

GPU 기반 대량입자 거동 시뮬레이션과 실험비교

박상욱* · 전철웅* · 손정현**† · 이재욱***

* 부경대학교 대학원 메카트로닉스공학과, ** 부경대학교 기계자동차공학과, *** 한국생산기술연구원

Comparison of GPU-Based Numerous Particles Simulation and Experiment

Sang Wook Park*, Chul Woong Jun*, Jeong Hyun Sohn**†* and Jae Wook Lee***

* Graduate School of Mechatronics Engineering, Pukyong Nat'l Univ.,

** Department of Mechanical and Automotive Engineering, Pukyong Nat'l Univ.

*** Ultimate Manufacturing Technology R&BD Group, Korea Institute of Industrial Technology

(Received March 8, 2014 ; Revised May 23, 2014 ; Accepted May 24, 2014)

Key Words: Discrete Element Method(이산 요소법), Contact(접촉), Dam-Break(댐 붕괴), Parallel Programming(병렬프로그래밍), Particle Dynamics(입자동역학)

초록 우리 주변에는 수많은 알갱이들이 서로 상호작용하면서 운동하는 모습을 주로 볼 수 있다. 본 연구에서는 수많은 알갱이들의 접촉에 따른 동적 거동을 분석하였다. 접촉에 의한 각 요소의 거동을 해석하기 위해 이산 요소법(Discrete element method)을 사용하였다. 접촉 관계에 있어서 접촉 판별을 위해 Neighboring-Cell 알고리즘을 사용하였고, 접촉력 계산에 Hertzian contact model 과 Tangential sliding friction contact 모델을 사용하였다. 수많은 알갱이의 접촉해석을 위해 GPU 기반의 병렬 프로그램을 구성하였다. 해석을 검증하기 위해서 댐 붕괴 실험을 수행하였다. 거동 종료시점에 입자들의 쌓인 형상을 시뮬레이션 결과와 비교하여 해석 프로그램의 신뢰성을 검증하였다.

Abstract: The dynamic behavior of numerous grains interacting with each other can be easily observed. In this study, this dynamic behavior was analyzed based on the contact between numerous grains. The discrete element method was used for analyzing the dynamic behavior of each particle and the neighboring-cell algorithm was employed for detecting their contact. The Hertzian and tangential sliding friction contact models were used for calculating the contact force acting between the particles. A GPU-based parallel program was developed for conducting the computer simulation and calculating the numerous contacts. The dam break experiment was performed to verify the simulation results. The reliability of the program was verified by comparing the results of the simulation with those of the experiment.

1. 서 론

우리 주변에는 수 많은 작은 알갱이들이 서로 지속적으로 충돌하면서 운동하는 모습들을 어렵지 않게 볼 수 있다. 특히 모래 및 자갈로 이루어진 경사진 지형의 붕괴현상, 흙 길을 달리는 자동차와 상호작용하는 흙 및 모래 등을 대표적인 예로 볼 수 있다. 이에 대량 요소들 사이의 충돌에 의

한 거동을 수치적으로 해석하기 위해 많은 연구들이 수행 되어 왔다. 1971 년에 Cundall⁽¹⁾에 의해 제안된 Distinct element method(DE)를 시작으로, 1985 년 Hocking, Williams 와 Mustoe 에 의해 Generalized discrete element method,⁽²⁾ 그리고 1988 년에 Shi 에 의해 제안된 Discontinuous deformation analysis(DDA) 형태로 지속적으로 연구가 수행되어 왔다.⁽³⁾

최근에 와서는 정형화된 수치방법을 이용하여 상호 입자간의 접촉 판별 법 및 충돌시의 상호 침투량에 따른 접촉력 계산에 특히 관심을 둔 연구와 함께 실험을 통해 DEM 시뮬레이션의 결과를

† Corresponding Author, jhsohn@pknu.ac.kr

© 2014 The Korean Society of Mechanical Engineers

검증하는 연구가 이루어지고 있다.^(4,5) 또한 수많은 입자들을 다루기 때문에 상호 접촉 판별 및 적분 계산에 많은 계산량이 수반되어 계산시간을 단축시키는 방법으로 병렬 컴퓨팅이 적용이 되고 있다.⁽⁶⁾ 병렬 컴퓨팅의 방법으로 다수의 CPU 코어를 이용한 방법이 적용되었다.

최근에는 GPGPU(General Purpose computing on a GPU)의 개념이 관심을 받으면서 2007 년에 NVIDIA 에서CUDA(Compute Unified Device Architecture)를 발표함과 동시에 자사의 그래픽카드에 CUDA를 적용시켜 시중에 내놓음으로써 GPGPU기반의 병렬 컴퓨팅이 최근 많은 적용분야에서 적용되고 있다. GPGPU는 컴퓨터 그래픽스를 위한 계산만을 다루는 GPU를 사용하여 CPU가 전통적으로 취급했던 응용프로그램들의 계산을 수행하는 기술이다. 또한 NVIDIA는 소비자들에게 CUDA를 적용한 그래픽 카드를 제공하기 위해 지속적으로 향상된 성능의 그래픽 카드를 생산하고 있다. 뿐만 아니라 손쉽게 GPU를 이용한 병렬 프로그래밍을 위해 다양한 컴퓨팅 언어를 바탕으로 컴파일러(compiler), 디버거(debugger), 라이브러리(library) 그리고 프로파일러(profiler)를 제공하고 있다. 이에 GPGPU를 활용한다면 해석시간을 감소 시킬 수 있다.

DEM 시뮬레이션 분야에서 Metariver Technology Co., Ltd.는 입자 기반의 다중물리 CAE 해석에 CUDA 기술을 적용하고 있다. 또한 Toby 는 지형을 알갱이로 표현하고 궤도차량의 시뮬레이션에 CUDA 기술을 적용하였다.⁽⁶⁾

본 연구에서는 이산 요소법을 적용하여 많은 구형의 강체가 접촉하며 움직이는 거동을 시뮬레이션 하기 위한 코드를 만들었으며, 거동에 대한 검증을 위하여 실험 장치를 제작하고 실험을 수행하여 시뮬레이션 결과와 비교 검증하였다. 이전까지 연구는 거동에 대한 시뮬레이션을 주로 연구하였으나, 본 연구에서는 고속카메라를 사용하여 시간 별 입자들의 거동을 측정하고 시뮬레이션과 비교하였다. 실험은 대량의 구를 일정 공간에 쌓아두고, 한쪽 벽을 제거하여 입자들의 흐름을 분석하는 실험이다. 실험을 수행하여 시뮬레이션에 필요한 중요 파라미터를 결정하였고, 실험과 시뮬레이션을 비교 분석하였다.

2. GPU 기반 대량입자 거동 시뮬레이션

2.1 입자동역학 해석절차

이산 요소법(discrete element method)은 모래와 같은 질량 입자들의 운동을 계산하기 위한 수치법

로 사용되어 왔으며, 입자거동 시뮬레이션에 가장 좋은 방법으로 알려져 있다. 각 입자에 대한 운동 방정식은 다음과 같다.

$$m_i \dot{v}_i = f_i, \quad \dot{r}_i = v_i \tag{1}$$

$$I_i \dot{\omega}_i = T_i \tag{2}$$

식 (1), (2) 에서 $m_i, r_i, v_i, I_i, \omega_i, T_i$ 는 각각 질량, 위치, 속도, 관성모멘트, 각속도 그리고 토크를 나타낸다.

시간 $t + \Delta t$ 에서의 위치와 속도를 결정하기 위해 Velocity-Verlet 적분법을 사용하였으며 그 식은 (3)~(5)와 같이 나타낸다.⁽⁷⁾

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{f_i(t)\Delta t^2}{2m_i} \tag{3}$$

$$v_i(t + \Delta t) = v_i(t) + \frac{(f_i(t) + f_i(t + \Delta t))\Delta t}{2m_i} \tag{4}$$

$$\omega_i(t + \Delta t) = \omega_i(t) + \frac{I_i^{-1}(T_i(t) + T_i(t + \Delta t))\Delta t}{2} \tag{5}$$

Fig. 1 은 Velocity-Verlet 적분법을 사용한 DEM 시뮬레이션 순서도를 나타낸다. Fig. 1 의 it 은 적분 스텝을 나타내고, n_it 은 최종 적분 스텝을 나타낸다.

2.2 접촉 판별 및 접촉력 계산

입자간의 접촉을 판별하기 위해 Neighboring-cell⁽⁸⁾ 방법을 사용하였다. Neighboring-cell 법은 공간을 일정한 크기로 나누어 cell 이라는 공간을 만들고 Fig. 2 와 같이 각 cell 에 속하는 입자를 나열한다.

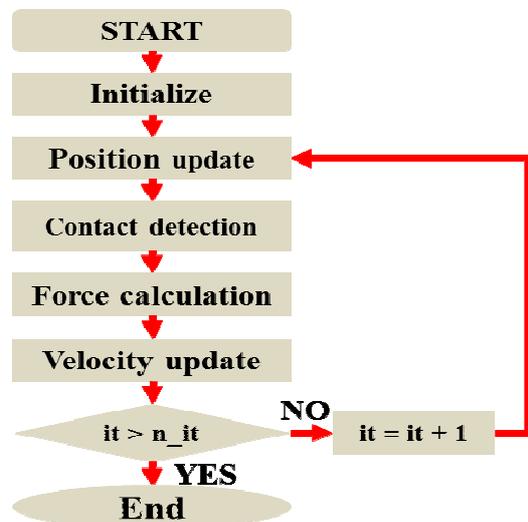


Fig. 1 Velocity-verlet algorithm

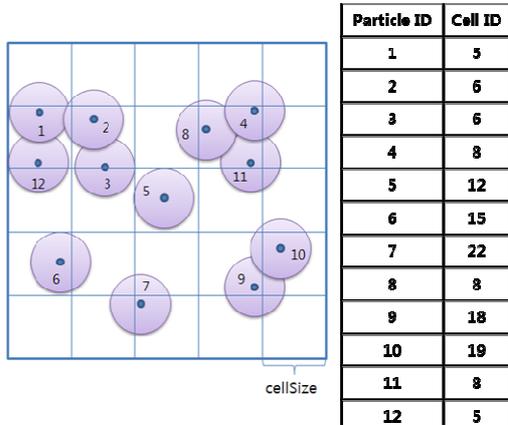


Fig. 2 The arrangement of particle in the grid-cell

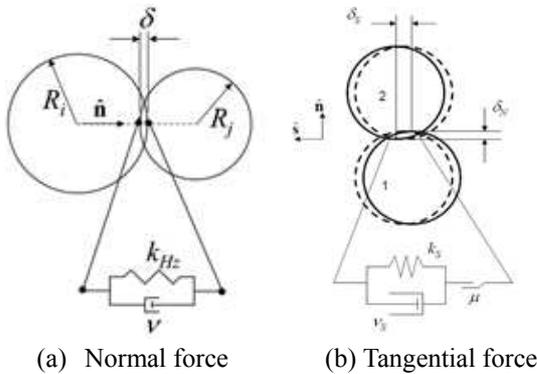


Fig. 3 Force model between two particles

나열된 배열은 cell의 ID를 기준으로 정렬하고 각 입자에 대해서 입자가 위치한 cell을 기준으로 바로 이웃한 cell에 존재하는 입자들과 접촉여부를 확인하여 접촉을 판별하고 침투여부를 파악하여 접촉력을 계산한다.⁽⁸⁾

Fig. 3의 (a)는 Hertzian 법선 방향 힘 계산 모델로 스프링 힘과 viscous 감쇠력을 표현하였고, (b)는 접선방향의 힘 계산 모델로 스프링 힘과 감쇠력 그리고 마찰계수로 표현된다. sliding friction 모델을 사용하였다.

$$\mathbf{f}_{ij}^e = \left(\frac{2}{3}\tilde{E}\sqrt{R_{eff}}h_{ij}^{3/2}\right)\hat{\mathbf{r}}_{ij} \quad (6)$$

$$\mathbf{f}_{ij}^v = -(\gamma_n\sqrt{R_{eff}}h_{ij}(\mathbf{v}_i - \mathbf{v}_j) \cdot \hat{\mathbf{r}}_{ij})\hat{\mathbf{r}}_{ij} \quad (7)$$

$$\mathbf{f}_{ij}^t = \min\{k_s\delta_s + v_s(\mathbf{v}_{ij}) \cdot \hat{\mathbf{s}}_{ij}, \mu_s|\mathbf{f}_{ij}^n|\}\cdot \hat{\mathbf{s}}_{ij} \quad (8)$$

$$\hat{\mathbf{s}}_{ij} = \frac{\Delta\mathbf{v}_t}{|\Delta\mathbf{v}_t|}, \Delta\mathbf{v}_t = (\mathbf{v}_{ij} - \mathbf{v}_{ij} \cdot \hat{\mathbf{n}}_{ij})\hat{\mathbf{n}}_{ij} \quad (9)$$

식 (6)은 법선 방향 스프링 힘을 나타내며, $\tilde{E} = E/(1 - \nu^2)$ 이고, E 는 영률, ν 는 푸아송비를 나타낸다. 식 (7)은 법선 방향 감쇠력을 나타낸다.

식 (8)에서 k_s , v_s 는 각각 접선 탄성 계수와 댐핑 계수를 나타낸다. 이고 두 입자 사이의 침투량은 $h_{ij} = R_i + R_j - |\mathbf{r}_{ij}|$ 이고, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ 이다.

그리고 $R_{eff} = R_i R_j / (R_i + R_j)$, $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij} / |\mathbf{r}_{ij}|$ 이다. $\hat{\mathbf{r}}_{ij}$ 은 접촉 평면에서 법선 방향의 단위 벡터이다. 식 (7)에서 γ_n 은 댐핑 계수 이다. 정지 마찰과 미끄럼 마찰은 접선방향의 스프링 힘에 의해 표현될 수 있다. 접선 방향의 힘은 (8)과 (9)와 같이 표현된다. 식 (8), (9)에서 $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ 이고, $\mathbf{f}_{ij}^n = \mathbf{f}_{ij}^e + \mathbf{f}_{ij}^v$ 이다. δ_s 은 접선 방향 스프링 변위 양을 나타내며, $\delta_s = |\Delta\mathbf{v}_t| \times \Delta t$ 로 계산된다. 입자를 회전시키는 토크는 다음과 같이 계산된다.

$$\mathbf{T}_i = (\mathbf{r}_i \cdot \hat{\mathbf{n}}) \times \mathbf{f}_i^t \quad (10)$$

2.3 CUDA 병렬 프로그래밍

DEM 시뮬레이션은 많은 작은 입자들이 상호작용하기 때문에 접촉 판별 및 적분 계산을 하는데 있어서 많은 시간을 요구한다. 이에 병렬 프로그래밍은 계산시간을 단축할 수 있는 좋은 방법이 될 수 있으며, GPGPU 기반의 병렬 프로그래밍의 한 방법인 CUDA 를 사용하여 계산 알고리즘을 병렬화 하였다. DEM 시뮬레이션은 반복계산에서 개별 입자가 독립적인 계산 형태를 지니기 때문에 특히 병렬화에 적합하다. 하나의 알고리즘에 대한 병렬 계산은 하나의 커널 함수(kernel function)에 의해 계산된다. 커널 함수 실행 시 필요한 블록(Block)과 스레드(Thread)의 수를 명시한다. 여기서 블록은 스레드의 집합이고 스레드는 커널 함수에 정의된 계산 과정을 처리하는 기본 계산 단위 이다. Fig. 4 는 커널 함수의 호출과 정의 부분을 나타내고 있다. 블록 수와 스레드 수는 기호 <<<...>>>에 의해 명시된다. 커널 함수는 `__global__`이라는 식별자로 구별되고 `index`는 0에서 `nThreads` 수 만큼의 각 스레드의 고유 번호를 나타내며 일련의 계산에 의해 주어진다. 입자 거동 해석은 각 입자에 독립적으로 계산되기 때문에 하나의 입자에 하나의 스레드를 할당하여 병렬 계산된다.

```
KernelFunction<<<nBlocks, nThreads>>>(arg1, arg2)
```

```
__global__ void KernelFunction(int* arg1, int* arg2)
{
    int index = __umul24(blockIdx.x, blockDim.x) + threadIdx.x;
    .....
}
```

Fig. 4 Call and definition of kernel function

Table 1 Simulation Parameter

Particles		Boundary	
Number	120,000	Size	900mm X400mm X500mm
Mass	0.2g		
Radius	3mm		
Young's modulus	3.25E+009 N/m ²	Young's modulus	3.2E+009 N/m ²
Poisson's ratio	0.34	Poisson's ratio	0.37
Density	12kg/m ³	Density	1185kg/m ³
Damping factor	6.4817	Damping factor	1.7586

Table 2 Material properties

Boundary	Material	Boundary Size		
	Acrylic	900x400x500(mm)		
Particles	Material	Number	Radius	Mass
	Polystyrene(BB)	120,000	3(mm)	0.2(g)

Table 3 Experiment case

Experiment	Case 1	Case 2	Case 3
Boundary distance	200mm	300mm	400mm
Initial Height	301mm	206mm	155mm

2.4 계수 결정

시뮬레이션의 경계조건은 실험과 동일한 900×400×500의 경계조건에서 각 경우 별로 200mm, 300mm, 400mm의 벽 경계를 시뮬레이션 시작 후 0.5 초에 제거하여 120,000여개의 볼들이 붕괴거동을 분석한다. 본 연구에서 사용되는 계수 값은 식 (6)~(9) 등에 사용되어지는 영률과 푸아송비, 밀도 그리고 탄성 계수값과 댐핑 계수값이 가장 중요한 파라미터이다.

Table 1는 시뮬레이션의 중요 변수를 나타낸다. 대량의 볼들의 변수는 실험에서 사용한 폴리스티렌(Polystyrene)의 물성치를 사용하였고, 마찬가지로 경계면 또한 실험에서 사용한 아크릴(Acrylic)의 물성치를 사용하였다. 탄성 계수는 각 요소의 영률(Young's modulus)과 푸아송비(Poisson's ratio)에 의해 결정된다. 댐핑 계수 γ_n 는 탄성계수, 질량 그리고 법선 반발계수에 의해 결정되며 그 식은 다음과 같다.

$$\gamma_n = \sqrt{\frac{4m'k}{1+\beta^2}} \tag{11}$$

$$\beta = \pi/\ln(\epsilon_N) \tag{12}$$

위 식에서 $m' = (m_i^{-1} + m_j^{-1})^{-1}$ 이며, k는 탄성 계수, ϵ_N 은 법선 반발계수이다. m_i 는 입자 i의 질량이며, m_j 는 입자 j의 질량이다.

3. 시뮬레이션과 실험의 비교

3.1 시뮬레이션 결과 비교

대형입자 거동 시험은 일정 공간에 플라스틱 재질의 작은 볼들을 쌓아두고 한쪽 벽을 제거한 후 볼들이 붕괴되는 거동을 분석하기 위해 수행하였

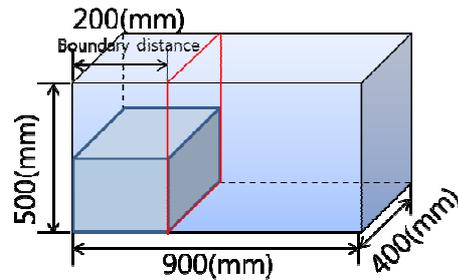


Fig. 5 Experimental equipment

다. 실험을 위해 아크릴재질의 900×400×500의 크기의 상자를 제작하였다. Table 2은 실험에서 사용한 상자 및 볼에 관한 물성치를 나타낸다.

Fig. 5의 200mm 부분은 플라스틱 구가 쌓여져 있는 공간으로써 총 120,000개의 구가 사용되었고, 구의 반지름은 3mm, 무게는 0.2g이다.

실험은 Table 3과 같이 3가지 경우에 대해서 각각 3번씩 실험을 수행하였다. Table 3의 Boundary distance는 Fig. 5에서 표시된 부분을 말하며, Initial Height는 초기 입자가 쌓여있는 높이를 말한다. 얻은 100mm의 단위 거리마다 최고점 높이를 컴퓨터 시뮬레이션 결과와 비교하였다. Fig. 6~8은 시뮬레이션 결과와 실험과의 데이터를 비교한 결과이다. 비교 결과에서 볼 수 있듯이 최대 오차는 8mm로 강체구 1개 크기의 차이를 보였다. 이 오차 값은 전체 붕괴 현상에서 매우 작은 오차로 볼 수 있으며, 붕괴 시뮬레이션이 실제 붕괴 현상에 사용할 수 있는 적합한 툴이라 할 수 있다.

Fig. 9는 실험과 시뮬레이션에서 칸막이를 제거 후 입자가 우측 벽에 도달하는 시점을 나타낸 결과로 최종적으로 쌓인 형상이 유사함을 확인할 수

있다. 실험에서 최초 0.49초에 우측벽에 강체구가 도달하였고, 시뮬레이션의 경우 0.54초에 우측벽에 입자가 도달 하였다.

Fig. 10 과 11 은 0.7 초와 0.9 초에서의 실험과 시뮬레이션을 비교한 것이다. 0.7 초에서의 실험과 시뮬레이션은 매우 유사함을 보였으며, 0.9 초에서는 실험에서의 입자들이 조금 더 우측벽에 도달하는 정도가 커짐을 볼 수 있었다.

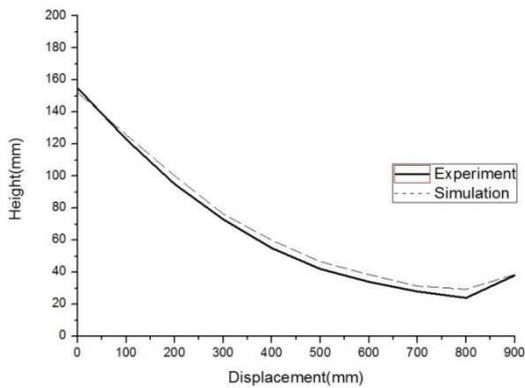


Fig. 6 Comparison of vertical disp. vs. horizontal disp. in case 1

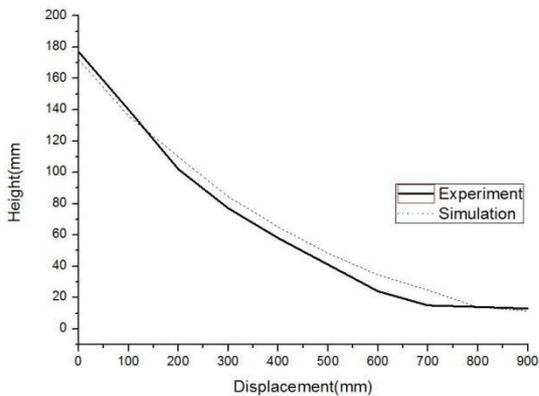


Fig. 7 Comparison of vertical disp. vs. horizontal disp. in case 2

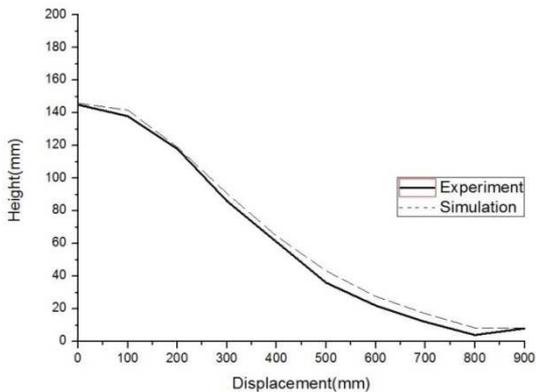


Fig. 8 Comparison of vertical disp. vs. horizontal disp. in case 3

3.2 해석 시간 비교

1 초 시뮬레이션을 수행하였을 때 CPU 기반 순차 해석시 소요시간과 GPU 기반 병렬 해석시 소요시간을 Fig. 9 에 나타내었다. 본 연구에 사용된 CPU 는 Intel core i7 이며, GPU 는 NVIDIA 사의 Tesla C2075 이다. CPU 의 코어는 4 개이고, GPU 의 코어 개수는 448 개 이다. 최대 계산시간을 기준으로 각 계산

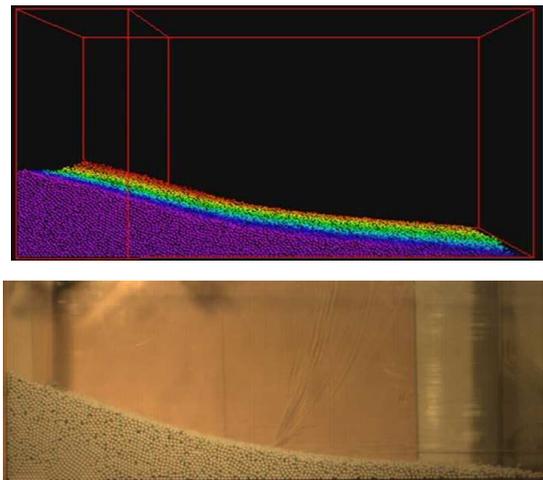


Fig. 9 Comparison of final configuration

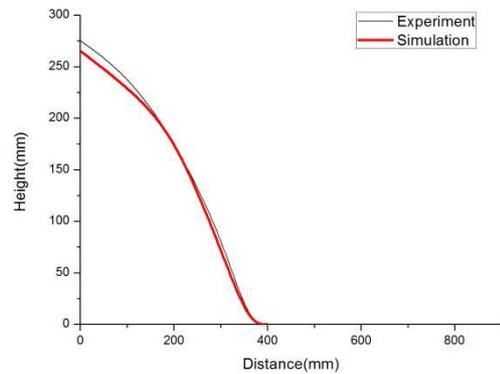


Fig. 10 Comparison of Experiment vs Simulation at 0.7s

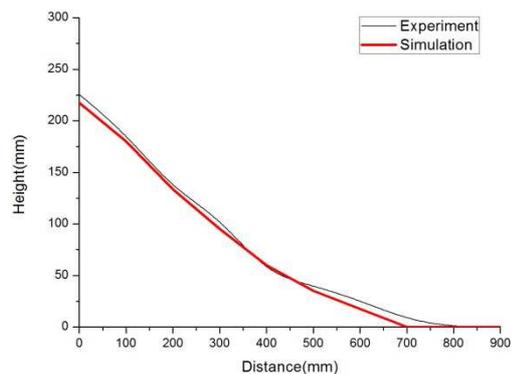


Fig. 11 Comparison of Experiment vs Simulation at 0.9s

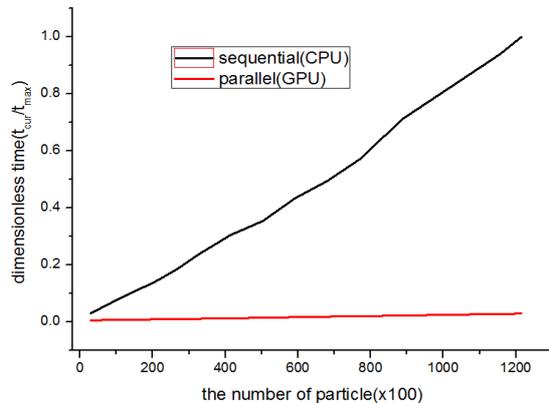


Fig. 12 Comparisons of computational time of sequential and parallel program

시간을 백분율화 하여 표현하였다. 입자의 수가 약 120,000개일 때 약 34배 가량 병렬 프로그램이 좋은 효율을 나타내었다.

4. 결론

본 연구에서는, 대량 입자의 접촉 거동을 해석하기 위해서 실험과 시뮬레이션을 수행하였다. 막이 벽의 거리에 따른 실험을 수행하여 입자들의 움직임을 분석하였고, 시뮬레이션과 비교하였다.

(1) 실험 결과 시뮬레이션과의 오차가 약 8mm 여서 상대적으로 적은 오차를 보였으며, 붕괴 시뮬레이션을 재현하는 데에 타당한 프로그램임을 보였다.

(2) CPU로 계산할 때와 GPU로 계산할 때의 계산 효율을 대량 입자들간의 접촉을 정의하고, 힘을 계산하는 과정과 계산 시간을 비교하였다. 입자의 수가 증가할수록 병렬화를 통한 효율성은 점점 높아짐을 확인 하였다.

향후 본 연구를 기초로하여 수많은 입자와 강체와의 접촉 해석 등을 연구할 계획이다.

후 기

이 논문은 부경대학교 자율창의학술연구비(2014년도)에 의하여 연구되었음

참고문헌 (References)

(1) Cundall, P. A. and Strack, O. D. L., 1979, "A Discrete

- Numerical Model for Granular Assemblies," *Geotechnique*, 29:47-65.
- (2) Williams, J. R., Hocking, G. and Mustoe, G. G. W., 1985, "The Theoretical Basis of the Discrete Element Method," *NUMETA '85*.
- (3) Shi, G. H., 1988, "Discontinuous Deformation Analysis - A New Numerical Model for the Statics and Dynamics of Block Systems," Ph.D. Thesis. Univ. of California, Berkeley, Calif.
- (4) André, D., Iordanoff, I., Luc Charles, J. and Néauport, J., 2012, "Discrete Element Method to Simulate Continuous Material by Using the Cohesive Beam Model," *Computer Methods in Applied Mechanics and Engineering* 213~216, 113~125
- (5) Datta, A., Mishra, B. K., Das, S. P. and Sahu, A., 2008, "A DEM Analysis of Flow Characteristics of Noncohesive Particles in Hopper," *Materials and Manufacturing Processes*, 23:195~202.
- (6) Heyn, T. D., 2009, "Simulation of Tracked Vehicles on Granular Terrain Leveraging GPU Computing," Department of Mechanical Engineering, University of Wisconsin- Madison.
- (7) Duong, N. H., 2011, "Modeling the Dynamics of Toothbrush Using Discrete Element Method," Kumoh National Institute of Technology.
- (8) Wassgren, C. and Sarkar, A., 2009, "Discrete Element Method (DEM) Course Module," PHARMAHUB, Purdue University, Lecture 7-9.
- (9) Mishra, B. K. and Murty, C. V. R., 2001, "On the Determination of Contact Parameters for Realistic DEM Simulations of Ball Mills," *Powder Technology*, Volume 115, Issue 3, 290-297.
- (10) Shabana, A. A., 2001, "Computational Dynamics 2Ed," Wiley-Interscience.
- (11) Green, S., 2010, "Particle Simulation Using CUDA," NVIDIA Corporation.
- (12) MSC Software, "ADAMS help," MSC Software.
- (13) Donze, F. V., Richefeu, V. and Magnier, S.-A., 2009, "Advances in Discrete Element Method Applied to Soil, Rock and Concrete Mechanics," in: *State of the art of geotechnical engineering*, *Electronic Journal of Geotechnical Engineering*, p. 44.
- (14) Tanaka, H., Oida, A., Daikoku, M., Inooku, K., Summikawa, O. and Nagasaki, Y., 2007, "DEM Simulation of Soil Loosening Process Caused by a Vibrating Subsoiler," *Agricultural Engineering International*, The CIGR Ejournal, Vol. 9, manuscript PM 05 010.