

다면체영역분할을 이용한 SPH의 충돌 및 병렬해석

The Contact and Parallel Analysis of Smoothed Particle Hydrodynamics (SPH) Using Polyhedral Domain Decomposition

탁 문 호[†]

Moonho Tak

Received: February 28th, 2024; Revised: February 29th, 2024; Accepted: March 11th, 2024

ABSTRACT : In this study, a polyhedral domain decomposition method for Smoothed Particle Hydrodynamics (SPH) analysis is introduced. SPH which is one of meshless methods is a numerical analysis method for fluid flow simulation. It can be useful for analyzing fluidic soil or fluid-structure interaction problems. SPH is a particle-based method, where increased particle count generally improves accuracy but diminishes numerical efficiency. To enhance numerical efficiency, parallel processing algorithms are commonly employed with the Cartesian coordinate-based domain decomposition method. However, for parallel analysis of complex geometric shapes or fluidic problems under dynamic boundary conditions, the Cartesian coordinate-based domain decomposition method may not be suitable. The introduced polyhedral domain decomposition technique offers advantages in enhancing parallel efficiency in such problems. It allows partitioning into various forms of 3D polyhedral elements to better fit the problem. Physical properties of SPH particles are calculated using information from neighboring particles within the smoothing length. Methods for sharing particle information physically separable at partitioning and sharing information at cross-points where parallel efficiency might diminish are presented. Through numerical analysis examples, the proposed method's parallel efficiency approached 95% for up to 12 cores. However, as the number of cores is increased, parallel efficiency is decreased due to increased information sharing among cores.

Keywords : SPH, Parallel analysis, Domain decomposition, Fluid analysis, Polyhedral domain decomposition, Contact and impact analysis

요 지 : 본 연구에서는 SPH 해석을 위한 다면체영역분할 기법이 소개된다. SPH 기법은 유체 유동 모사를 위한 수치해석기법으로 무요소기법(meshless method) 중 하나이다. 유동성 지반 또는 고체-유체 상호작용 해석 등에 유용하게 쓰일 수 있다. SPH는 입자기반 해석이기 때문에 입자가 많을수록 결과의 정확도는 높아지지만 수치적 효율성은 떨어진다. 일반적으로 해석의 효율성을 높이기 위해 병렬 프로세싱 알고리즘과 함께 쓰이는데 직교좌표계 기반의 영역분할 기법이 대표적이다. 그러나 복잡한 기하학적 형태나 동적 경계조건에서 유동 모사 등을 병렬 해석하기 위해서는 직교좌표계 영역분할 방법이 적합하지 않다. 소개하는 다면체영역분할 기법은 이와 같은 문제에서 병렬효율성을 높일 수 있는 장점을 갖는다. 다양한 형태의 3차원 다면체 요소로 분할하여 문제에 적합하게 모델링할 수 있다. SPH 입자들의 물리적 값들은 smoothing 길이 이내의 주위 입자들 정보를 이용하여 계산된다. 영역분할 시 물리적으로 분리될 수 있는 입자정보들을 코어간 공유할 수 있는 방법과 병렬효율성이 떨어질 수 있는 cross-point에서의 정보공유 방법이 소개된다. 수치해석 예제를 통하여 제안된 방법의 병렬효율성은 12코어까지 95%에 근접하였다. 이후 코어가 증가할수록 코어간 공유되는 정보량이 많아져 병렬효율성이 떨어지는 문제가 발생되기도 하였다.

주요어 : SPH, 병렬해석, 영역분할기법, 유동해석, 다면체영역분할, 충돌해석

1. 서 론

지반에 대한 수치해석은 격자기반의 유한요소해석(FEM)이 대표적이다. 이는 지반을 연속체로 가정하고 매크로 상태에서 재료 구성모델을 적용해 거동을 예측한다. 그러나 변형이 크거나 유동 현상이 있을 경우 격자기반의 수치해석은 격자의 왜곡현상으로 결과의 한계가 있다(Tak & Park,

2013). 컴퓨팅 해석 속도가 보장된다면 입자기반의 해석이 가장 좋은 대안이 될 수 있다. 지반을 모사하기 위한 입자기반 수치해석으로는 SPH(Smoothed Particle Hydrodynamics), DEM(Discrete Element Method) 등이 있다. 이 중 SPH는 연속체 역학을 기반으로 하고 있기 때문에 매크로의 점착력이 유지되고 있는 문제를 모사하기에 적합한 접근방법이다.

SPH 같은 입자기반의 수치해석에서는 개별 입자들마다

[†] Assistant Prof., Innovation Center for Engineering Education, College of Engineering, Hanyang University (Corresponding Author : pivotman@hanyang.ac.kr)

움직임이 계산되기 때문에 고효율의 컴퓨팅 파워가 요구된다. 계산 효율성을 높이기 위해 병렬해석 알고리즘이 필수적으로 고려되어야 하는데, 영역분할기법은 입자기반 해석에서 병렬효율성을 높인 알고리즘 중 하나이다. 대표적인 영역분할기법은 직교영역분할기법이다. 직교좌표계에서 계산영역을 분리하고 계산에 참여하는 코어들을 할당하는 방식이다. 코어간 병목현상을 최소화하기 위해 2차원에서는 정사각형, 3차원에서는 정육면체를 사용하는데, 해석하는 문제에 따라 높은 난이도의 알고리즘 개발 및 적용이 요구된다.

입자들이 운동방정식에 의해 독립적으로 움직이는 DEM과 다르게 SPH의 입자 거동은 주위 입자들과의 거리에 영향을 받는다. 그래서 SPH에 영역분할을 적용하려면 거리를 연속시키기 위한 매우 까다로운 조건들을 고려해야 한다. 한 분할된 계산 영역에서 주위 영역들을 탐색해야 하고, 영역의 경계면에서 smoothing 길이 이내의 입자들을 찾아야 한다. 그리고 그 정보들을 서로의 영역들이 공유해야 한다. 직교좌표계 기반의 영역분할에서는 이러한 고려사항들을 비교적 쉽게 적용할 수 있는 장점이 있다. 대표적인 연구가 Oger et al.(2016)가 개발한 ORB(Orthogonal Recursive Bisection) 알고리즘인데, 유동해석에서 병렬효율성을 높인 결과를 보여주고 있다. Yang et al.(2020)은 ORB를 사용하여 지반에서의 입자유동해석을 SPH로 구현하였다. 계산 영역의 경계면을 기준으로 smoothing 길이 안쪽과 바깥쪽의 입자들의 정보를 CPU간 공유를 하고 SPH 입자의 움직임을 계산하는 방법을 제안하였다. 그러나 이는 직교좌표계 영역분할의 장점을 잘 활용한 방법이지만 하나, 계산하고자 하는 영역의 형상이 복잡하거나 움직임 등이 있을 때는 직교좌표계 영역분할이 오히려 해석의 난이도를 높이는 결과를 초래한다. 본 논문에서는 임의 형상으로 계산영역을 분할할 수 있는 방법을 소개하고자 한다. 계산영역의 형태는 규칙적이지 않고 모델 형상에 따라 분할이 된다. 각 계산영역은 3차원 상에서 사면체 또는 육면체가 될 수 있고, 크기 또한 임의로 조절이 가능하다. 제안하는 방법의 장점은 직교좌표계로 분할하기 힘들었던 영역분할 문제를 해결할 수 있고, 계산영역을 실시간으로 변경시킬 수 있다.

2. SPH의 지배방정식

SPH는 유동에 관한 수치해석으로 운동방정식 중에 하나인 Navier-Stokes 방정식을 이산화 하여 구현한다(Randles & Liberskyb, 1996). 만약에 SPH 입자 a 가 가속도를 갖는

다고 하면 다음과 같이 이산화 된 방정식으로 나타낼 수 있다.

$$\frac{D\mathbf{v}(\mathbf{r}_a)}{Dt} = - \sum_{b=1}^N m_b \left[\frac{p(\mathbf{r}_a)}{\rho_a^2} + \frac{p(\mathbf{r}_b)}{\rho_b^2} \right] \nabla W_{ab} + \sum_{b=1}^N m_b \left[\frac{\mu(\mathbf{r}_a) + \mu(\mathbf{r}_b)}{\rho_a \rho_b} \mathbf{v}(\mathbf{r}_{ab}) \right] \left(\frac{1}{|\mathbf{r}_{ab}|} \frac{\partial W_{ab}}{\partial \mathbf{r}_{ab}} \right) \quad (1)$$

여기서, $\mathbf{v}(\mathbf{r}_a)$ 는 입자 a 위치 \mathbf{r}_a 에서의 속도 벡터, D/Dt 는 시간에 따른 변화율(시간도함수)을 나타낸다. m_b 는 입자 b 의 질량, N 은 a 입자를 제외한 주위 입자의 총 수, ρ 는 입자의 밀도, $p(\mathbf{r})$ 는 유체압력, W_{ab} 는 커널함수로 입자 a, b 입자 사이의 거리 \mathbf{r}_{ab} 에 따른 함수, $\mu(\mathbf{r})$ 는 점착응력을 나타낸다.

Eq. (1)에서의 밀도 ρ 는 주위 입자들의 질량들과 커널함수의 조합으로 정의될 수 있다.

$$\rho_a = \sum_{b=1}^N m_b W_{ab} \quad (2)$$

여기서, 커널함수 W_{ab} 는 다음과 같다(Morris, 1996).

$$W_{ab} = \begin{cases} (3-R)^5 - 6(2-R)^5 + 15(1-R)^5 & 0 \leq R < 1 \\ (3-R)^5 - 6(2-R)^5 & 1 \leq R < 2 \\ (3-R)^5 & 2 \leq R \leq 3 \end{cases} \quad (3)$$

여기서, 상수 k 는 1차원일 경우 $120/h$, 2차원일 경우 $7/478\pi h^2$, 3차원일 경우 $3/359\pi h^3$ 를 쓸 수 있다. 그리고 $R = r/h = |\mathbf{r} - \mathbf{r}'|/h$ 로 정의될 수 있다.

3. 다면체 영역분할 알고리즘

3.1 분할영역(decomposition)에서의 분할면(face) 정의

3차원 공간상에서 임의 형상으로 분할된 영역은 다수의 분할면(face)들로 이뤄져 있다. 분할면들은 꼭지점들로 정의되고 주위 영역과 공유를 한다. 하나의 분할면에는 한 쌍의 분할영역이 존재하는데, 이를 찾기 위해 각 분할면에는 고유값이 지정되고 모든 분할영역들이 분할면 검색을 통하여 쌍을 찾는다. 그리고 각 분할영역에서 분할면들이 지정되면 내부방향으로의 법선 벡터들이 정의된다(Fig. 1).

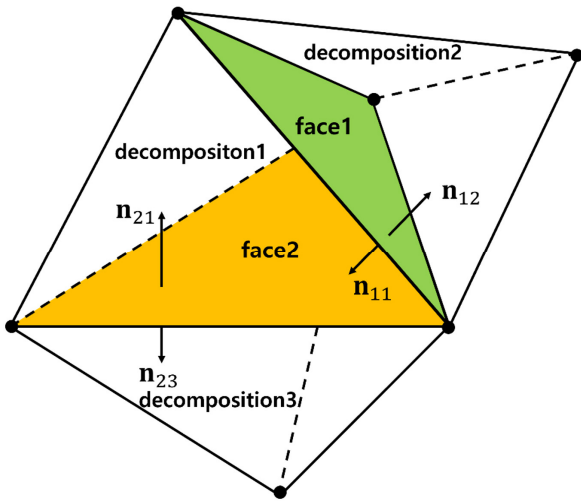


Fig. 1. Tripartite domain and bipartite plane (is the normal vector of partition face 1 in partition face 2)

3.2 분할영역 내부의 SPH 입자

직교좌표 기반의 영역분할에서는 SPH 입자들이 어느 영역에 속하는지 입자의 직교 좌표값으로 쉽게 찾을 수 있다. 그러나 제안되는 방법에서는 영역의 분할 형상이 다면체이므로 좌표값이 아닌 부피의 개념을 도입하여 입자의 위치를 판단한다. 예를 들어 Fig. 2와 같이 하나의 입자가 어느 분할 영역에 속해 있다면 그 분할영역의 부피는 각 분할면에서 입자까지의 다면체들의 부피합과 같다는 개념이다. 이 접근 방법은 다양한 형상의 다면체들을 분할영역에 쉽게 적용할 수 있는 장점을 갖는다. 그러나, 계산에 참여하는 모든 분할 영역의 분할면과 SPH 입자들 간의 조건 검색이 필요하므로 방대한 계산이 요구된다. 이를 해결하기 위해 참여하는 모든 코어들에 균등하게 SPH 입자들을 할당하고 독립적인 검색 과정을 진행한다. 그리고 각 코어에서 검색된 결과들은 MPI 라이브러리의 MPI_Allgatherv 메소드를 통하여 공유

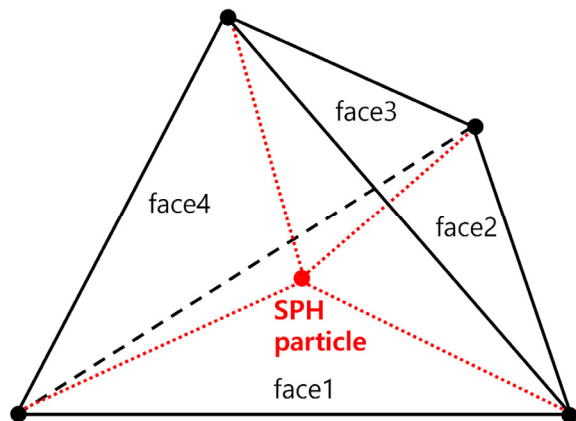


Fig. 2. If an SPH particle is located inside a tetrahedron, it can be divided into four smaller tetrahedron

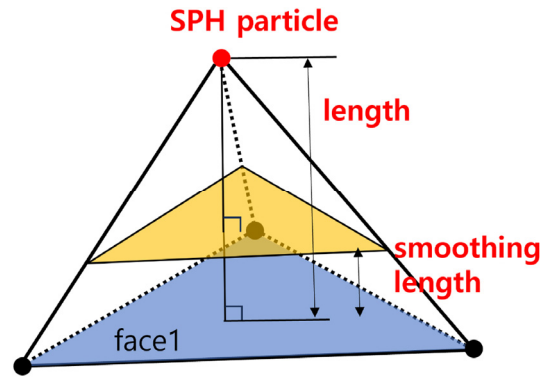


Fig. 3. A SPH particle located within the partitioned region calculates the distance from each boundary surface and determines whether it is within the smoothing length

된다. 한편, 경계면으로부터 smoothing 길이 이내에 위치해 있는 입자들은 따로 관리되어 저장 된다(Fig. 3). 이는 경계면을 공유하고 있는 반대쪽 분할영역에게 입자 정보를 넘겨주기 위한 것으로 경계면 근처에 위치해 있는 입자 밀도를 계산함에 있어 매우 중요한 역할을 한다. 이때 SPH 입자가 2개 이상 경계면에서 smoothing 길이 이내에 위치해 있다면 상대(opposite) 분할영역이 다르므로 문제가 되지 않는다.

3.3 Cross-point 주위의 입자

영역을 분할할 때 한 꼭지점을 공유하는 분할영역은 무수히 많다. 직교좌표 영역분할의 경우 최대 8개이지만 다면체의 경우 제한이 없다. 영역분할기법에서 이 꼭지점을 cross-point 라고 부르는데 공유하는 분할영역을 최소화해야 병렬효율성을 높일 수 있다. SPH 입자가 cross-point 근처에 있을 때는 3.2에서 설명되었던 방법으로 찾기가 힘들다. 경계면에서 연직방향으로만 검색이 되기 때문에 경계면들이 이어지는 구간에서는 음영구간이 발생할 수 있다. 이를 해결하기 위해 cross-point를 기준으로 smoothing 길이 안의 입자들을 따로 저장하고, 공유하는 다면체영역에 정보를 전달하였다. 만약 설정된 smoothing 길이의 값이 클 경우 경계면의 smoothing 길이와 중첩이 될 수 있는데, cross-point의 정보를 우선으로 하여 코어에 공유하도록 하였다.

4. SPH 수치해석

4.1 모델링

유동체 모사를 위하여 반지름이 0.5m, 높이가 1m인 경계

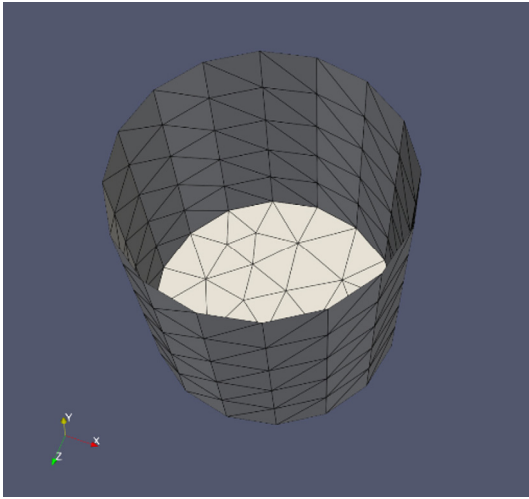


Fig. 4. Boundary face

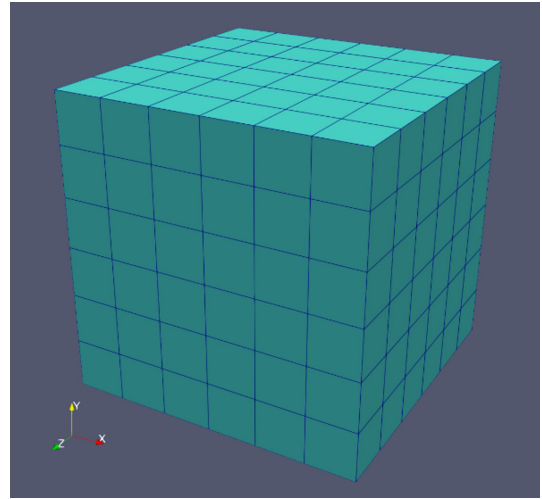


Fig. 6. Cube domain decomposition model

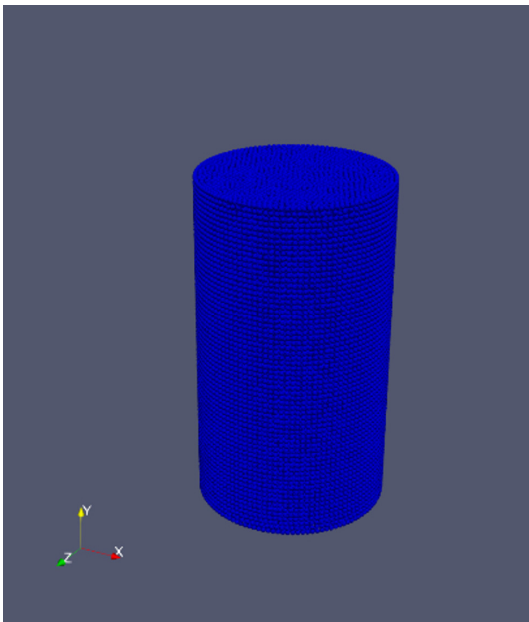


Fig. 5. SPH particles

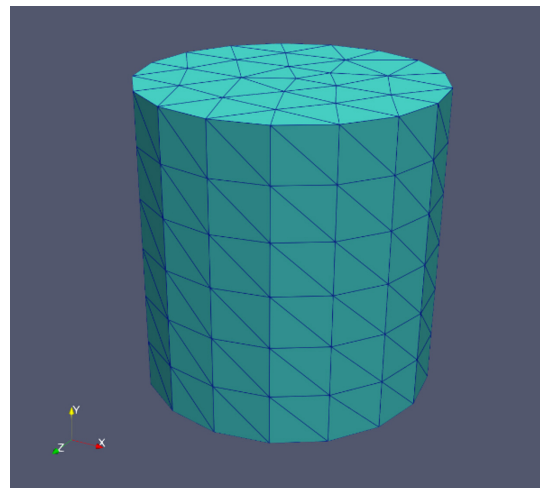


Fig. 7. Polyhedron domain decomposition model 1

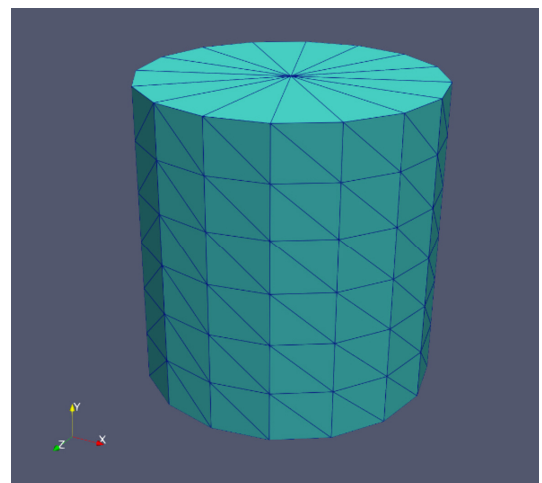


Fig. 8. Polyhedron domain decomposition model 2

영역은 총 204개의 삼각형 요소로 모델링 되었다(Fig. 4). SPH 입자는 경계영역 충돌 시 탄성충돌로 가정되었다. Fig. 5와 같이 유동을 위한 SPH 입자는 총 8,490개로 원통형태로 나열되었으며, 입자간 간격은 0.01m, 총 높이는 0.7m, 단위중량은 $1,000\text{kg/m}^3$ 로 적용하였다. 음속은 350m/s, 동점성계수는 $0.001\text{Pa}\cdot\text{s}$, Smoothing 길이는 0.03m, 동적해석을 위한 시간간격은 0.00035초가 사용되었다.

Fig. 6~8과 같이 영역분할은 총 3가지 경우에 대해서 모델링 되었다. 육면체 영역분할은 216개, 사면체 영역분할 모델 1은 1,063개, 사면체 영역분할 모델 2는 470개의 분리된 다면체 계산영역들이 사용되었다. 사면체 영역분할 모델 2는 모델 1과 다르게 원통의 중심에 사면체 꼭지점들을 집중시켰다. 이를 통해 극대화된 cross-point 모델에서 제안된 점

근방법의 병렬효율성을 검토하고자 하였다. Fig. 9는 경계영역, SPH 입자, 영역분할 모델을 동시에 보여주고 있다. 영역분할의 크기는 경계영역모델 보다 크게 설정되었는데, 이는

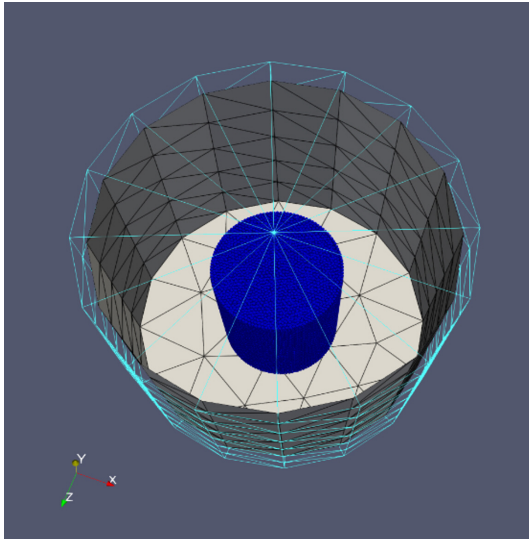


Fig. 9. Total model

SPH 입자가 경계면 충돌 시 발생할 수 있는 경계면 점프 오차를 피하기 위해서이다.

4.2 해석과정

제안된 모델에서 SPH 입자들 간의 거동해석을 위하여 총 2CPU의 28코어(3.1GHz, 56스레드)와 256GB 메모리가 탑재된 워크스테이션이 사용되었다. 소프트웨어는 C++ 기반으로 작성되었고, 병렬프로세싱을 위하여 Intel사의 OneAPI 모듈이 사용되었다.

제안된 알고리즘의 코드는 다음과 같이 총 6단계로 구성된다.

- Step 1(분할영역, 분할면 정의): 다면체 영역분할에 대한 분할면을 찾고 공유하고 있는 계산 영역을 찾는 단계이다. 그리고 분할영역에 내에 위치해 있는 입자와 경계면 및 cross-point 근처에 위치해 있는 입자들을 찾는다.
- Step 2(밀도계산): Eq. (2)를 통하여 밀도를 계산한다. smoothing 길이 안에서의 입자 정보를 이용해야 하기 때문에 1)다면체 영역내부 입자, 2) 경계면 근처 입자, 3) Cross-point 근처 입자에 대한 정보를 활용하여 계산한다.
- Step 3(밀도 및 압력 공유): 입자에 대한 정보(밀도 및 압력)를 MPI 라이브러리를 통하여 모든 코어들에 공유한다.
- Step 4(가속도 계산): Eq. (1)의 가속도를 계산한다. Smoothing 길이 안에서의 입자 정보를 활용하여 가속도가 계산되고 leap-frog 방법을 통하여 속도와 위치가 계산된다.

- Step 5(가속도 공유): 3단계와 마찬가지로 입자에 대한 정보(가속도, 속도, 위치)를 MPI 라이브러리를 통하여 모든 코어들에 공유한다.
- Step 6(파일출력): 계산이 끝난 입자 정보들을 파일로 출력한다.

4.3 해석결과

Fig. 10은 단일 시간 간격에서 코어수에 따른 Speedup을 나타내는 그래프이다. 여기서 Speedup은 병렬프로세싱시간/순차프로세싱시간으로 병렬의 효율성을 판단할 수 있는 지표이다. 코어의 수에 따라 Speedup도 선형적으로 증가(기울기가 1)하는 것이 이상적이지만, 프로그램에서의 순차화 과정과 하드웨어의 물리적 한계로 Speedup은 임의 순간부터 증가하지 않는다. 암달의 법칙(Amdal's Law)에 의하면 프로그램이 95% 병렬화가 되었다면 Speedup은 최대 20을 넘길 수 없다. Fig. 10에서 볼 수 있듯이 사면체분할 1, 2 모델의 결과 값들은 12코어까지 95%의 병렬화 Speedup 값들과 거의 차이가 없다. 반면 육면체분할은 병렬효율성이 사면체 분할보다 낮음을 알 수 있다. 수치해석에 사용된 모델들은 12코어에서 가장 병렬효율성이 좋게 나타났고, 16코어 이하부터 모든 모델에서 병렬효율성이 떨어지는 것을 볼 수 있다.

Fig. 11부터 13은 단일 시간 간격에서 수치모델에서의 코어-단계별 해석 시간을 나타낸 그래프이다. 밀도 계산(단계 2)과 가속도 계산(단계 4)에서 소요되는 시간이 크게 나타났다. 이는 smoothing 길이 이내의 입자들을 찾고 계산하는 과정에서 나타나는 당연한 결과이다. 그러나 20코어부터는 해당 단계에서 해석시간이 증가함을 볼 수 있는데, 이는 계산하는 분할영역이 작아지고 고려되는 주위 분할영역의 입자 수가 증가하였기 때문이다. 이 결과는 코어간 정보를 공유하는 단계(단계 3, 5)에서도 동일하게 나타나는데, 분할영역이 많아질수록 정보 공유량이 증가되어 해석시간이 증대

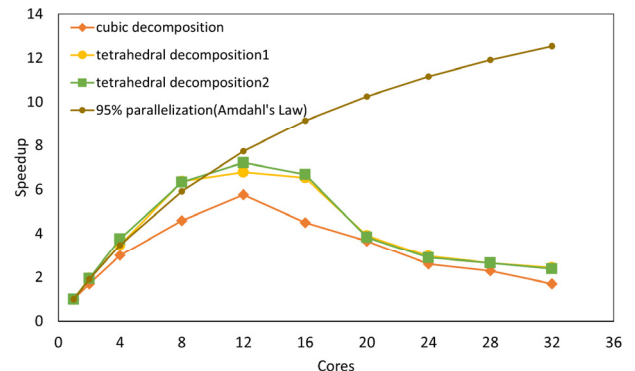


Fig. 10. Speedup by core usage

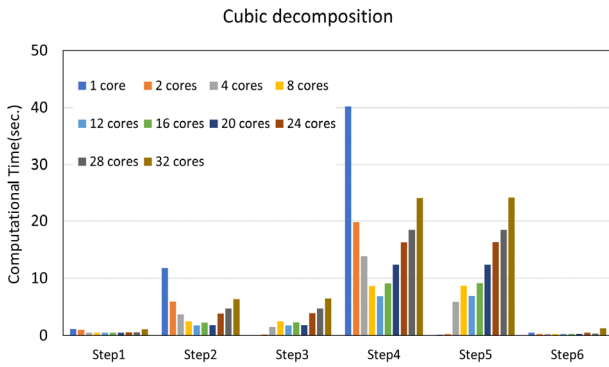


Fig. 11. Computational time in cube decomposition

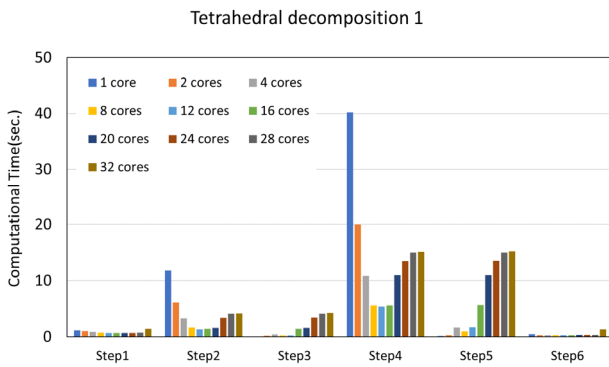


Fig. 12. Computational time in the tetrahedral decomposition model 1

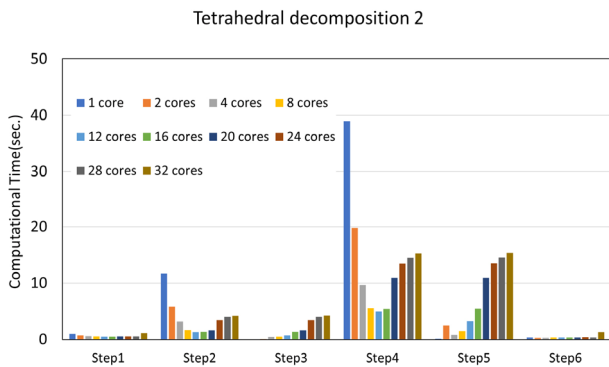


Fig. 13. Computational time in the tetrahedral decomposition model 2

됨을 알 수 있다.

Fig. 14부터 16까지는 단일 시간 간격에서 순차해석, 12 코어, 20코어를 사용하였을 경우 단계별 해석시간을 비교한 그래프이다. 본 그래프를 통하여 cross-point 증가에 따른 병렬효율성을 판단할 수 있다. Cross-point에 근접해 있는 SPH 입자들이 많고 공유하고 있는 분할영역이 많을수록 정보 공유 크기는 증가한다. 사면체분할 모델 2에서는 cross-point가 극대화되어 있는데, Fig. 15에서의 밀도, 압력 공유, 가속도 공유 단계를 보면 모델 2가 모델 1보다 증가함을 알 수 있다. 모델 2가 모델 1보다 사면체 영역분할 수가 작음에도 불구하고 정보 공유수가 높다는 것은 의미 있는 결과이다. 하지만, Fig. 16에서와 같이 코어수가 증가하면 사면체모델

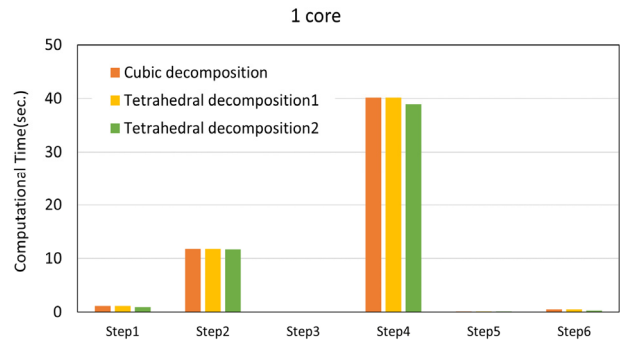


Fig. 14. Computational time in sequential analysis

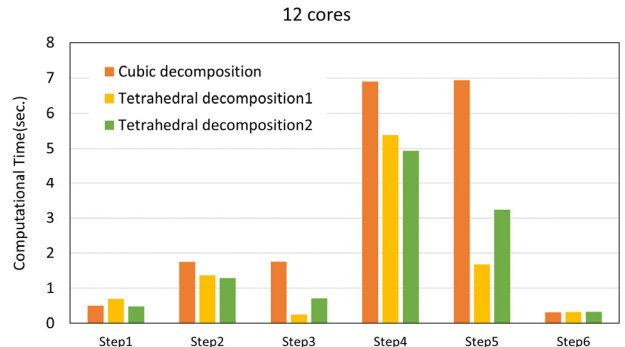


Fig. 15. Computational time on 12 cores

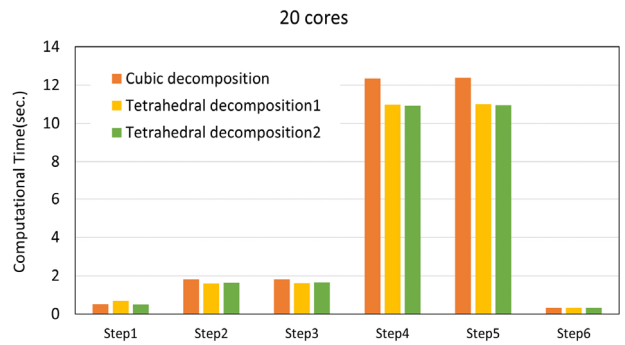


Fig. 16. Computational time on 20 cores

1과 모델 2의 차이는 거의 없다. 이는 앞에서 설명했듯이 코어수가 늘어남으로 인해 모델 1에서 계산하고자 하는 분할영역이 작아지고 정보공유 크기가 증가했기 때문으로 판단된다.

Fig. 17부터 20까지는 영역분할모델 2에 대해 시간에 따른 SPH 입자의 위치를 나타낸 그림이다. 총 12코어가 사용되었다.

5. 결 론

본 논문에서는 SPH의 다면체 영역분할 알고리즘을 제안하고 수치해석을 통하여 우수성을 검증하였다. 병렬해석 시

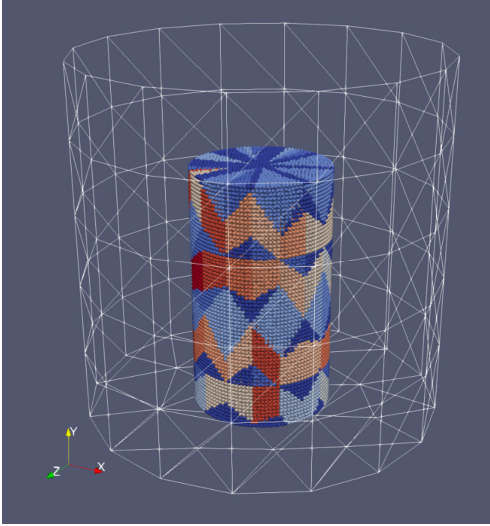


Fig. 17. SPH particles at 0 sec. (decomposition model 2, The use of 12 colors represents decompositions)

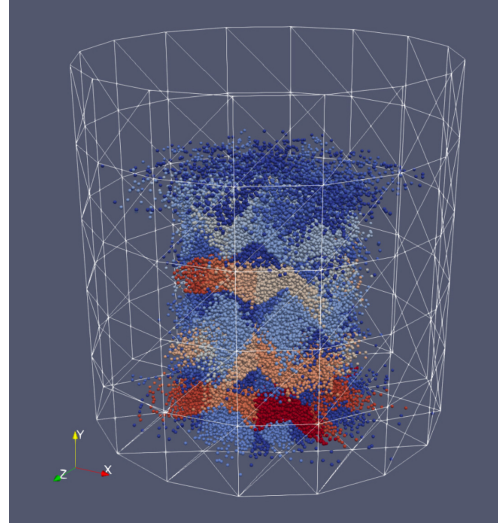


Fig. 20. SPH particles at 3.5 sec. (decomposition model 2, The use of 12 colors represents decompositions)

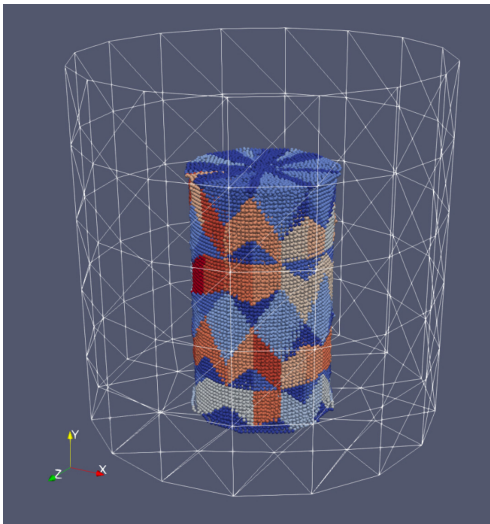


Fig. 18. SPH particles at 0,35 sec. (decomposition model 2, The use of 12 colors represents decompositions)

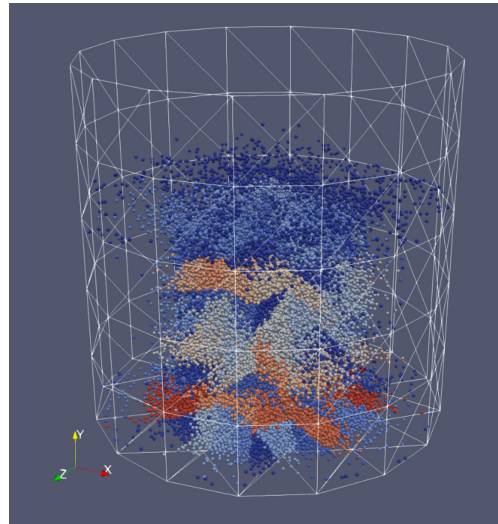


Fig. 21. SPH particles at 5,25 sec. (decomposition model 2, The use of 12 colors represents decompositions)

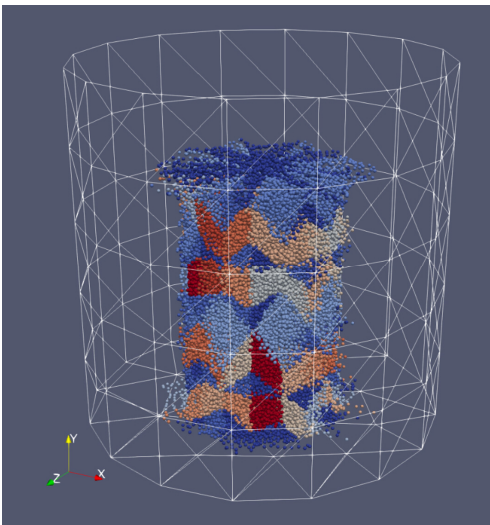


Fig. 19. SPH particles at 1,75 sec. (decomposition model 2, The use of 12 colors represents decompositions)

계산하고자 하는 계산영역의 형태가 복잡할 경우 직교좌표계 기반의 영역분할기법은 잉여영역이 발생하여 병렬효율성이 떨어질 수 있다. 다면체영역분할은 분할형태가 자유로워 어떤 형태의 SPH 모델에도 적용할 수 있는 장점이 있다. 다만, 수치해석 예제 결과에서 나타났듯이 수치모델 문제에 적합하게 다면체영역분할을 모델링해야 한다. 코어 증가와 cross-point에 공유하고 있는 분할영역 수 등이 고려되어야 한다. 수치해석 예제의 경우 제안된 알고리즘은 12코어까지 병렬화 95%의 결과를 보여줬지만 16코어 이후부터는 코어 간 정보공유 크기가 증가하여 병렬효율성이 떨어지는 것을 볼 수 있었다. 다면체영역분할은 직교영역분할보다 주변영역으로의 정보 전달량이 많지만 최대 병렬효율성을 발휘하는 임계 코어수 내에서는 우수한 결과를 보여주고 있다.

감사의 글

본 연구는 2019년 정부(교육부)의 재원으로 한국연구재단 이공분야 기초연구사업의 지원을 받아 수행된 연구임(NRF-2019R111A1A01059014).

References

1. Morris, J. P. (1996), Analysis of smoothed particle hydrodynamics with applications, Ph. D. thesis, Monash University, Australia.
2. Oger, G., Le Touzé, D., Guibert, D., De Leffe, M., Biddiscombe, J., Soumagne, J. and Piccinalli, J. G. (2016), On distributed memory MPI-based parallelization of SPH codes in massive HPC context. *Computer Physics Communications*, Vol. 200, pp. 1~14.
3. Randles, P.W. and Liberskyb, L.D. (1996), Smoothed Particle Hydrodynamics: Some recent improvements and applications, *Computer Methods in Applied Mechanics and Engineering*, Volume 139, No. 1~4, pp. 375~408.
4. Tak, M. H. and Park, T. H. (2013), High scalable non-overlapping domain decomposition method using a direct method for finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, Vol. 264, pp. 108~128.
5. Yang, E., Bui, H. H., De Sterck, H., Nguyen, G. D. and Bouazza, A. (2020), A scalable parallel computing SPH framework for predictions of geophysical granular flows, *Computers and Geotechnics*, Vol. 121, pp. 103474.