Numerical method for interaction between multiparticle and complex structures

Kensuke Yokoi

Mathematics Department, University of California, Los Angeles, California 90095-1555, USA and Department of Electronics and Mechanical Engineering, Chiba University, Chiba, 263-8522, Japan

(Received 28 April 2005; published 28 October 2005)

We propose a numerical method for dealing with interactions between multiple particles and complex structures. In the method, the structures are represented on a grid by using the level set method. The interactions of particles and structures are calculated by a method based on the discrete element method. The method can treat the interaction between multiparticle and complex structures robustly.

DOI: 10.1103/PhysRevE.72.046713 PACS number(s): 02.70.Bf, 83.10.Pp, 83.10.Rs, 95.75.Pq

I. INTRODUCTION

Phenomena with particles and structures appear in various research fields such as physics, engineerings, and geophysics. However, numerical studies on the phenomena are difficult because interaction between complex structures and multiparticles must be taken into account. In the paper, we propose a numerical method to study these phenomena.

To simulate interaction among particles, the discrete element method (DEM) has been widely used [1–5]. In the DEM, particles overlap during collision and the dynamics is defined through the force acting on the collision particles. The DEM can deal with many-body collisions and sustained contact between particles. The present method is based on the DEM. So far, to compute interaction among particles and structures, the structures are expressed by particles [3,5], linear elements in the two-dimensional case and surface elements in the three-dimensional case [3,6,7]. In the paper, we propose a different approach based on the level set method [8–10,14]. In this paper, the particle shape is restricted to a circle.

In the present method, structures are represented by the zero level set (zero contour) of the level set function. Although the level set method is an interface capturing method, it is convenient to compute the interaction among particles and structures. The important features of the level set method are that the unit normal to the interface and the distance from the interface are well defined. These characteristics play an important role in coupling the DEM and the level set method.

The present method calculates beforehand the level set function for the structure on a grid. This means the distance from the structure and the normal vector of the structure are assigned on each grid point. Here, the normal vector is calculated as the gradient of the level set function. To compute collision between a structure and a particle, the distance between the particle and the structure, and the normal direction for the structure are required. This information can be obtained by interpolating the level set function at the center of the particle. If line elements or polygons are used to express the structure, complicated procedures are required such as computing smallest distances between the particle and elements. To calculate these, we must calculate distances between point (the particle center) and point (element vertex), and point and line (line element). In the three-dimensional case, additionally, the point-surface distance must be computed. However, in the present method these procedures are not required.

II. NUMERICAL METHOD

A. Particle-particle interaction

Equations for translational and rotational motion for a spherical particle are

$$m\frac{d^2\mathbf{r}}{dt^2}=\mathbf{F} \tag{1}$$

and

$$\mathbf{I}\frac{d\boldsymbol{\omega}}{dt}=\mathbf{T}, \tag{2}$$

where $\mathbf{r}$ is the position of the particle center, m the mass of particle, $\mathbf{F}$ the sum of all contact forces from other particles, ω the angular velocity, $\mathbf{T}$ the torque due to contact forces, and $\mathbf{I}$ the moment of inertia.

Contact forces between spherical particles are modeled by a linear spring, a dashpot, and a friction slider [1]. The normal interaction is expressed by a linear spring and a dashpot [Fig. 1(a)], and the tangential interaction is expressed by a linear spring, a dashpot, and a friction slider [Fig. 1(b)].

We consider two disks i and j of diameters d_i and d_j, with masses m_i and m_j, particle centers $\mathbf{r}_i$ and $\mathbf{r}_j$, velocities at

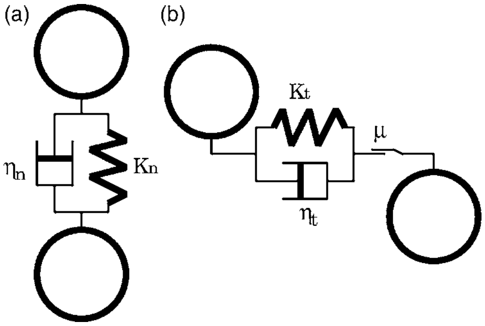

FIG. 1. Schematic figure of the discrete element method. K, η, and μ refer to the linear spring, the dashpot, and the slider, respectively.

mass center $\mathbf{c}_i$ and $\mathbf{c}_j$, and angular velocities ω_i and ω_j. The contact forces are calculated in contact, namely,

$$\Delta \equiv \frac{d_i + d_j}{2} - |r_{i,j}| > 0, \tag{3}$$

where $\mathbf{r}_{i,j} \equiv \mathbf{r}_i - \mathbf{r}_j$. The normal component of the contact force $F_n^{i,j}$ due to the particle j acting on the particle i is

$$F_n^{i,j} = 2Mk_n\Delta - 2M\eta_n v_n, \tag{4}$$

with

$$v_n = (\mathbf{c}_i - \mathbf{c}_j) \cdot \mathbf{n}, \tag{5}$$

$$\mathbf{n} = \frac{\mathbf{r}_{i,j}}{|\mathbf{r}_{i,j}|}, \quad \mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j, \tag{6}$$

where M is the reduced mass $[M = m_i m_j/(m_i + m_j)]$, k_n the spring constant, η_n the normal damping coefficient, v_n the normal component of the relative velocity, and $\mathbf{n}$ the unit normal. The tangential component of the contact force $F_s^{i,j}$ is

$$F_s^{i,j} = \min(|h_s^{i,j}|, \mu|F_n^{i,j}|)\mathrm{sgn}(h_s^{i,j}), \tag{7}$$

with

$$h_s^{i,j} = -2Mk_s u_s - 2M\eta_s v_s, \tag{8}$$

$$u_s = \int_{t_0}^{t} v_s dt, \tag{9}$$

$$v_s = (\mathbf{c}_i - \mathbf{c}_j) \cdot \mathbf{s} + \left(\frac{d_i}{2}\omega_i + \frac{d_j}{2}\omega_j\right), \tag{10}$$

where μ is the Coulomb friction coefficient for slider, k_s the tangential spring constant, u_s the tangential displacement, t_0 the time at the impact, η_s the tangential damping parameter, v_s the tangential velocity, and $\mathbf{s}$ the unit tangential vector. $\mathbf{F}_i$ and $\mathbf{T}_i$ are calculated as follows:

$$\mathbf{F}_i = \sum_j (F_n^{i,j} + F_s^{i,j}), \tag{11}$$

$$\mathbf{T}_i = \frac{d_i}{2}\sum_j (\mathbf{n} \times F_s^{i,j}). \tag{12}$$

As the time step, $\Delta t < 2\sqrt{2M/k_n}$ introduced in [1] is used.

B. Structure-particle interaction (level set method)

The interfaces of the structures are expressed by the level set method. The level set method is an interface capturing method and has been applied to various problems with interfaces [8–10]. This method expresses the surface of an $N-1$ dimension as a zero level (or contour) of an N-dimensional level set function ψ. The signed distance function

$$\psi = 0 \quad \text{at the interface,}$$

$$|\nabla\psi| = 1 \quad \text{for the whole region,} \tag{13}$$

is used as the level set function as shown in Fig. 2.

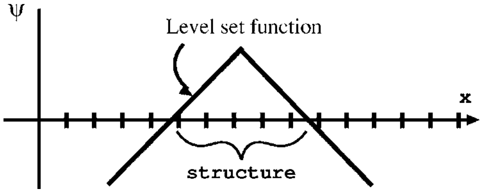

FIG. 2. Schematic figure of a level set function in the one-dimensional case.

In the paper, the level set method is used on a regular Cartesian fixed grid. Although the Cartesian fixed grid is used, the level set formulation can express subgrid information and complex geometries as shown in Figs. 2 and 3.

An advantage of the level set method is that the unit normal is always well defined from the level set function

$$\mathbf{n}_{ls} = \frac{\boldsymbol{\nabla}\psi}{|\boldsymbol{\nabla}\psi|}. \tag{14}$$

The unit normal is useful for computing interaction between particles and structures by using the distance function.

To construct the level set function for structures, we can use methods found in [9,11–13] such as the fast marching method for solving the eikonal equation

$$|\boldsymbol{\nabla}\psi| = 1. \tag{15}$$

The interaction between the structures expressed by the level set function and particles are computed based on the DEM. To compute the interaction, information about the distance from the interface and the normal direction to the interface are needed. In the level set formulation, this information is well defined. Therefore, the DEM is slightly modified by using the level set function ψ and the $\mathbf{n}_{ls}$ of Eq. (14). The procedures of DEM are replaced as follows:

$$(3) \Rightarrow \ \Delta \equiv \frac{d_i}{2} - |\psi| > 0, \tag{16}$$

$$(4) \Rightarrow \ F_n^{i,ls} = k_n\Delta - \eta_n v_n, \tag{17}$$

$$(5) \Rightarrow \ v_n = \mathbf{c}_i \cdot \mathbf{n}_{ls}, \tag{18}$$

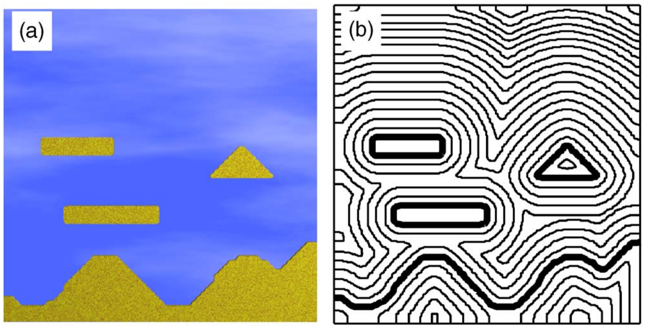

FIG. 3. (Color online) An example of a level set function in the two-dimensional case. (a) shows the shape of structures. (b) shows the contour lines of the level set function for the structure. The thick and thin lines represent the zero level set and the contour lines of the level set function. A 70×70 Cartesian grid is used.

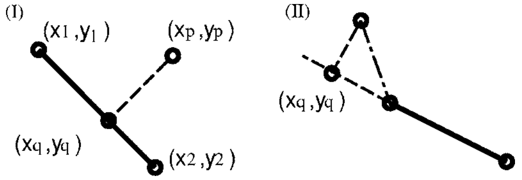

FIG. 4. Configuration to compare computation times.

$$(6) \Rightarrow \ \mathbf{n}_{ls} = \mp \frac{\nabla \psi}{|\nabla \psi|}, \qquad (19)$$

$$(7) \Rightarrow \ F_s^{i,ls} = \min(|h_s^{i,ls}|, \mu |F_n^{i,ls}|)\mathrm{sgn}(h_s^{i,ls}), \qquad (20)$$

$$(8) \Rightarrow \ h_s^{i,ls} = - k_s u_s - \eta_s v_s, \qquad (21)$$

$$(9) \Rightarrow \ u_s = \int_{t_0}^{t} v_s dt, \qquad (22)$$

$$(10) \Rightarrow \ v_s = \mathbf{c}_i \cdot \mathbf{s}_{ls} + \frac{d_i}{2}\omega_i; \qquad (23)$$

here ψ_i is ψ at the position of particle center. ψ_i is estimated by using an interpolation. The bilinear interpolation is used in this paper. In Eq. (19), the + sign is used for $\psi_{structure} > 0$ and the − sign is used for $\psi_{structure} < 0$. As the time step for the interaction between a particle and level set function, $\Delta t < 2\sqrt{m_i/k_n}$ is used.

III. COMPUTATION TIMES

In this type of simulation, the calculation of the minimum distance or contact detection is dominant. To compare computational time between the present method and the traditional direct method using elementary Euclidean geometry, we compare the computational time to compute the minimum distance between a point and a line element as shown in Fig. 4.

In the present method, the minimum distance between the point and the line element is calculated as follows:

$$i = \mathrm{int}\left(\frac{x_p}{\Delta x}\right), \quad j = \mathrm{int}\left(\frac{y_p}{\Delta y}\right), \qquad (24)$$

$$X = \frac{x_p - i\Delta x}{\Delta x}, \quad Y = \frac{y_p - j\Delta y}{\Delta y}, \qquad (25)$$

$$d_{min} = (1-X)(1-Y)\psi_{i,j} + X(1-Y)\psi_{i+1,j} + (1-X)Y\psi_{i,j+1} + XY\psi_{i+1,j+1}, \qquad (26)$$

where x_p and y_p are the position of the point, i and j are the index of the cell including the point (x_p, y_p), int(a) means the integer part of a, ψ is the level set function, and d_{min} is the minimum distance. The cell including the point is found by Eq. (24). The minimum distance is computed by using the bilinear interpolation [Eqs. (25) and (26)].

In the traditional direct method, the minimum distance is calculated as follows:

$$a_x = x_2 - x_1, \quad a_y = y_2 - y_1, \qquad (27)$$

$$s = \frac{(x_p - x_1)a_x + (y_p - y_1)a_y}{a_x^2 + a_y^2}, \qquad (28)$$

$$x_q = x_p + s a_x, \quad y_q = y_p + s a_y, \qquad (29)$$

if $(\min(x_1,x_2) < x_q < \max(x_1,x_2)$ and $\min(y_1,y_2) < y_q < \max(y_1,y_2))$

TABLE I. Computation times to calculate the minimum distance between the line element and the point. Direct(I) is the result in the case of Fig. 4(I) that (x_q, y_q) is on the line element. Direct(II) is the result in the case of Fig. 4(II), that the distance between the point and either edge of the line element is the minimum distance.

Method	Present	Direct(I)	Direct(II)
Time	1	2.32	2.37

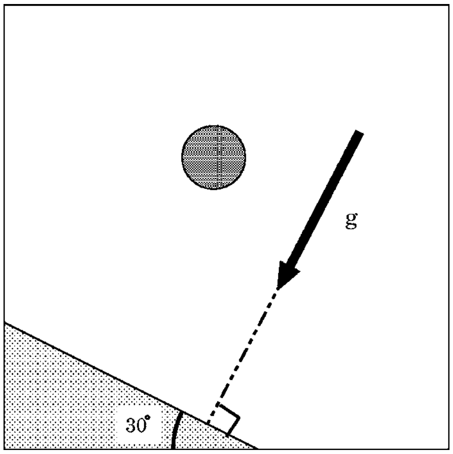

FIG. 5. Schematic figure of a test problem.

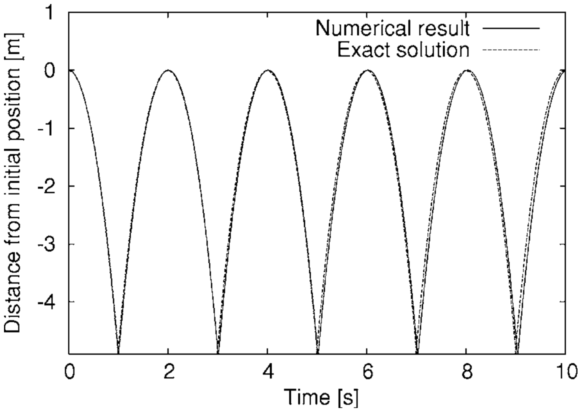

FIG. 6. Time evolution of the height of the particle from the slope. The solid line and the dotted line represent the numerical result and exact solution, respectively. Five cycles are plotted. $k_n = 5 \times 10^7$ is used.

TABLE II. Error of our algorithm.

Time (s)	Error_1	Error_2
2	2.98×10^{-4}	6.08×10^{-5}
4	1.25×10^{-3}	2.55×10^{-4}
6	2.87×10^{-3}	5.86×10^{-4}
8	5.14×10^{-3}	1.05×10^{-3}
10	8.08×10^{-3}	1.65×10^{-3}

$$d_{min} = \sqrt{(x_q - x_p)^2 + (y_q - x_p)^2}, \tag{30}$$

else

$$d_{min} = \min(\sqrt{(x_1 - x_p)^2 + (y_1 - y_p)^2}, \sqrt{(x_2 - x_p)^2 + (y_2 - y_p)^2}), \tag{31}$$

where (x_1, y_1) and (x_2, y_2) are the positions of the edges of the line element as shown in Fig. 4. The parametric equation is used for the line. The distance between (x_q, y_q) and (x_p, y_p) is the minimum distance of (x_p, y_p) to an infinite line through (x_1, y_1) and (x_2, y_2). If (x_q, y_q) exists on the line element as shown in Fig. 4, the distance between (x_q, y_q) and (x_p, y_p) is the minimum distance [Eq. (30)]. If (x_q, y_q) is not on the line element as shown in Fig. 4, the distance between (x_p, y_p) and either (x_1, y_1) or (x_2, y_2) is the minimum distance [Eq. (31)].

We compare the computational time of these two cases on a computer (CPU Intel Pentium IV 2.4 GHz, operating system Vine Linux 2.4). The present method was about two times faster than the direct method as shown in Table I.

Additionally the present method does not depend on the number of elements. Even in the case of calculating the minimum distance among a point and M pieces of line element, the procedure and computation time are the same if the level set function has been computed based on all the elements. In the traditional formulation, the computational time depends on the number of the element M. If a sorting technique such as [7] is used (normally it is used), the computation time is reduced drastically. But we cannot write which one is better generally. It is because the time reduction by the sorting techniques strongly depends on situations like the choice of sorting algorithm, programming techniques, and particle distributions. However, if the same sorting technique is used for both the present method and the traditional method, the present method must be faster.

TABLE III. The result of resolution study I. Error_3 is defined as (numerical result)-(exact solution) at $t=2$ s.

Δx	Error_3
0.0125	3.76×10^{-4}
0.025	2.98×10^{-4}
0.05	3.40×10^{-4}
0.1	2.98×10^{-4}
0.2	3.91×10^{-4}
0.4	3.89×10^{-4}

TABLE IV. The results of resolution study II. Error_4 is defined as $\sqrt{(x_{present} - x_{direct})^2 + (y_{present} - y_{direct})^2}$ and computed at $t=1.02$ s (immediately after the collision) and $t=1.50$ s.

Δx	Error_4 ($t=1.02$ s)	Error_4 ($t=1.50$ s)
0.0125	7.19×10^{-2}	1.98×10^{-1}
0.025	6.95×10^{-2}	2.06×10^{-1}
0.05	8.25×10^{-2}	5.23×10^{-1}
0.1	6.73×10^{-2}	2.78×10^{-1}
0.2	7.42×10^{-2}	1.43×10^{-1}
0.4	1.27×10^{-1}	6.92×10^{-2}

Our method does have some defects. Our method requires additional internal memory to store the level set function for the structure. The accuracy in computing the interaction between particles and flat surface is rather worse because an interpolation is used to compute the distance. If a particle hits on the flat plate, the particle must be bounded for the direction. However in our formulation, the direction is rather dispersive depending on the grid resolution as reported in the section on convergence tests.

IV. VALIDATION

To certify the present method, we carried out a simple test problem. Figure 5 shows the configuration.

The slope is represented by the level set function on a Cartesian fixed grid of 50×50. The direction of gravity is perpendicular to the slope. In this test problem, damping and rotation of the particle are not taken into account. In this configuration, if the particle is released with the velocity$=0$ from height h for the slope, the particle must return to the same position at $t=2\sqrt{2h/g}$. The numerical result compared with the exact solution is shown in Fig. 6.

Table II displays the errors at $t=2$, 4, 6, 8 and s.

Error_1 and Error_2 are defined as $\text{Error}_1 = |(\text{numerical result}) - (\text{exact solution})|$ and $\text{Error}_2 = \text{Error}_1/h$.

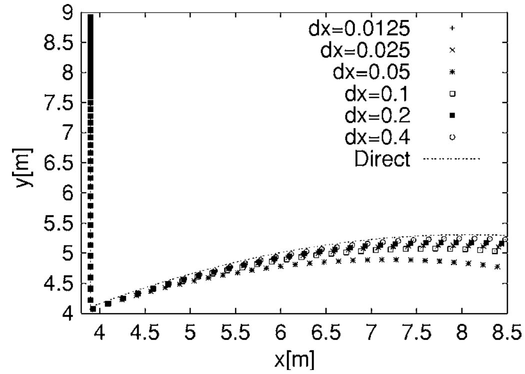

FIG. 7. Trajectory of the particle in the case of resolution study II. The configuration is the same as in Fig. 5 except for the gravity direction. The gravity direction is just straight down.

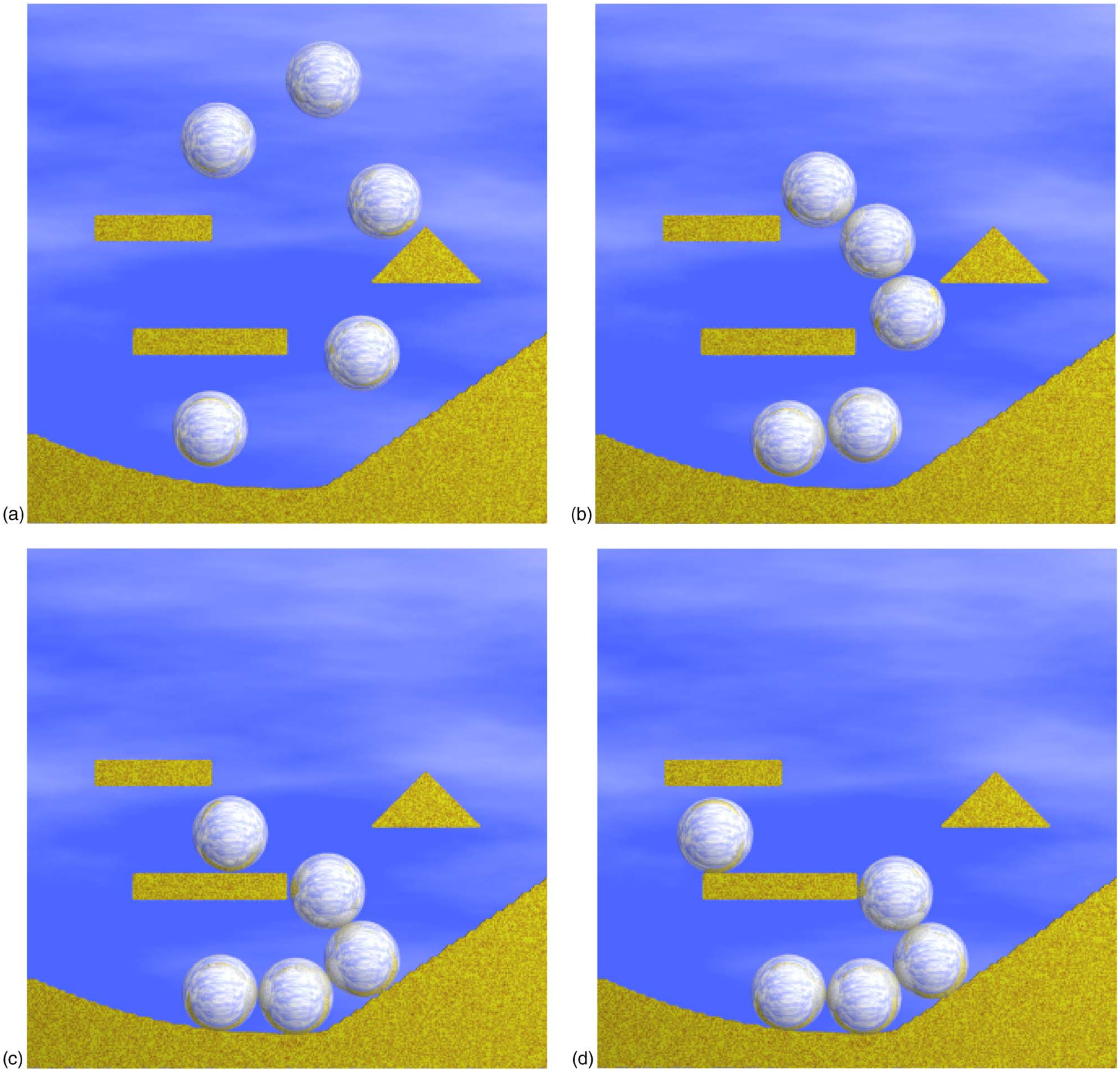

FIG. 8. (Color online) Snapshots of interaction between five particles and the structures at $t=0$, 1, 2, 3 s. The movie is available from [15].

V. CONVERGENCE TESTS

Resolution studies are done for the above problem (resolution study I) and for a problem where gravity is pointing straight down instead of in the normal direction as in the above problem (resolution study II). These results are shown in Tables III and IV.

Although we simulated six kinds of resolution for both cases, the dependence on gird sizes is not observed. This may be because the structure is flat and the level set function is a smooth function of $|\psi|=1$. Therefore the bilinear interpolation (26) gives almost the same answer. Although the error is small immediately after the collision in the case of resolution study II, the error becomes larger as time increases as shown in Fig. 7.

In this configuration, the numerical error of the collision calculation is dominant. The trajectory is just calculated by the Newton law with the Runge-Kutta method except for the collision computation. Therefore the increase of the error is a result of the slight error in the collision calculation.

VI. NUMERICAL RESULTS

We carried out numerical simulations in which the particles interact with structures in the gravity field. As a set of parameters, we use $k_n=5\times10^7$, $\eta_n=2\sqrt{k_n}$, $k_s=0.2k_n$, $\eta_s=\eta_n$, and $\mu=0.5$. A 140×140 grid is used for the level set function. Figure 8 shows the numerical results when five particles interact with structures.

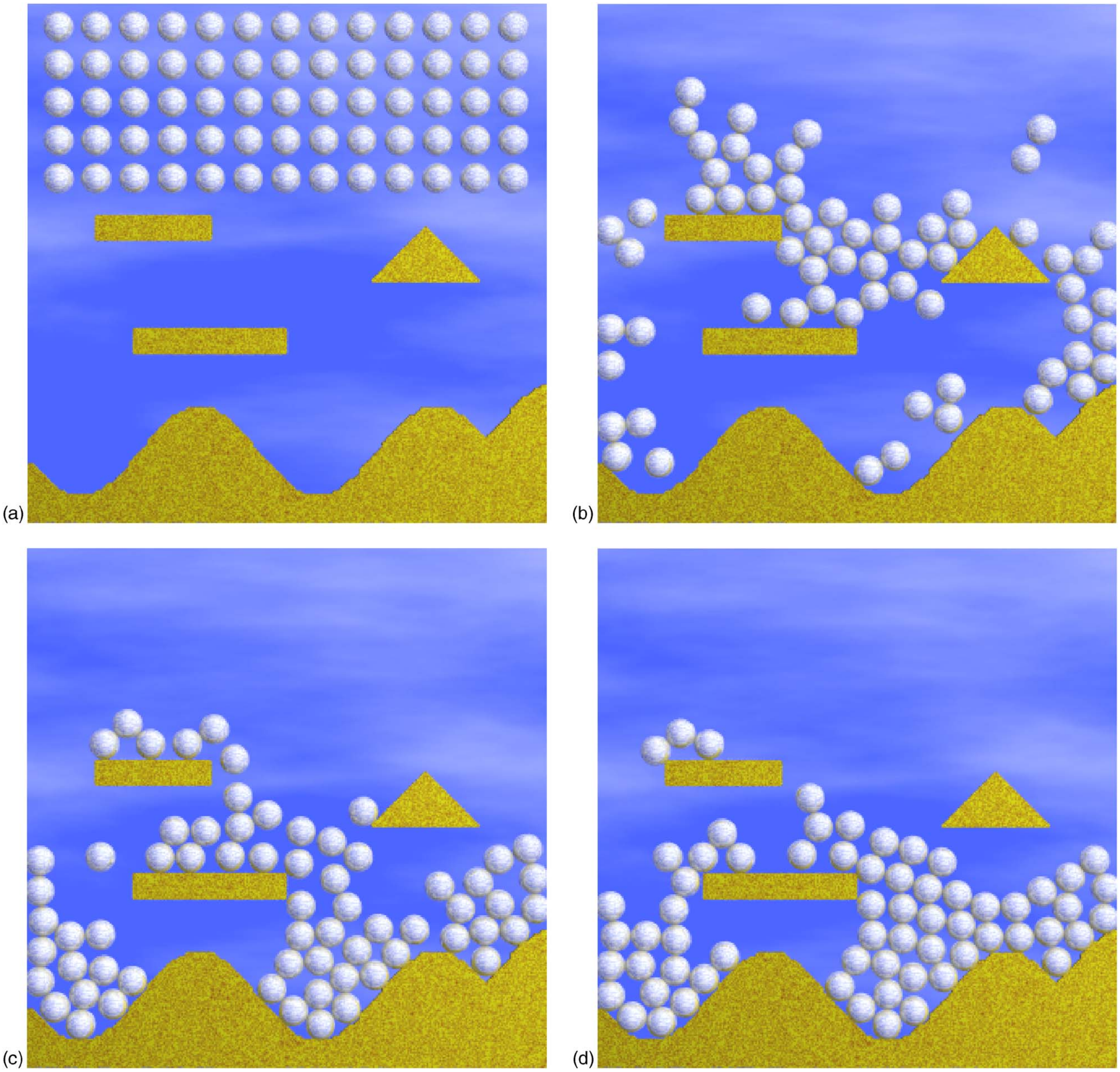

FIG. 9. (Color online) Snapshots of interaction between 65 particles and the structures at $t=0$, 1, 2, 3 s. The movie is available from [15].

A simulation with 65 particles is also performed, as shown in Fig. 9.

The results show that the method is robust.

VII. SUMMARY

We have proposed a numerical method based on the level set method and the DEM. The validity of the method has been shown by a test problem. The method can deal with interaction between multiple particles and complex structures robustly.

ACKNOWLEDGMENTS

I would like to thank Takayuki Aoki, Yousuke Imai, Rong Tian, Liu Hao, Feng Xiao, Jeremy Brandman, Stanley Osher, Igor Yanovskiy, Jason Chung, and Jinjun Xu for their useful discussions and comments. This work is supported in part by a Grant-in-Aid from Japan Society for the Promotion of Science (Grant No. 17-4003).

[1] P. A. Cundall and O. D. L. Strack, Geotechnique **29**, 47 (1979).

[2] C. S. Campbell, Annu. Rev. Fluid Mech. **22**, 57 (1990).

[3] Eng. Comput. 9(2) (1992), special issue, edited by G. Mustoe.

[4] K. Yamane, M. Nakagawa, S. A. Altobelli, T. Tanaka, and Y. Tsuji, Phys. Fluids **10**, 1419 (1998).

[5] N. Mitarai and H. Nakanishi, Phys. Rev. E **67**, 021301 (2003).

[6] A. S. Glassner, *Graphics Gems I* (Academic Press, New York, 1990).

[7] B. V. Mirtich, Ph.D. thesis, Computer Science Department, University of California at Berkeley, 1996 (unpublished).

[8] S. Osher and J. A. Sethian, J. Comput. Phys. **79**, 12 (1988).

[9] J. A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, Cambridge, U.K. (1999).

[10] S. Osher and R. Fedkiw, *Level Set Methods and Dynamics Implicit Surface*, Applied Mathematical Sciences, Vol. 153 (Springer, Berlin, 2003).

[11] D. Adalsteinsson and J. A. Sethian, J. Comput. Phys. **148**, 2 (1999).

[12] K. Yokoi, Phys. Rev. E **65**, 055701(R) (2002).

[13] Y. R. Tsai, J. Comput. Phys. **178**, 175 (2001).

[14] M. Sussman, P. Smereka, and S. Osher, J. Comput. Phys. **114**, 146 (1994).

[15] http://www.kensuke.biz