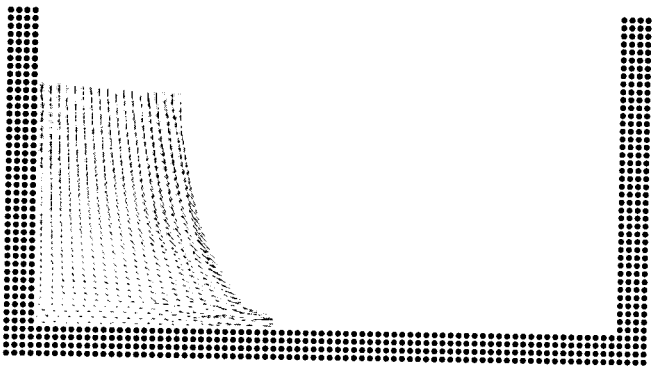


Fig. 6. Configuration of particles at 0.1 s with $r_{en} = r_{eLap} = 4.0l_0$ in Eq. (3).

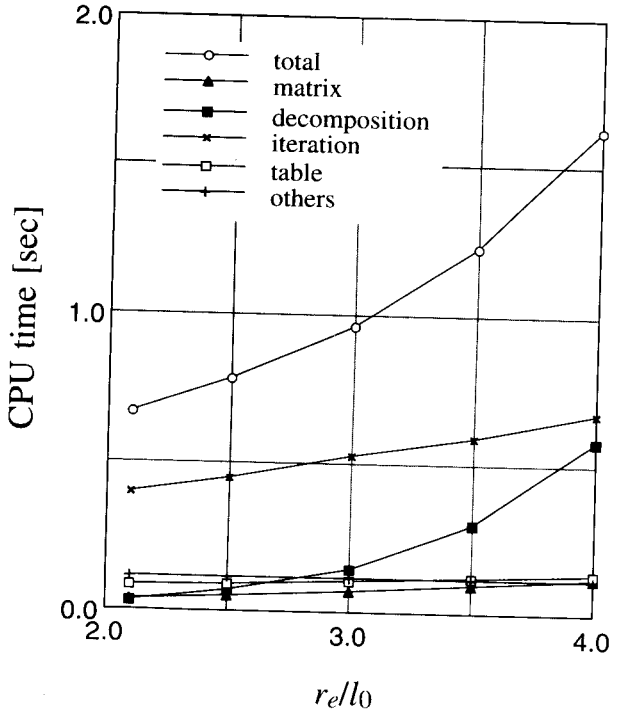


Fig. 7. Computation time in one time step.

and the iteration procedure. When the kernel size increases, the CPU time of the decomposition becomes relatively larger because the CPU time of the decomposition and the iteration is in proportion to $N_P N_B^2$ and $N_P^{1.5} N_B^{0.5}$, respectively, where N_P is the total number of particles and N_B is the number of neighboring particles for interactions. The number of neighboring particles is in proportion to the square of the kernel size in two dimensions. Generation of the table, where the neighboring particles are listed, is supposed to be dominant in a large problem involving 10^6 particles since this routine is in proportion to N_P^2.

The old kernel function used in Refs. 20, 21, and 22 is also tested. This function is

$$w(r) = \begin{cases} -(2r/r_e)^2 + 2 & (0 \le r < \frac{1}{2} r_e) \\ (2r/r_e - 2)^2 & (\frac{1}{2} r_e \le r < r_e) \\ 0 & (r_e \le r) \end{cases} . \quad (20)$$

The most important difference with the current kernel function [Eq. (3)] is that the value is finite at $r = 0$. The number of particles on the free surface is shown in Fig. 8. One case of the present kernel function [Eq. (3)] is also included in the figure. The calculation is successful when $1.4 l_0 \le r_{en} = r_{eLap} \le 1.7 l_0$. This range is considerably narrow. When $r_{en} = r_{eLap} \ge 1.8$, the trajectory deviates upward from the correct one. For instance, the configuration of particles at 0.35 s with $r_{en} = r_{eLap} = 2.0 l_0$ is shown in Fig. 9. We can see that particles are clustered everywhere. With Eq. (20), two particles can occupy the same position, satisfying the incompressibility condition if $2w(0) < \beta n_0$. These particles are regarded as the free surface, and no repulsive force acts between them. Therefore, particles are clustered. This trouble can be avoided if $w(0) = \infty$, such as in Eq. (3). Therefore, the calculation is more stable

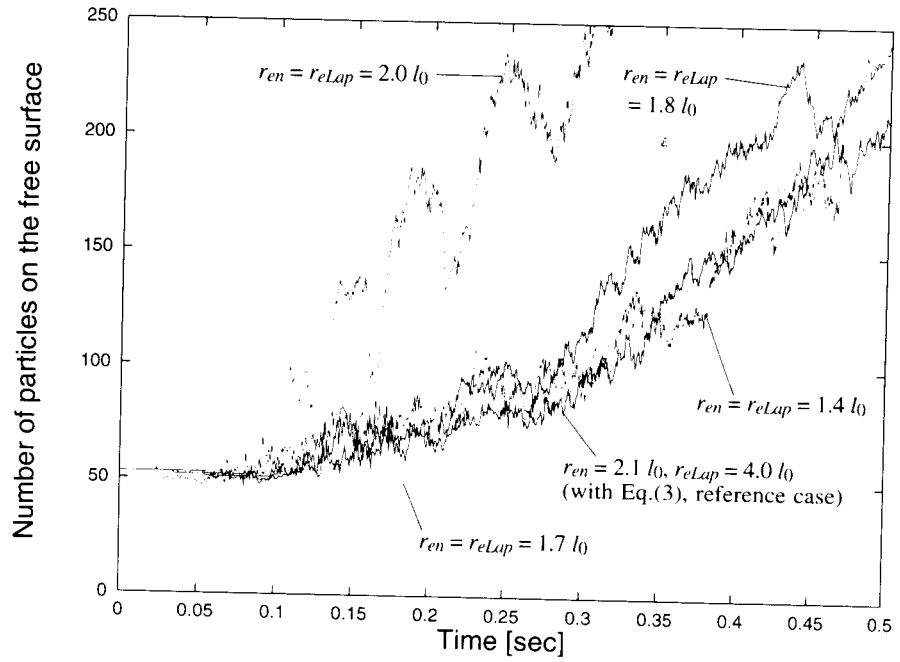


Fig. 8. Effect of kernel size used in Eq. (20).

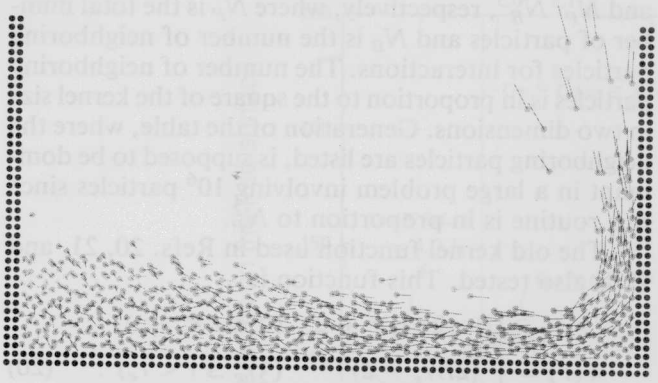


Fig. 9. Configuration of particles at 0.35 s with $r_{en} = r_{eLap} = 2.0l_0$ in Eq. (20).

TABLE I

Calculation Parameters

Kernel size	
Particle number density, r_{en}	$2.1l_0$
Laplacian, r_{eLap}	$4.0l_0$
Time step control	
Courant condition	$u_{max}\Delta t/l_0 \leq 0.2$
Maximum limit	$\Delta t \leq 10^{-3}$
Free surface parameter, β	0.97
Distance between neighboring particles in the initial configuration, l_0	8.0×10^{-3} m

with the current kernel function. Selected values of calculation parameters are summarized in Table I.

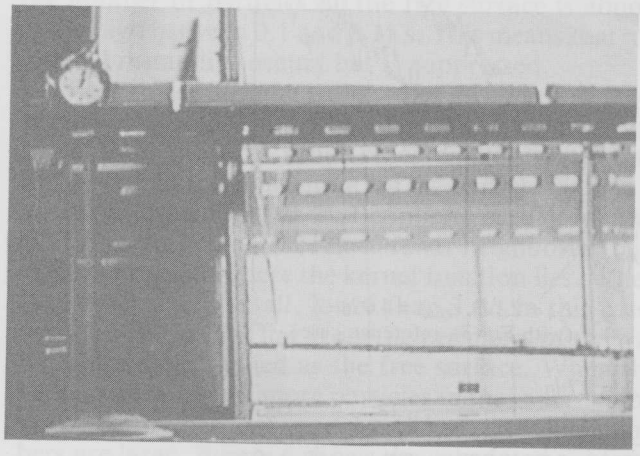
IV.C. Calculation Result and Comparison with the Experiment

Calculation of the collapse of a water column is carried out with the selected parameters listed in Table I. Viscosity and surface tension are neglected. In Fig. 10, configurations of the fluid are shown at 0.1-s intervals from 0.0 to 1.0 s for the experimental (the photographs) and calculated (the computer-generated graphs) results.

A removable board supports the initial water column in the experiment. This board is pulled up within 0.05 s and collapse starts. In the calculation, the water column is represented by 648 particles, which are located like a square grid. The distance between two neighboring particles l_0 is 8.0×10^{-3} m.

The left, right, and bottom walls are represented by 474 particles. Their coordinates are fixed, and velocities are zero. The particles on the inner first line of the walls are involved in the pressure calculation. As the source term of the incompressibility model, the particle number densities are needed at these particles. Thus, two other lines of particles should be added outside because $r_{en} = 2.1l_0$, otherwise the particle number densities are small and the wall particles are recognized as the free surface. In MPS, the wall boundary is represented by arranging fixed particles. This is simpler than the grid methods.

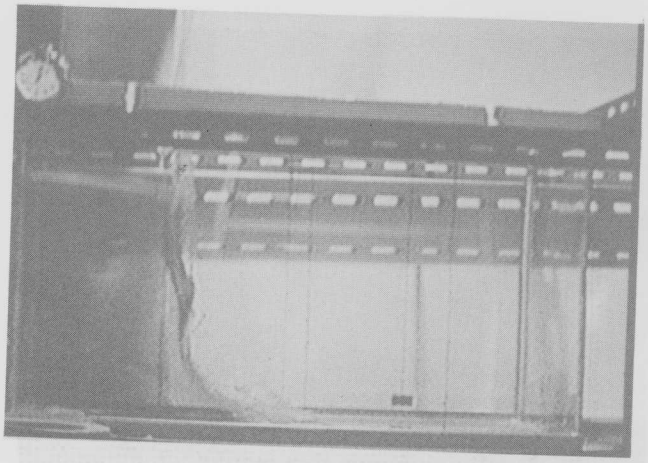
In Fig. 10, the flow velocity vectors are shown as lines from the particles' centers. The velocity scale is 10^{-2}, which means that velocity 1.0 m/s is shown by line length 10^{-2} m. At 0.1 s (Fig. 10b), the right surface of the water column is disturbed by the motion of the supporting board in the experiment. The collapsing



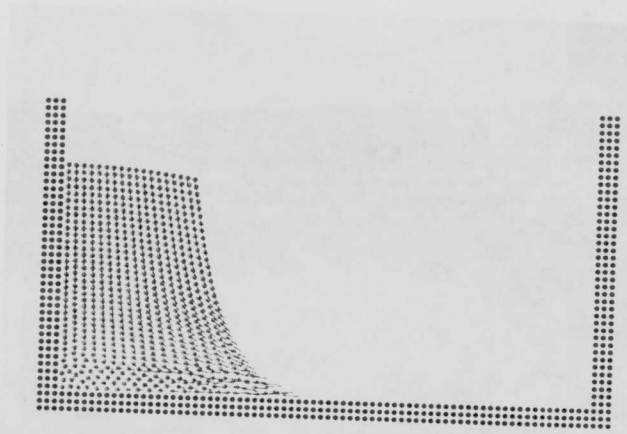
(a-1) T=0.0sec (experiment)

(a-2) T=0.0sec (calculation)

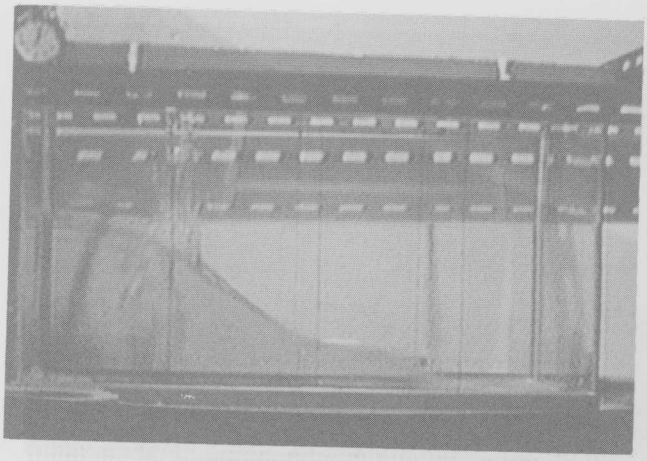
Fig. 10a. Comparisons between experimental and calculated collapse of a water column.



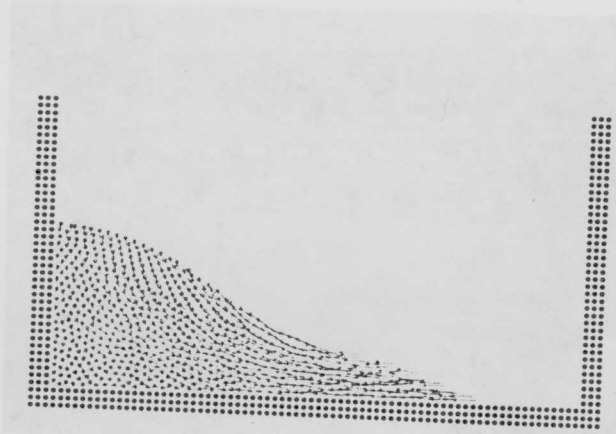
(b-1) T=0.1sec (experiment)



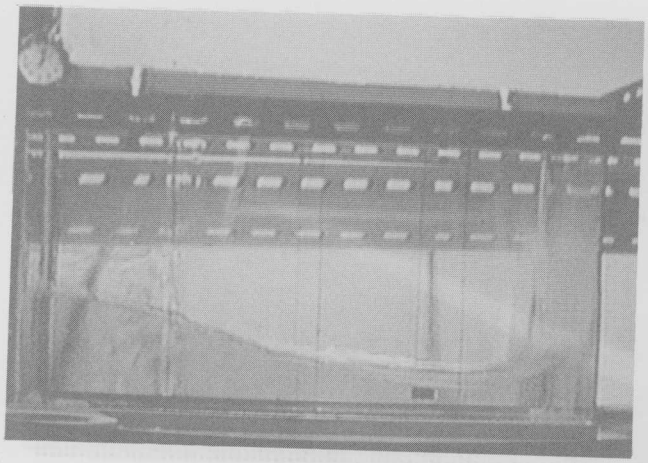
(b-2) T=0.1sec (calculation)



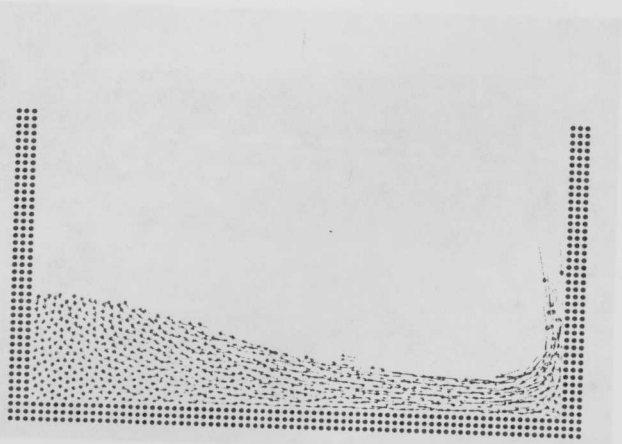
(c-1) T=0.2sec (experiment)



(c-2) T=0.2sec (calculation)



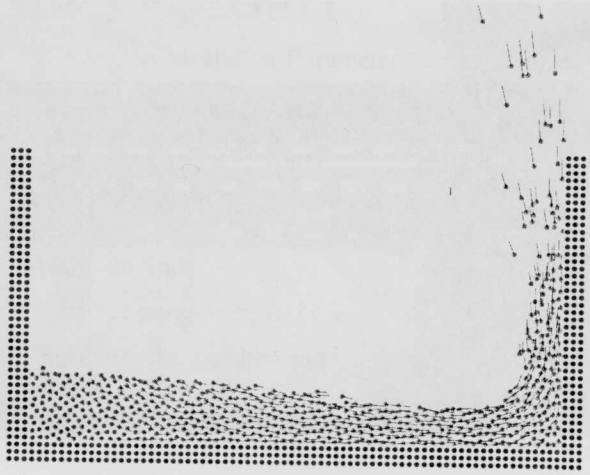
(d-1) T=0.3sec (experiment)



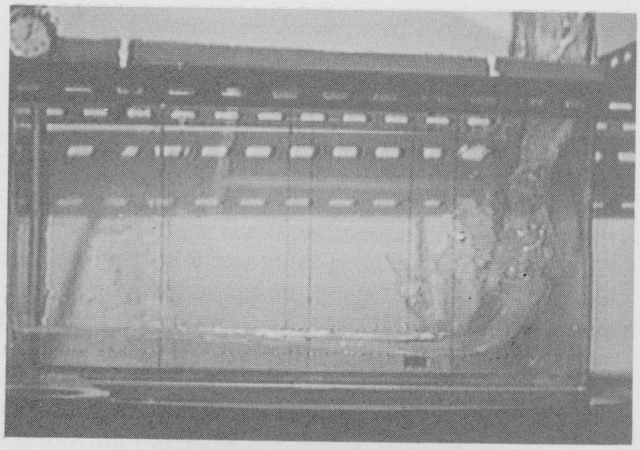
(d-2) T=0.3sec (calculation)

Figs. 10b, 10c, and 10d. Comparisons between experimental and calculated collapse of a water column.

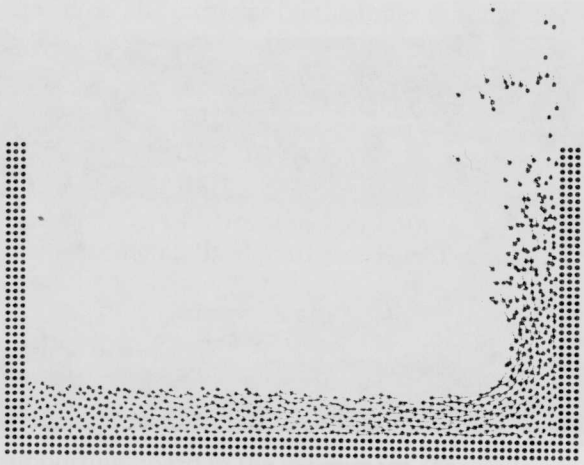
(e-1) T=0.4sec (experiment)



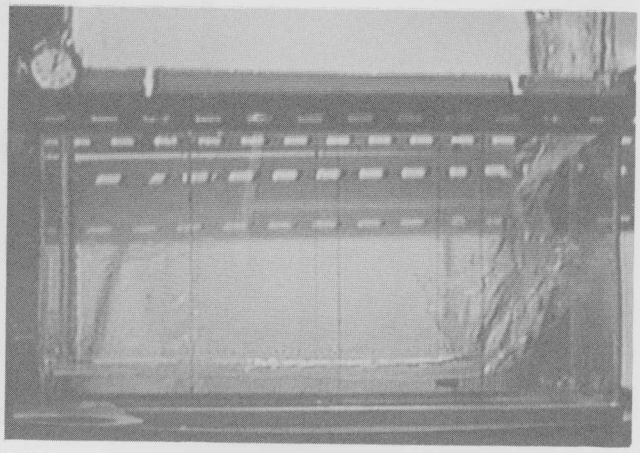
(e-2) T=0.4sec (calculation)



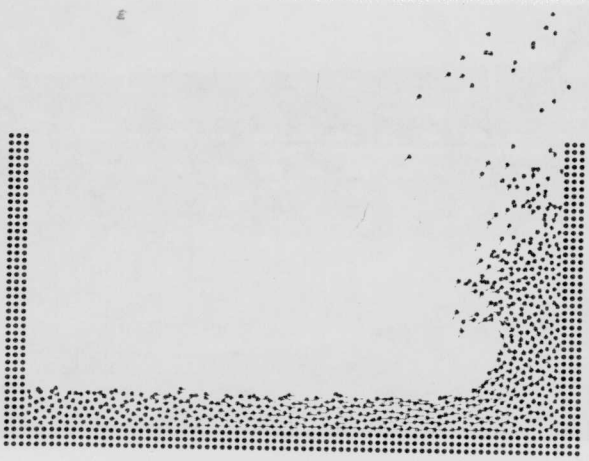
(f-1) T=0.5sec (experiment)



(f-2) T=0.5sec (calculation)

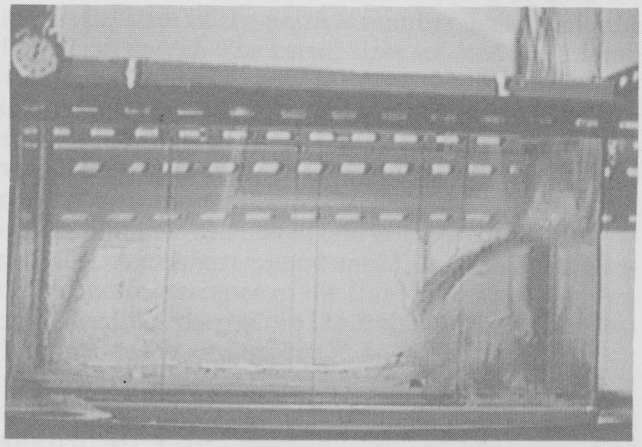


(g-1) T=0.6sec (experiment)

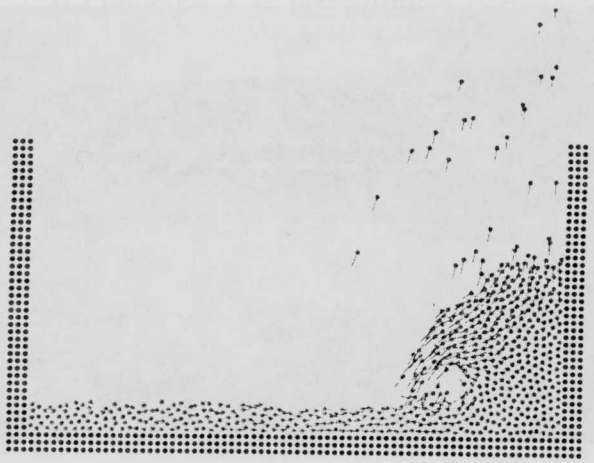


(g-2) T=0.6sec (calculation)

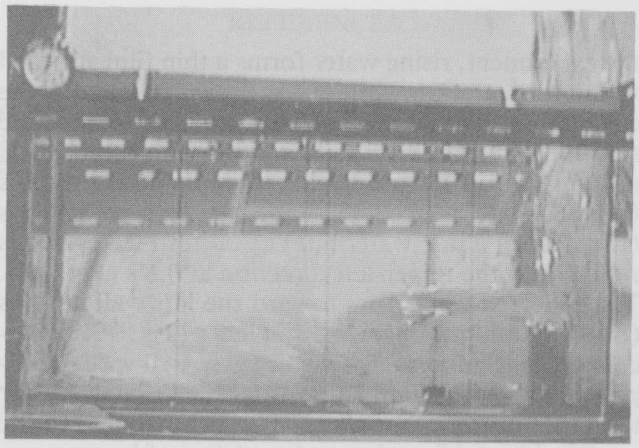
Figs. 10e, 10f, and 10g. Comparisons between experimental and calculated collapse of a water column.



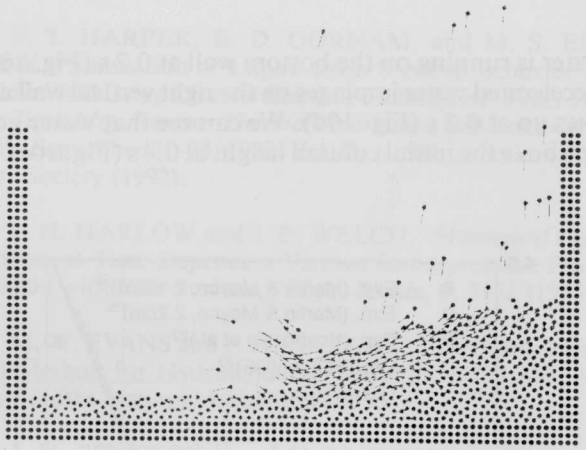
(h-1) T=0.7sec (experiment)



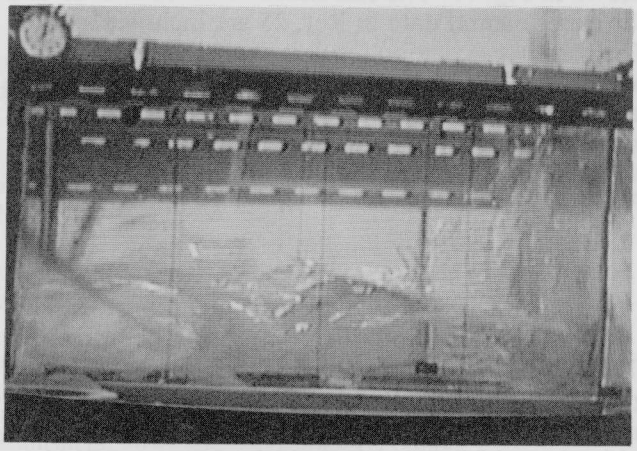
(h-2) T=0.7sec (calculation)



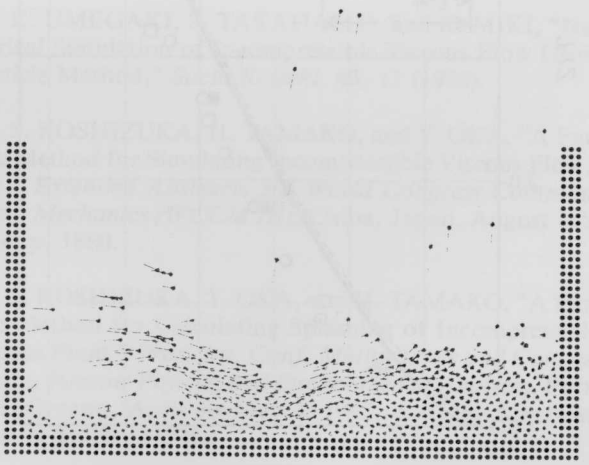
(i-1) T=0.8sec (experiment)



(i-2) T=0.8sec (calculation)



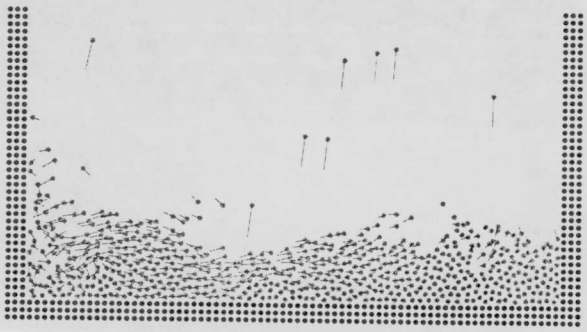
(j-1) T=0.9sec (experiment)



(j-2) T=0.9sec (calculation)

Figs. 10h, 10i, and 10j. Comparisons between experimental and calculated collapse of a water column.

(k-1) T=1.0sec (experiment)



(k-2) T=1.0sec (calculation)

Fig. 10k. Comparisons between experimental and calculated collapse of a water column.

water is running on the bottom wall at 0.2 s (Fig. 10c). Accelerated water impinges on the right vertical wall and rises up at 0.3 s (Fig. 10d). We can see that water goes up above the initial column height at 0.4 s (Fig. 10e). In the experiment, rising water forms a thin film attached to the vertical wall, while it is more fragmented in the calculation. The particle size is too large to simulate this thin film. The rising water is losing its momentum at 0.5 s (Fig. 10f) and begins to come down at 0.6 s (Fig. 10g). A mushroom shape can be seen at 0.7 s (Fig. 10h). Falling water hits the bottom water and is reflected in the upper-left direction at 0.8 s (Fig. 10i). The reflected water flies toward the left wall at 0.9 s (Fig. 10j) and impinges on it at 1.0 s (Fig. 10k). Calculated configurations of water agree well with the experimental results until 1.0 s. Agreement is much better than that in Refs. 21 and 22. This is attributed to the modified kernel function and modeling of incompressibility in MPS.

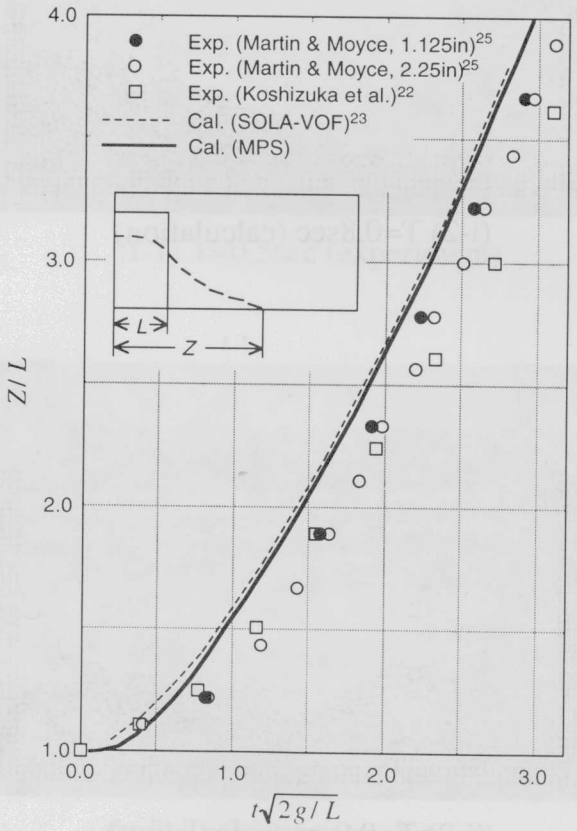


Fig. 11. Motion of the leading edge in collapse of a water column.

Motion of the leading edge, which is the front of the collapsing water running on the bottom wall, is shown in Fig. 11. The calculated result in Ref. 23 and the experimental data in Ref. 25 are included as well. The current calculation result is almost the same as that of SOLA-VOF. In the experiments, the speed of the leading edge is slower than that of the calculations. This is due to the friction between the fluid and the bottom wall. We can see that the leading edge is rounded in Fig. 10c-1.

V. CONCLUSIONS

The MPS method for analyzing incompressible flows with free surfaces has been developed. Pressure gradient, diffusion, incompressibility, and free surfaces are modeled by using deterministic particle interactions with a kernel function. As a result of the modified kernel function and incompressibility model in MPS, numerical stability and computation time are markedly improved. Calculation parameters are investigated with

test calculations. Different values should be used for the kernel sizes of the particle number density and the Laplacian model. The kernel sizes selected are 2.1 and $4.0l_0$ for the particle number density and the Laplacian model, respectively. Parameter β used in the free surface model is not sensitive to the result when it is within the range of 0.8 to 0.99; 0.97 is selected for this study. With these selected parameters, collapse of a water column is calculated and compared with the experiment. Agreement is good until 1.0 s, even if fragmentation and coalescence of the fluid take place. Motion of the leading edge before the fragmentation is almost the same as the result of SOLA-VOF.

If other physical processes that have been solved by using grids are modeled as particle interactions and added to MPS, application of MPS will be wider. For instance, particle interaction models representing surface tension and phase change are necessary for many problems of multiphase flows.

REFERENCES

1. E. MEIBURG, "Comparison of the Molecular Dynamics Method and the Direct Simulation Monte Carlo Technique for Flows Around Simple Geometries," *Phys. Fluids*, **29**, 3107 (1986).

2. G. A. BIRD, "Direct Simulation of High-Vorticity Gas Flows," *Phys. Fluids*, **30**, 364 (1987).

3. U. FRISCH, D. D'HUMIERES, B. HASSLACHER, P. LALLEMAND, Y. POMEAU, and J.-P. RIVET, "Lattice Gas Hydrodynamics in Two and Three Dimensions," *Complex Sys.*, **1**, 649 (1987).

4. B. J. DALY, F. H. HARLOW, J. E. WELCH, E. N. WILSON, and E. E. SANMANN, "Numerical Fluid Dynamics Using the Particle-and-Force Method," LA-3144, Los Alamos National Laboratory (1965).

5. R. A. GINGOLD and J. J. MONAGHAN, "Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics," *J. Comput. Phys.*, **46**, 429 (1982).

6. J. J. MONAGHAN and R. A. GINGOLD, "Shock Simulation by the Particle Method SPH," *J. Comput. Phys.*, **52**, 374 (1983).

7. J. J. MONAGHAN, "An Introduction to SPH," *Comput. Phys. Commun.*, **48**, 89 (1988).

8. W. BENZ, "Smoothed Particle Hydrodynamics: A Review," *The Numerical Modeling of Nonlinear Stellar Pulsations—Problems and Prospects*, p. 269, Kluwer Academic Publishers, Norwell, Massachusetts (1989).

9. E. A. C. HENNEKEN and V. ICKE, "SPH Faces Emery's Jump," *Comput. Phys. Commun.*, **74**, 239 (1993).

10. L. D. LIBERSKY, A. G. PETSCHEK, T. C. CARNEY, J. R. HIPP, and F. A. ALLAHDADI, "High Strain Lagrangian Hydrodynamics—A Three-Dimensional SPH Code for Dynamic Material Response," *J. Comput. Phys.*, **109**, 67 (1993).

11. J. T. BATINA, "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Aircraft Applications," AIAA Paper 93-0333, American Institute of Aeronautics and Astronautics (1993).

12. T. BELYTSCHKO, Y. Y. LU, and L. GU, "Element-Free Galerkin Methods," *Int. J. Numer. Methods Eng.*, **37**, 229 (1994).

13. Y. Y. LU, T. BELYTSCHKO, and L. GU, "A New Implementation of the Element Free Galerkin Method," *Comput. Methods Appl. Mech. Eng.*, **113**, 397 (1994).

14. T. BELYTSCHKO, L. GU, and Y. Y. LU, "Fracture and Crack Growth by Element Free Galerkin Methods," *Modeling Simul. Mater. Sci. Eng.*, **2**, 519 (1994).

15. F. T. HARPER, E. D. GORHAM, and M. S. ELGENK, "Simulation of Liquid Drop Breakup Behavior in a Flow Field Using Discrete Element Techniques," *Proc. 5th Int. Topl. Mtg. Reactor Thermal Hydraulics*, Salt Lake City, Utah, September 21–24, 1992, Vol. II, p. 605, American Nuclear Society (1992).

16. F. H. HARLOW and J. E. WELCH, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Phys. Fluids*, **8**, 2182 (1965).

17. M. W. EVANS and F. H. HARLOW, "The Particle-in-Cell Method for Hydrodynamic Calculations," LA-2139, Los Alamos National Laboratory (1957).

18. J. U. BRACKBILL and H. M. RUPPEL, "FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions," *J. Comput. Phys.*, **65**, 314 (1986).

19. K. UMEGAKI, S. TAKAHASHI, and K. MIKI, "Numerical Simulation of Incompressible Viscous Flow Using Particle Method," *Suchi Kaiseki*, **43**, 17 (1992).

20. S. KOSHIZUKA, H. TAMAKO, and Y. OKA, "A Particle Method for Simulating Incompressible Viscous Flow," *Proc. Extended Abstracts 3rd World Congress Computational Mechanics (WCCM III)*, Chiba, Japan, August 1–5, 1994, p. 1880.

21. S. KOSHIZUKA, Y. OKA, and H. TAMAKO, "A Particle Method for Calculating Splashing of Incompressible Viscous Fluid," *Proc. Int. Conf. Mathematics and Computations, Reactor Physics, and Environmental Analyses*, Portland, Oregon, April 30–May 4, 1995, p. 1514, American Nuclear Society (1995).

22. S. KOSHIZUKA, H. TAMAKO, and Y. OKA, "A Particle Method for Incompressible Viscous Flow with Fluid Fragmentation," *Comput. Fluid Dynamics J.*, **4**, 29 (1995).

23. C. W. HIRT and B. D. NICHOLS, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," *J. Comput. Phys.*, **39**, 201 (1981).

24. B. RAMASWAMY and M. KAWAHARA, "Lagrangian Finite Element Analysis Applied to Viscous Free Surface Fluid Flow," *Int. J. Numer. Methods Fluids*, **7**, 953 (1987).

25. J. C. MARTIN and W. J. MOYCE, "An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane," *Philos. Trans. R. Soc. London Ser. A*, **244**, 312 (1952).

26. W. MASCHEK, C. D. MUNZ, and L. MEYER, "Investigations of Sloshing Fluid Motions in Pools Related to Recriticalities in Liquid-Metal Fast Breeder Reactor Core Meltdown Accidents," *Nucl. Technol.*, **98**, 27 (1992).

27. C. HIRSCH, *Numerical Computation of Internal and External Flows*, John Wiley & Sons, New York (1988).