Journal of the Korean Geo-Environmental Society
25(4): 13～20. (April 2024) https://journal.kges.or.kr

ISSN 1598-0820 (Print)
ISSN 2714-1233 (Online)
DOI https://doi.org/10.14481/jkges.2024.25.4.13

Cartesian 좌표기반 동적영역분할을 고려한 SPH의 충돌 및 병렬해석

The Contact and Parallel Analysis of SPH Using Cartesian Coordinate Based Domain Decomposition Method

탁 문 호†
Moonho Tak

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

ABSTRACT : In this paper, a parallel analysis algorithm for Smoothed Particle Hydrodynamics (SPH), one of the numerical methods for fluidic materials, is introduced. SPH, which is a meshless method, can represent the behavior of a continuum using a particle-based approach, but it demands substantial computational resources. Therefore, parallel analysis algorithms are essential for SPH simulations. The domain decomposition algorithm, which divides the computational domain into partitions to be independently analyzed, is the most representative method among parallel analysis algorithms. In Discrete Element Method (DEM) and Molecular Dynamics (MD), the Cartesian coordinate-based domain decomposition method is popularly used because it offers advantages in quickly and conveniently accessing particle positions. However, in SPH, it is important to share particle information among partitioned domains because SPH particles are defined based on information from nearby particles within the smoothing length. Additionally, maintaining CPU load balance is crucial. In this study, a highly parallel efficient algorithm is proposed to dynamically minimize the size of orthogonal domain partitions to prevent excess CPU utilization. The efficiency of the proposed method was validated through numerical analysis models. The parallel efficiency of the proposed method is evaluated for up to 30 CPUs for fluidic models, achieving 90% parallel efficiency for up to 28 physical cores.

Keywords : SPH, Parallel analysis, Domain decomposition, Dynamic domain decomposition, Load balance, Contact and impact method

요 지 : 본 논문에서는 유동체를 해석할 수 있는 수치해석기법 중 하나인 SPH(Smoothed Particle Hydrodynamics)의 병렬해석 알고리즘이 소개된다. 무요소법(meshless method)의 SPH는 연속체 거동을 입자기반으로 표현하기 때문에 컴퓨팅하는데 높은 자원을 요구한다. 그래서 병렬해석 알고리즘은 SPH 시뮬레이션에서 필수적으로 고려되어야 한다. 계산영역을 일정한 간격으로 분할시켜 독립적으로 해석하는 영역분할 알고리즘은 병렬해석 알고리즘 중에 가장 대표적인 방법이다. 그리고 그 중 Cartesian 좌표계의 영역분할 방법은 입자들의 좌표를 빠르고 편리하게 검색할 수 있는 장점이 있어, DEM(Discrete Element Method)이나 MD(Molecular Dynamics)에서 대중적으로 사용되고 있다. 그러나 SPH의 경우 입자들이 smoothing 길이 이내의 주위 입자 정보가 필요하기 때문에 분할 영역 간의 입자정보 공유가 중요하다. 그리고 이에 따른 CPU의 로드밸런스가 중요하다. 본 연구에서는 직교 영역분할의 크기를 동적으로 미소화 시켜 잉여 CPU가 발생하지 않도록 하는 높은 병렬효율성의 알고리즘이 제안되었다. 그리고 수치해석 모델을 통하여 효율성을 검증하였다. 유동체 모델에 대해 총 30 CPU까지 제안된 방법의 병렬효율성을 검토하였고, 28개의 물리적 코어 수까지 90%의 병렬효율성을 얻을 수 있었다.

주요어 : SPH, 병렬해석, 영역분할, 동적영역분할, 로드밸런스, 충돌해석

1. 서 론

유동성 있는 지반이나 유체의 흐름을 수치적으로 해석하기 위해서는 격자 또는 입자기반의 접근방법이 필요하다. 격자기반의 수치해석으로 CFD(Computational Fluid Dynamics)가 가장 대표적인데, 고정된 격자에서 절점의 상태 값을 계산하는 방식으로 Eulerian 방법이라고도 한다. Navier-Stokes 방정식 또는 연속방정식을 모델링하기 쉽고 계산량이 많지 않아 대중적으로 사용되는 방법이다. 그러나 자유경계면 또는 동적인 구속조건이 고려되는 상황이라면 격자가 고정되어 있는 CFD 방법이 적절하지 않다. 이러한 경우 격자의 이동을 허용하는 Lagrangian 방법의 유한요소해석(Finite Element Method)이 적합하다. 그러나 격자의 대변형을 처리하는 과정에서 많은 해석시간이 요구되고(Zienkiewicz et al., 2005) 격자의 왜곡현상이 발생할 경우 수렴값을 찾기 힘든 단점을 갖고 있다.

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

무요소 방법은 격자기반의 단점을 보완하기 위해 만들어진 입자기반의 수치해석 방법이다. 유동 해석에서는 SPH 방법이 대표적이다. 각 유동성 입자들은 질량을 갖고 있고 밀도는 주위 입자들로부터 결정된다. 그리고 Navier-Stokes 방정식을 커널함수로 이산화하고 가속도를 계산하는 방식을 채택하고 있는데, 입자의 수가 많을수록 결과의 정확도가 높아지는 특징을 갖는다. SPH는 1977년 천체물리분야에서 처음 소개되었고(Lucy, 1977; Gingold & Monaghan, 1977), 유체와 고체 분야까지 연구 범위가 확장돼 연구되어 왔다(Benz & Asphaug, 1995; Johnson & Beissel, 1998; Monaghan, 1994). 입자의 거동을 자유롭게 모사할 수 있는 장점을 갖고 있어 FSI(Fluid-Structure Interaction) 해석에서도 사용되고 있다(Sun et al., 2020; Mahdavi, 2018; Kwon & Cho, 2010).

SPH에서 물체의 유동을 정확하게 모사하기 위해서는 충분한 입자의 수가 필요하고 입자들 간의 간격을 유지시켜 밀도의 변화를 최소화시켜야 한다. 이는 해석시간과 연관이 있는데, 하나의 입자가 주위 입자들을 찾아내고 밀도와 속도를 계산하는 과정에서 해석시간은 계산영역의 크기와 관련이 깊다. 즉, 계산하고자 하는 영역을 줄이면 해석의 효율성이 높아진다. 영역분할기법은 계산영역을 줄이고 독립적으로 해석할 수 있는 방법으로 입자기반 해석에서 필수적으로 적용되는 병렬해석기법 중 하나이다. SPH, DEM, MD와 같은 입자기반의 수치해석에서는 영역분할의 행태를 Cartesian 좌표기반의 정육면체를 사용하는데, 입자들의 위치를 찾기 쉽다는 장점을 갖고 있다. 그러나 시간에 따라 입자들의 움직임이 발생하여 국부적으로 위치해 있을 경우 크기가 고정되어 있는 영역분할 기법은 계산의 비효율성을 증대시킬 수 있다. 본 논문에서는 SPH의 병렬해석을 위해 Cartesian 좌표계 기반의 동적 미소영역분할 알고리즘이 소개된다. 그리고 유동모델 해석을 통해 코어수에 따른 로드밸런스가 계산되고, 제안된 알고리즘의 병렬효율성을 검증한다.

2. Smoothed Particle Hydrodynamics (SPH)

2.1 지배방정식

SPH를 통하여 유동해석을 진행하기 위해서는 Navier-Stokes 방정식을 기반으로 한 지배방정식을 커널함수를 통해 이산화해야 한다. SPH는 절점의 위치 이동을 허용하는 Lagrange 방식을 사용하는데, Lagrange에서의 연속체 거동은 Eq. (1)과 같이 운동량방정식으로 나타낼 수 있다(Randles & Liberskyb, 1996).

$$\frac{D\boldsymbol{v}(\boldsymbol{r}_a)}{Dt}=\frac{1}{\rho_a}\nabla(-p(\boldsymbol{r}_a)+\mu(\boldsymbol{r}_a)) \tag{1}$$

여기서, D/Dt는 운동량 변화율, $\boldsymbol{r}_a$는 유동 입자 a에 대한 위치 벡터, $\boldsymbol{v}(\boldsymbol{r}_a)$는 입자의 속도 벡터를 나타낸다. 그리고 ρ_a는 a 입자의 밀도, ∇는 gradient, $p(\boldsymbol{r}_a)$는 등방압력, $\mu(\boldsymbol{r}_a)$는 점착응력을 나타낸다.

2.2 지배방정식의 이산화

커널함수의 근사화는 Eq. (2)의 Dirac sifting property로부터 시작된다.

$$f(x)=\int f(x')\delta(x-x')dx' \tag{2}$$

여기서, $f(x)$는 위치 x에 대한 함수로 δ는 디랙델타함수(Dirac delta function)를 나타내고, 함수 $f(x)$는 주위 다른 입자들 위치에 대한 적분형태로 정의가 된다. 디랙델타함수는 smoothed 커널함수로 대체 가능하며, 함수 $f(x)$는 Eq. (3)과 같이 근사화된 식으로 나타낼 수 있다.

$$f(x)\cong\int f(x')\,W(x-x',h)dx' \tag{3}$$

여기서, $W(x-x',h)$는 smoothed 커널함수를 나타내며 디랙델타함수와 마찬가지로 적분한 값은 1을 갖는다. 그리고 h는 커널함수의 영향을 미치는 거리로 smoothing 길이라고도 한다. Smoothed 커널함수는 SPH에서 사용되기 위해 다음과 같이 3가지 경우가 고려된다(Morris, 1996).

$$W(R,h)=k\begin{cases}(3-R)^5-6(2-R)^5+15(1-R)^5 & 0\le R<1\\ (3-R)^5-6(2-R)^5 & 1\le R<2\\ (3-R)^5 & 2\le R\le 3\end{cases} \tag{4}$$

여기서, 상수 k는 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 있고, 두 유동입자 사이간의 거리벡터를 나타낸다.

한편, Eq. (3)에서의 적분은 다음과 같이 이산화하여 표현할 수 있다.

$$f(x)=\sum_{i=1}^{N}\frac{m_i}{\rho_i}f(x_i)\,W(x-x_i,h) \tag{5}$$

여기서, N은 입자 i가 포함되어 있는 계산영역 내에서의 총 입자수를 나타내고, m_i는 i입자의 질량을 나타낸다. 이산화된 Eq. (5)는 밀도와 운동량방정식에 적용이 가능하다. 입자 a에 대한 밀도와 운동량방정식 Eq. (1)을 Eq. (5)에 대입하여 정리하면 다음과 같이 나타낼 수 있다.

$$\rho_a = \sum_{b=1}^{N} m_b W_{ab} \tag{6}$$

$$\frac{D\boldsymbol{v}(\boldsymbol{r}_a)}{Dt} = -\sum_{b=1}^{N} m_b \left[\frac{p(\boldsymbol{r}_a)}{\rho_a^2} + \frac{p(\boldsymbol{r}_b)}{\rho_b^2} \right] \nabla W_{ab} + \sum_{b=1}^{N} m_b \left[\frac{\mu(\boldsymbol{r}_a) + \mu(\boldsymbol{r}_b)}{\rho_a \rho_b} \boldsymbol{v}(\boldsymbol{r}_{ab}) \right] \left(\frac{1}{|\boldsymbol{r}_{ab}|} \frac{\partial W_{ab}}{\partial r_{ab}} \right) \tag{7}$$

여기서, 커널함수 $W_{ab} = W(\boldsymbol{r}_a - \boldsymbol{r}_b, h)$는 a 입자와 b 입자 간 거리 $|\boldsymbol{r}_{ab}|$에 대한 함수로 거리에 따라 Eq. (4)와 같다. 그리고 압력 $p(\boldsymbol{r})$는 유동압력으로 a 입자에 작용하는 비압축성 유동압력 $p(\boldsymbol{r}_a)$은 상태방정식 Eq. (8)과 같이 나타낼 수 있다.

$$p(\boldsymbol{r}_a) = c^2 \rho_a \tag{8}$$

여기서, c는 음속을 나타낸다.

3. SPH의 영역분할 알고리즘

SPH 방법을 통한 신뢰도 높은 유동 해석을 진행하기 위해서는 smoothing 길이 안에 많은 수의 입자들이 분포되어야 한다. 각 입자들은 smoothing 길이 안의 다른 입자들에 의해 밀도가 계산되고, Eq. (7)을 통해 속도가 계산된다. 영역분할기법은 수많은 입자들의 계산을 효율적으로 도와줄 수 있는 장점을 갖는다. 입자들의 움직임을 고려하여 계산영역이 정해지고 작은 단위로 분할하여 병렬해석이 가능하게 된다. 일반적으로 계산영역은 균등하게 고정한 정적영역분할기법을 사용하는데, Cartesian 좌표계에서 입자들의 위치를 쉽게 찾을 수 있는 장점을 갖는다. 그러나 경계영역이 움직이거나 유동 입자들의 속도 변화가 클 경우 입자들이 비균질하게 분포되어 로드밸런스가 낮아질 수 있다. 본 논문에서 제안되는 동적영역분할기법은 분할된 영역에서 입자들의 분포가 균등하게 유지되도록 하고, 코어간 병렬해석에서 로드밸런스를 높여줄 수 있다.

3.1 SPH에서의 영역분할 및 MPI 정보전달

DEM, MD와 같은 독립적인 입자의 움직임을 해석하는 기법들과 다르게 SPH에서는 주위 입자들 위치에 영향을 받는다. 그리고 분할영역 간에 정보를 전달하는 과정에서 smoothing 길이 만큼의 주위 영역에 대한 추가 정보가 필요하다(Fig. 1).

각 계산영역의 입자들 거동은 물리적으로 분리된 코어에서 계산된다. 각 코어는 독립된 메모리로 연결되어 있으며, 이들 연결구조를 노드(node)라 불린다. 노드간 통신은 MPI (Message Passing Interface) 규약을 통해 데이터가 전달된다 (Fig. 2). MPI는 다양한 병렬 알고리즘를 구성할 수 있는 함수들을 제공한다. Fig. 1과 같이 2개로 분할된 계산영역이 고려될 시 계산영역 1은 smoothing 길이만큼의 입자정보를 계산영역 2에 보내야 하고, 계산영역 2 또한 동일한 길이에 대해 입자정보를 계산영역 1에 보내야 한다. 한 계산영역에

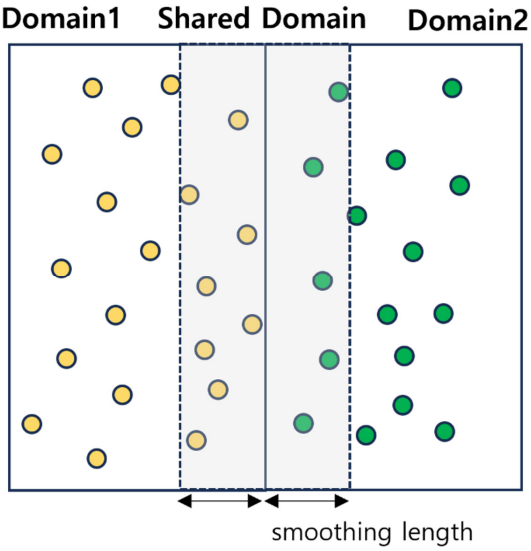

Fig. 1. Computational domains of the SPH method

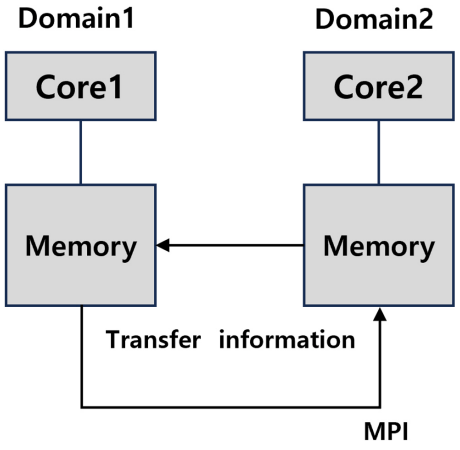

Fig. 2. Message passing interface between two cores on the SPH method

인접되어 있는 영역이 많을수록 MPI를 이용한 정보전달의 양이 많아지는데, 해석 속도를 저하시키는 원인이 될 수 있다.

3.2 동적영역분할 알고리즘

본 논문에서는 계산영역이 시간해석 단계마다 변경되는 동적영역분할 알고리즘이 제안된다. 독립적인 노드가 담당하는 각 계산영역은 시간에 따라서 그 영역이 변경되며, 계산영역들은 동일한 수들의 SPH 입자들을 포함한다. 이는 병렬해석의 로드밸런스에서 매우 중요한 역할을 하며 전체 해석시간에도 큰 영향을 끼친다. 제안되는 동적영역분할 알고리즘은 다음과 같이 총 10개 단계로 구분된다.

Step 1. SPH particle 입력: SPH 입자에 대한 입자 정보를 불러들이는 과정으로 계산을 담당하는 모든 노드에서 진행된다. 유동에 필요한 입자들의 위치, 시간간격, smoothing 길이 등이 입력된다.
Step 2. 전체 계산영역 입력: SPH 입자들 거동이 계산되는 전체 계산영역을 정의한다. 여기서 전체 계산영역은 미소화된 분할영역들을 포함시키는 그룹화된 영역이다. 계산을 담당하는 모든 노드에서 진행된다.
Step 3. 전체 계산영역의 경계조건: SPH 입자들은 전체 계산영역 범위를 넘어설 수 없으며, 이를 위한 경계조건을 적용시킨다. 경계면에 충돌한 SPH 입자는 벽에 반사됨을 가정한다. 임의 경계면에 대한 반사각을 계산하기 위해서 각 경계면은 법선 벡터가 계산된다. 이는 모든 노드에서 계산이 진행된다.
Step 4. 전체 계산영역의 분할: 전체 영역은 임의 수의 그룹화된 미소 영역으로 분할된다. 미소 영역들은 SPH 입자의 포함여부에 따라 계산영역 또는 잉여영역으로 구분된다. 분할과정에서 생성된 미소 영역들은 인접해 있는 계산영역들을 찾고, 계산영역 간의 경계면에서 normal 벡터를 계산하게 된다. 그리고 미소영역들은 SPH 입자가 포함되어 있는지 계산되며, 입자가 불포함된 영역은 잉여로 남게 된다. 이 단계는 처음 생성된 SPH 입자들을 미소영역들이 감싸는 (Wrapping) 단계로 그룹집단을 생성한다. 입자의 간격이 균질하고 밀도가 일정하면 그룹화된 미소계산영역에서의 입자수들은 거의 동일하다. 본 단계에서는 모든 노드가 계산에 참여하게 된다.
Step 5. 시간해석 진행
Step 5-1. SPH 입자 이동에 의한 계산영역의 재분할: 이동된 입자들이 어느 미소 계산영역과 그룹에 포함되는지 확인하여 업데이트되는 과정이다. 그리고 계산영역의 각 경계면에서 smoothing 길이 이내의 입자들을 판별하고 저장한다. 입자들에 대한 정보는 인접한 영역으로 MPI를 통하여 전달된다.
Step 5-2. 가용 노드 할당: 미소 계산영역들의 그룹이 확정되면 가용할 수 있는 노드들을 판단하여 그룹에 할당한다. 계산에 참여하는 노드수는 미소 계산영역들의 그룹 수보다 클 수 없으며, 각 노드들은 1개 이상의 미소 계산영역 그룹에서 SPH 입자 거동 계산에 참여하게 된다.
Step 5-3. 초기화: 각 노드에 할당된 계산영역에서 모든 변수들을 초기화시킨다.
Step 5-4. 계산영역 내에서의 smoothed 함수 및 거리 계산: 계산영역 내에 있는 SPH 입자들은 주위 SPH 입자들과의 거리를 계산한다. 그리고 smoothing 길이 안에서 smoothed 함수, 입자의 밀도를 계산한다. 계산영역의 경계면에 인접한 입자들은 Step 5-1로부터 받은 정보를 활용한다.
Step 5-5. 압력계산: 각 노드에서는 입자의 밀도를 통하여 압력을 계산한다.
Step 5-6. 노드간 정보공유: 각 노드에서 계산된 밀도, 압력들은 MPI를 통하여 모든 노드에 공유된다.
Step 5-7. 가속도 계산: 공유된 밀도와 압력을 사용하여 각 노드에서는 Eq. (7)에서 정의된 가속도를 계산한다.
Step 5-8. 속도 및 위치 계산: Step 5-7로부터 계산된 가속도는 leap-frog 방법을 통하여 속도와 가속도가 계산된다.
Step 5-9. 입자의 경계면 충돌 처리: 입자가 경계면에 충돌하여 반사될 경우 leap-frog에서 전후 속도가 계산된다.
Step 5-10. 입자 위치, 속도 공유를 위한 버퍼 생성: 모든 노드에서 입자상태를 저장하기 위한 버퍼 저장공간을 생성하고 초기화시킨다.
Step 5-11. 입자 위치, 속도 공유: MPI를 통하여 모든 노드들이 계산될 입자의 위치, 속도를 공유한다.
Step 5-12. 공유된 입자의 처리: 사용된 버퍼를 삭제하고 공유된 데이터를 각 계산영역의 입자에 업데이트시킨다.
Step 5-13. 결과출력: 각 입자들의 위치 정보는 Post-Processing을 위하여 파일로 출력된다.

4. SPH 수치해석

4.1 모델링

가로, 세로, 높이가 모두 1m인 수조에 0.25m 높이의 토석류와 같은 유동체가 있다고 가정한다. 그리고 이 유동체는 y 방향으로 0.05m 떨어진 위치에서 중력에 의해 바닥에 닿는다. 경계박스는 삼각형 요소로 모델링 되었으며, SPH 입자들의 경계면 역할을 한다(Fig. 3). 유동체의 경우 7,082개의

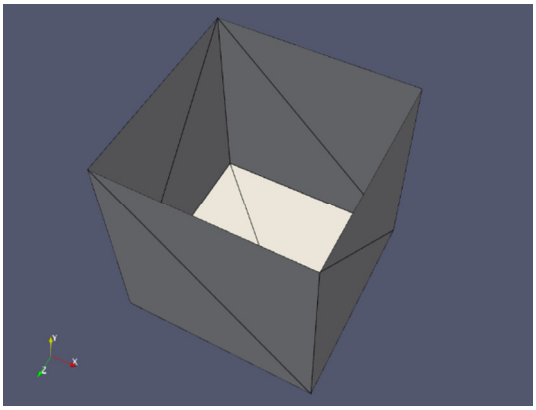

Fig. 3. Boundary box

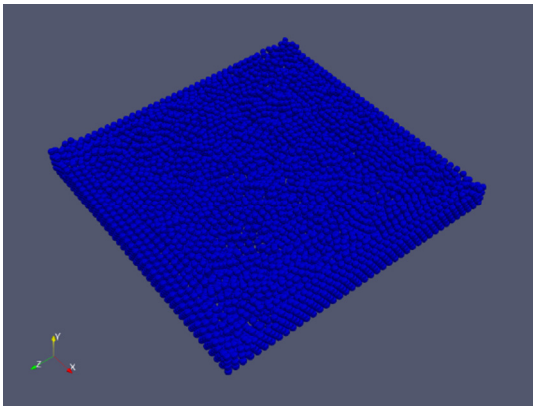

Fig. 4. SPH particles

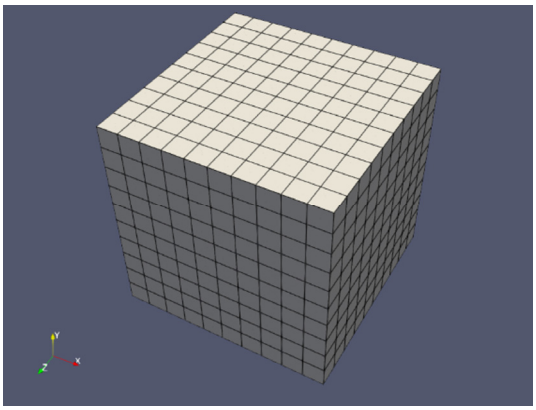

Fig. 5. Decomposed domains

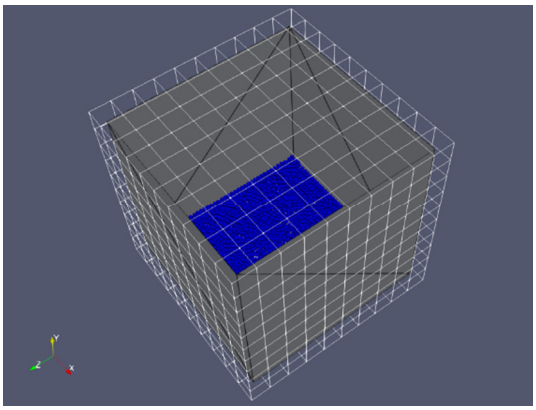

Fig. 6. Assemble model

SPH 입자들로 모델링 되었다(Fig. 4). 유동체에 대한 단위중량은 1,000kg/m^3, Eq. (8)에서의 음속 c는 350m/s, 동점성 계수는 0.001Pas가 사용되었다. 밀도 계산을 위하여 smoothing 길이는 0.01m로 고정하였다. 병렬해석을 위한 영역은 총 1,000개로 분할하였으며, 분할영역들의 크기는 모두 같다(Fig. 5). 이 분할영역들은 SPH 입자의 위치와 참여하는 노드 수에 따라 미소계산영역과 잉여영역으로 분리되고 그룹화된다. 그리고 해석이 진행되는 동안 SPH 입자의 이동이 고려되어 계산영역이 변경된다. 경계박스면과 계산영역이 동일하면 경계면에서 이탈된 입자의 처리가 어려워 분할된 영역의 크기는 경계박스 보다 크게 모델링 되었다(Fig. 6).

4.2 해석과정

유동체 해석을 위한 제안된 방법은 Intel C++ 컴파일러를 통하여 프로그래밍되었다. 병렬해석 진행을 위하여 Intel OneAPI에서 제공하는 MPI 모듈이 사용되었다. 해석을 위한 시스템은 총 28 코어(3.1GHz, 56쓰레드), 256GB 메모리의 SMP(Shared Memory Processor) 장비가 사용되었다. 코어들이 하나의 메모리에 연결되어 있는 구조이지만, MPI와 함께 노드 기반의 MPP(Massive Parallel Processor) 시스템 구현이 가능하도록 하였다.

해석에 참여하는 코어 수에 따라서 총 9개의 케이스가 진행되었다. 각 코어당 1개의 노드가 있다고 가정하였고, MPI를 통하여 노드간 정보를 전송 및 공유하였다. 각 노드들은 1개 이상의 계산영역을 갖고 있고, 이 계산영역에는 SPH 입자들이 포함되어 있다. 1,000개로 분할된 영역에서 계산영역의 선택은 자동으로 선택되도록 하였다(Fig. 7~9). 동적 유동을 위해 시간간격은 0.001초 간격으로 2,000번(2초) 해석을 진행하였다.

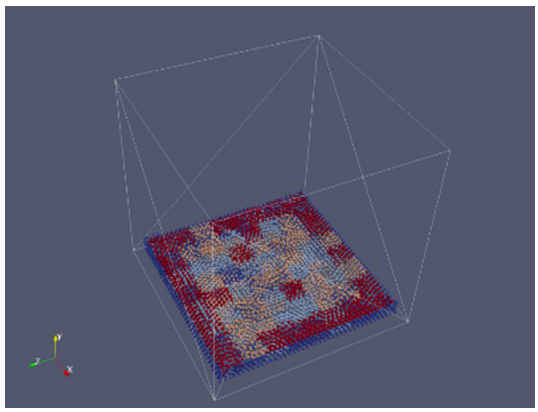

Fig. 7. 4-group decompositions (4 colors)

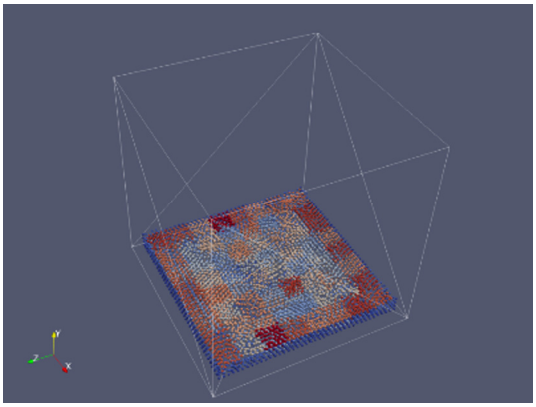

Fig. 8. 16-group decompositions (16 colors)

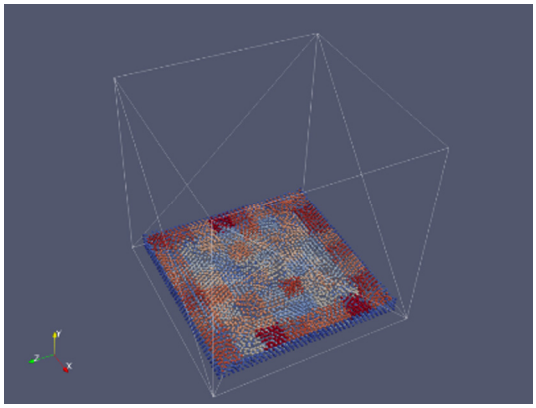

Fig. 9. 28-group decompositions (28 colors)

4.3 해석결과

Fig. 10은 3-2에 제안된 각 과정에서 소요되는 해석시간을 단계별로 나타내는 그래프이다. 해석시간을 가장 많이 차지하고 있는 Step 4는 영역분할에서 계산영역과 잉여영역을 분리하는 단계이고, 계산영역들끼리 인접한 영역들과 경계면들을 공유하는 과정이다. 코어 수에 따라서 큰 차이는 나지 않으나 30개의 코어를 사용하였을 경우는 계산 시간이 급격히 증가함을 볼 수 있다. 이는 물리적으로 28개의 코어 시스템에서 30개의 노드를 발생시켜 일부 노드에서 부하가 증가되었기 때문으로 판단된다. Step 5-1은 시간 반복해석

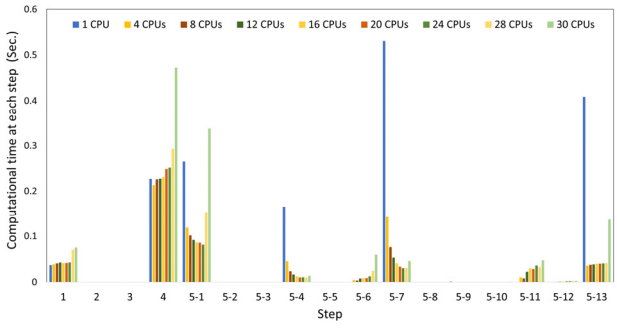

Fig. 10. Computational time at each step

중에 계산영역을 재분할하는 과정인데, 입자의 위치 이동을 고려한 재분배 과정이다. 재분배된 입자에 관한 정보는 모든 영역에서 MPI를 통하여 공유를 하는데, 24 코어까지는 소요시간이 감소하다가 28, 30개의 코어에서는 늘어난 것을 볼 수 있다.

Fig. 11에서 14까지는 해석에서 각 코어별로 소요되는 시간을 정리한 그래프이다. Fig. 13, 14의 Step 5-1에서 볼 수 있듯이 거의 모든 코어가 0.1초 이상의 소요시간을 나타내고 일부 코어는 정상적으로 0.05초 정도에서 끝내는 것을 볼 수 있다. 이는 노드간 정보공유 시 여유 노드가 없을 경우 부하가 발생됨을 의미한다. Step 5-4에서는 입자들이 smoothing 길이 안의 다른 입자들을 찾고 밀도를 결정한다. 이 단계에서의 소요시간은 계산영역 안에 존재하는 입자들의 수가 작을수록 낮게 나타난다. 그리고 노드간 정보 공유가 없기 때문에 계산영역이 총 노드의 수보다 크더라도 계산 속도에는 크게 영향을 미치지 않는다(Fig. 13, 14). Step 5-7은 Eq. (7)을 계산하는 과정인데, 커널함수의 미분, 입자간 거리

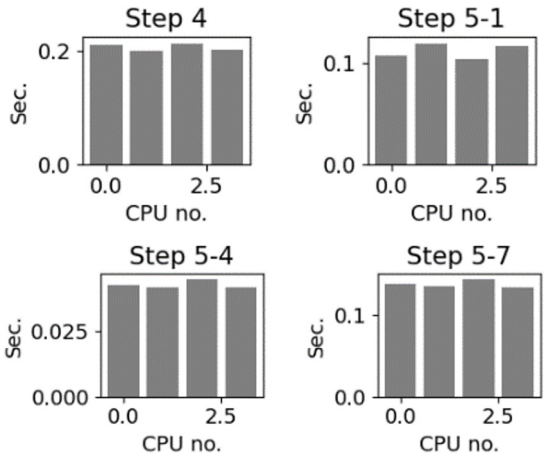

Fig. 11. Computational time at Step 4, 5-1, 5-4, 5-7 for 4 cores (CPUs)

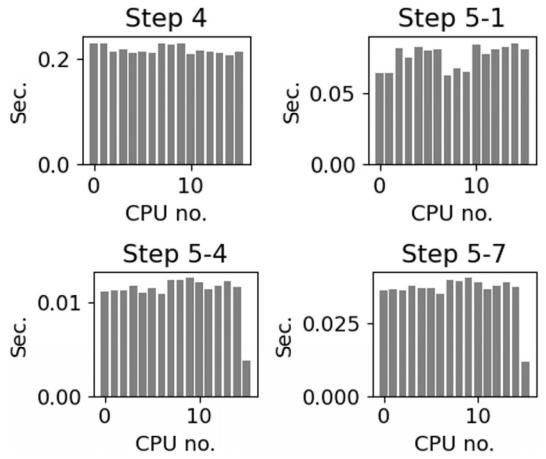

Fig. 12. Computational time at Step 4, 5-1, 5-4, 5-7 for 16 cores (CPUs)

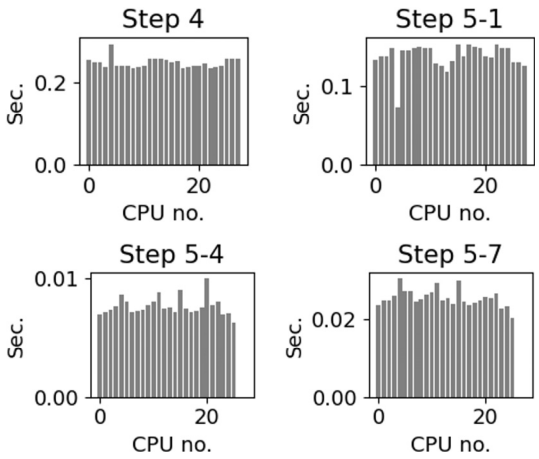

Fig. 13. Computational time at Step 4, 5–1, 5–4, 5–7 for 28 cores (CPUs)

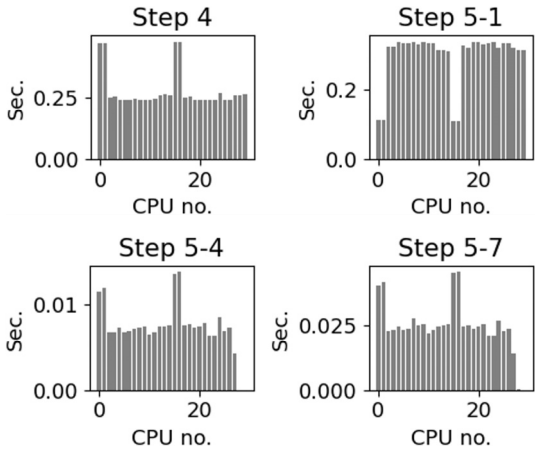

Fig. 14. Computational time at Step 4, 5–1, 5–4, 5–7 for 30 decompositions

계산 등이 포함된다. 본 과정 또한 한 입자에서 주위 입자들 간의 거리를 계산하는 과정에서 소요시간이 크게 나타났다.

Fig. 15는 코어수에 따른 총 해석시간과 병렬효율성 지표 SpeedUp(1 코어런타임 / 병렬 코어런타임)을 나타낸다. 병렬해석 상에서 코어가 증가할수록 해석 속도는 일정한 값에 수렴한다는 Amdahl's Law의 양상을 보이고 있다(Eq. (12)).

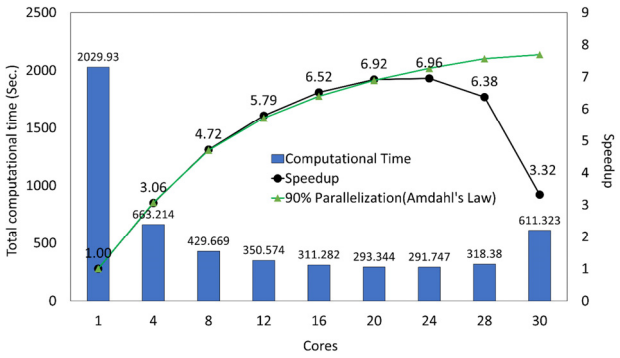

Fig. 15. Total computational time and speedup for each core

특히, 24 노드 해석까지의 결과는 병렬화 90%에 해당되는 Amdahl 그래프의 결과 값과 거의 유사하다. 이는 제안되는 알고리즘의 병렬화가 90%에 근접함을 의미한다.

$$Speedup = \frac{1}{(1-P) + \frac{P}{S}} \tag{9}$$

여기서, P는 프로그램 전체에서 병렬화가 되어 있는 비율, S는 사용된 코어수를 나타낸다.

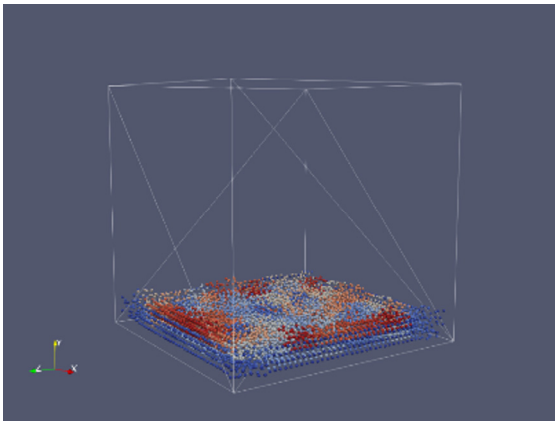

Fig. 16. SPH particles at 0.3 seconds with 28 decompositions

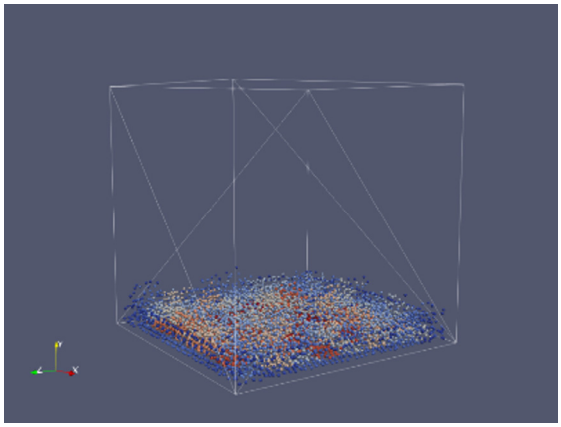

Fig. 17. SPH particles at 0.6 seconds with 28 decompositions

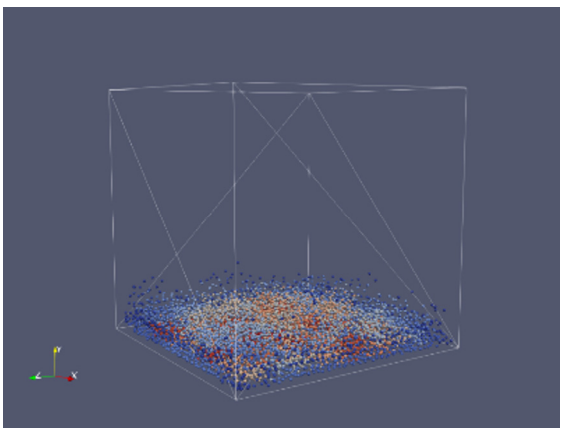

Fig. 18. SPH particles at 0.9 seconds with 28 decompositions

한편, 28개와 30개의 노드를 사용한 해석에서는 반대로 해석시간이 증가하고 있는데, 이는 물리적 노드의 수를 넘어서는 계산영역을 지정함으로써 발생되는 노드의 부하가 원인일 것으로 판단된다.

Fig. 16~18은 28 노드를 사용할 경우 시간에 따른 SPH 입자의 움직임에 대하여 나타내고 있다. 색상은 28 노드가 할당되어 있는 계산영역을 표현하는데 시간에 따라서 입자들에 적용되는 계산영역이 변경됨을 알 수 있다.

5. 결 론

본 논문에서는 SPH 수치해석에서 동적으로 계산영역이 변경되는 영역분할기법에 대해 제안되었다. 그리고 프로그래밍을 통하여 유동체 유동에 관한 병렬해석을 진행하였다. 병렬수치해석 결과 제안된 방법은 병렬화가 90% 근접함을 알 수 있었다. 본 논문에서 제안되는 동적영역분할은 코어들의 로드밸런스를 향상시키기 위해 제안되었다. Fig. 11, 12에서와 같이 계산에 참여하는 모든 코어에서 거의 동일한 해석시간을 볼 수 있었다. 로드밸런스 향상을 통하여 전체 프로그램에서 높은 병렬효율성을 얻을 수 있었다. 하지만 제안된 알고리즘은 직교영역분할의 계산영역과 잉여영역을 구분하는 단계에서 해석시간의 소비가 크게 증가하는 현상이 발생하였다. 이를 개선할 수 있는 알고리즘 개발이 추가로 필요하다.

감사의 글

본 연구는 2019년 정부(교육부)의 재원으로 한국연구재단 이공분야 기초연구사업의 지원을 받아 수행된 연구임(NRF-2019R1I1A1A01059014).

References

1. Benz, W. and Asphaug, E. (1995), Simulations of brittle solids using smooth particle hydrodynamics, Computer Physics Communications, Vol. 87, No. 1, pp. 253~265.
2. Gingold, R.A. and Monaghan, J.J. (1977), Smoothed Particle Hydrodynamics: Theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society, 181, pp. 375~389.
3. Johnson, G. R. and Beissel, S. R. (1998), Normalized smoothing functions for SPH impact computations, International Journal for Numerical Methods in Engineering, Vol. 39, pp. 2725~2741.
4. Kwon, J.H. and Cho, H.C. (2010), Computational modeling of two phase interaction in solid-liquid flows using a coupled SPH-DEM model, ournal of the Korean Institute of Mineral and Energy Resources Engineers, Vol. 47, No. 2, pp. 119~126 (In Korean).
5. Lucy, L.B. (1977), A numerical approach to the testing of the fission hypothesis, Astronomical Journal, Vol. 82, No. 12, pp. 1013~1024.
6. Mahdavi, A. (2018), Pseudo-fluid particles for fluid-rigid body coupling in SPH, KSCE Journal of Civil Engineering, Vol. 22, No. 11, pp. 4194~4204.
7. Monaghan, J. J. (1994), Simulating free surface flow with SPH, Journal of Computational Physics, Vol. 110, pp. 399~406
8. Morris, J. P. (1996), Analysis of smoothed particle hydrodynamics with applications, Ph. D. thesis, Monash University, Australia.
9. 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
10. Sun, W. K., Zhang, L.W. and Lew, K. M. (2020), A smoothed particle hydrodynamics-peridynamics coupling strategy for modeling fluid-structure interaction problems, Computer methods in applied mechanics and engineering, Vol. 371, pp. 1~27.
11. Zienkiewicz, O. C., Taylor, R.L. and Nithiarasu, P. (2005), The Finite Element Method for Fluid Dynamics, Butterworth-Heinemann, pp. 4~27.