

GPU 가속 기술을 이용한 격자 볼츠만법 기반 원유 확산 과정 시뮬레이션

하솔¹ · 구남국^{1*} · 노명일²

¹서울대학교 공학연구소, ²서울대학교 조선해양공학과 및 해양시스템공학연구소

GPU-accelerated Lattice Boltzmann Simulation for the Prediction of Oil Slick Movement in Ocean Environment

Sol Ha¹, Namkug Ku^{1*}, and Myung-II Roh²

¹Engineering Research Institute, Seoul National University

²Department of Naval Architecture & Ocean Engineering and Research Institute of Marine Systems Engineering, Seoul National University

Received 27 March, 2013; received in revised form 30 July, 2013; accepted 21 August, 2013

ABSTRACT

This paper describes a new simulation technique for advection-diffusion phenomena over the sea surface using the lattice Boltzmann method (LBM), capable of predicting oil dispersion from tankers. The LBM is used to solve the pollutant transport problem within the framework of the ocean environment. The sea space is represented by the lattices, where each lattice has the information on oil transportation. Since dispersed oils (i.e., oil droplets) at sea are transported by convection due to waves, buoyancy, and turbulent diffusion, the conservation of mass and many physical oil transport rules were used in the prediction model. Since the LBM is modeled using the uniform lattices and simple rules, it can be easily accelerated by the parallel mechanism, for example, GPU-accelerated method. The proposed model using the LBM is used to simulate a simple pollution event with the oil pollutants of 10,000 kL. The simulation results indicate that the LBM method accelerated with the GPU is 6 times faster than that without the GPU.

Key Words: CUDA (Compute Unified Device Architecture), GPGPU (General-Purpose computing on Graphics Processing Units), Lattice Boltzmann method, Oil slick movement and spreading

1. 서 론

1.1 연구 배경 및 필요성

2010년 12월 7일 오전 7시 15분경 태안군 만리

포 북서방 약 8km 해상에서 예인선으로부터 분리된 부선과 유조선(홍콩선적 ‘허베이 스피릿’ 호)의 충돌로 인해 대량의 원유가 유출되는 사고가 발생하였다. 당시 발생한 사고의 원유 유출량은 국내 원유 유출 사고 중 가장 큰 규모인 12,547 kL로서 주변 해양 환경에 막대한 손실을 입혔다. 해양 유류 오염 사고는 유출 지점과 주변 지역의 해양과

*Corresponding Author, knk80@snu.ac.kr
©2013 Society of CAD/CAM Engineers

연안 생태계에 치명적인 영향을 미치기에 유출된 원유의 확산을 초기에 방지하는 것이 중요하다.

사고로 인해 유출된 원유는 유출 지점으로부터 주변 지역으로 확산되며, 바람, 파도, 해류, 그리고 해수의 온도와 같은 주변 환경에 따라 유류가 확산되는 방향과 그 정도가 달라진다. 특히 초기 유출 후 원유가 다른 지역으로 확산되는 것을 차단하기 위해서는 주변 환경을 고려하여 원유의 확산 방향과 확산 정도를 단시간 내에 예측하는 작업이 요구된다. 따라서 본 논문에서는 초기 원유 유출 시 주변 해양 환경을 고려하여 원유의 확산 방향과 확산 정도를 단시간 내에 예측하기 위해 격자 볼츠만 방법에 GPU 기반 병렬 처리 방식을 적용하여 원유의 확산 과정을 예측하는 시뮬레이션을 수행하였다.

1.2 관련 연구 현황

초기 원유 유출 후 원유의 확산 과정을 예측하는 방법으로는 시스템을 연속체로 보고 해석하는 거시적(macroscopic)인 방법과 시스템을 구성하는 입자의 행동을 해석하여 결과를 얻는 미시적(microscopic)인 방법으로 나뉜다. 거시적인 관점에서의 해석 방법은 주변 해양 환경과 원유 확산 과정에 대한 지배 방정식을 정립하고, 수치적인 방법을 이용하여 해를 구하는 과정으로 현상을 해석한다. 예로서 Inan 등은 diffusion 방정식을 이용하여 원유의 확산 과정을 예측하였으며¹¹⁾, Halil 등은 advection-diffusion 방정식을 이용하여 오염 물질의 이동 및 확산 과정을 예측하였다¹²⁾. 거시적인 방법은 지배 방정식의 정확도에 따라 원유의 확산 과정을 보다 정확하게 예측할 수 있으나, 계산량이 많고 예측 결과를 얻기까지 계산 시간이 오래 걸리는 한계점이 있다.

미시적인 관점에서의 해석 방법으로는 cellular automata, lattice gas automata, lattice Boltzmann method 등이 있다. Cellular automata, lattice gas automata에 기반한 모델은 원유가 유출된 주변 지역을 균일한 격자로 분할한 후 각 격자에 일정한 규칙을 부여하여 원유의 확산 과정을 예측하는 모델을 말한다. Nakano 등은 공간을 정방형 격자로 나누고 각 cell이 가진 원유의 양이 단위 시간 동안 이웃한 4개의 cell로 균일하게 확산된다는 간단한 규칙을 이용하여 원유의 확산 과정을 예측하였다¹³⁾. Sirakoulis 등은 각 cell에 지형, 원유량, 해류

의 방향과 속도, 바람의 방향과 속도에 대한 6가지 속성을 부여하고, 주변 cell의 영향을 고려하여 각 cell의 원유량을 예측할 수 있는 cellular automata 모델을 제시하였다¹⁴⁾. 또한 해수의 온도에 따라 원유가 증발하는 특성도 고려하였다. Rusinovic 등과 Shyue 등은 Sirakoulis 등이 제시한 모델에 파도에 의한 수직 방향으로의 원유 확산(vertical transportation), 원유의 용해(dissolution), 해안선에서의 침전(deposition), 유상화(emulsification) 등을 추가로 고려하였다^{15,16)}. Ha 등은 Cell-DEVS(discrete event system specification) 형식론을 기반으로 한 oil spreading cellular automata 모델을 제시하고 이를 이용하여 원유의 확산 과정을 시뮬레이션 하였다¹⁷⁾.

Deng 등, Chopard 등, Shi와 Guo, Tubbs, Zhou은 오염 물질의 확산 과정을 보다 정확하게 예측하기 위해 lattice Boltzmann method를 도입하였다¹⁸⁻¹²⁾. 이들은 오염 물질의 이동과 확산을 표현하는 advection-diffusion 방정식과 수학적으로 동일한 특성을 가지는 lattice Boltzmann model을 제시하고 이를 이용하여 오염 물질의 확산 과정을 시뮬레이션 하였다. Tubbs와 Tsai는 lattice Boltzmann method의 병렬 처리 특성을 착안하여 GPU를 이용한 병렬 처리 기법을 적용함으로써 계산 수행 속도를 향상시키는 연구를 수행하였다¹³⁾.

본 논문에서는 원유의 확산 과정을 예측하기 위해 advection-diffusion 방정식에 기반하여 lattice Boltzmann method를 이용한다. 또한 lattice Boltzmann method의 병렬 처리 특성을 고려하여 GPU를 이용한 병렬 처리 기법을 도입함으로써 계산 수행 속도 향상을 도모하였다. 본 논문의 2장에서는 원유 확산에서의 수학적 모델에 대해 설명한다. 3장에서는 2장에서 설명한 원유 확산 과정의 수학적 모델을 근사화 하기 위한 lattice Boltzmann 모델을 제시한다. 4장에서는 제시한 lattice Boltzmann 모델에 GPU를 이용한 병렬 처리 기법을 도입하여 계산을 수행하는 과정을 설명한다. 마지막으로 5장에서는 시뮬레이션 결과를, 그리고 6장에서는 본 논문의 결론에 대해 서술한다.

2. 원유 확산 과정 예측을 위한 수학적 모델

본 논문에서는 원유의 유출 후 이동, 확산 과정을 예측하기 위해 오염 물질의 이동 및 확산에 사

용되는 이류-확산 방정식(advection-diffusion equation)을 이용하였다. 2차원에서의 이류-확산 방정식(advection-diffusion equation)은 다음과 같이 주어진다^[8].

$$\frac{\partial C}{\partial t} + \frac{\partial u_i C}{\partial x_i} = D \frac{\partial^2 C}{\partial x_i^2} \quad (1)$$

여기서 C 는 오염 물질의 농도, u 는 유속, D 는 확산 계수를 의미한다. 만약 외부에서 지속적으로 유입되는 유류가 존재한다면 이를 생성(source) 항으로 고려할 수 있으며 식 (2)는 다음과 같이 표현된다.

$$\frac{\partial C}{\partial t} + \frac{\partial u_i C}{\partial x_i} = D \frac{\partial^2 C}{\partial x_i^2} + F(\vec{x}, t) \quad (2)$$

여기서 F 는 생성(source) 항이다. 이류-확산 방정식을 이용하여 원유의 확산 과정을 예측하기 위해서는 주어진 방정식의 해를 구해야 한다. 방정식의 해를 수치적으로 구하는 방법으로는 finite difference method, finite element method를 비롯한 다양한 수치 해석 방법이 존재한다. 본 논문에서는 기존에 널리 사용되는 수치 해석 방법 대신 lattice Boltzmann method(LBM)를 이용하여 수치적인 해를 구하는 방법을 적용하였다. LBM은 기존의 수치 해석 방법과 비교하여 공간을 균일 격자로 나누고 각 격자와 인접 격자들 간의 규칙을 정의함으로써 주어진 물리 현상을 비교적 손쉽게 정의할 수 있다. 또한 모든 격자의 규칙이 동일하게 정의되기에 병렬 처리에 적합하다는 특징이 있다.

3. 원유 확산 예측을 위한 Lattice Boltzmann 모델

3장에서는 원유 확산 과정 예측을 위한 lattice Boltzmann 모델에 대해 설명한다. 3.1절에서는 먼저 lattice gas automata와 lattice Boltzmann method의 개념을 간략히 설명한다. 3.2절에서는 이류-확산 방정식을 근사화하는 lattice Boltzmann 모델에 대해 설명한다.

3.1 Lattice Gas Automata와 Lattice Boltzmann Method

Lattice Boltzmann method는 lattice gas automata(LGA)의 이론을 기반으로 한다. Hardy 등은 Fig. 1(a)와 같이 공간을 정방형 격자로 분할하고 격자

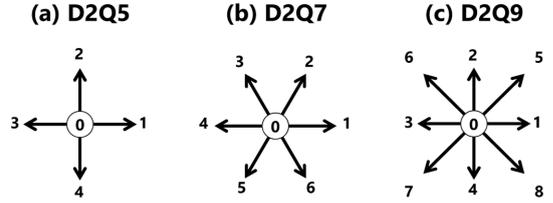


Fig. 1 Examples of the vectors in lattices: (a) D2Q5, (b) D2Q7, (c) D2Q9

상의 입자가 4방향으로 이동하면서 다른 입자와 충돌하는 LGA 모델을 제안하였다^[4]. 향후 이를 보완한 육각 격자 형태의 LGA 모델이 Frisch 등에 의해 제안되었다^[15](Fig. 1(b) 참조). 육각 격자 기반의 모델은 Navier-Stokes 방정식과 수학적으로 동치임이 증명되었기에 유체를 포함한 여러 분야에서 널리 사용되었다.

LGA는 주어진 공간을 격자 형태로 이산화하여 표현하기에 복잡한 경계면 형태의 정의와 경계면의 동적 변화를 취급하기에 용이하다. 또한 입자들 사이의 국소적인 상호 작용을 전체 영역의 모든 격자에 대해 한꺼번에 계산하기에 병렬 처리에 적합하다는 특징이 있다. 그러나 LGA는 입자의 이동과 충돌에서 착안한 Boolean 연산을 기반으로 하기에 얻어진 결과에는 통계적인 흠여짐(statistical noise)이 발생하는 문제점이 있다.

Lattice Boltzmann method(LBM)는 LGA의 noise 문제를 해결하기 위해 입자들의 충돌을 Boolean 연산으로 표현하지 않고 연속적인 분포 함수로 표현한 방법이다. LBM에서 일반적으로 흐름의 영역은 규칙적인 격자에 의해 이산화되며, 격자 상에서 가상의 입자는 충돌(collision)과 이동(propagation)을 반복한다. LGA와의 차이점으로는 격자 상에 위치하는 가상 입자의 유무를 0과 1로 표현하는 것이 아니라 연속적인 입자 밀도 분포 함수로 표현하는 것에 있다. 따라서 LGA와 달리 통계적인 흠여짐(statistical noise)이 제거되어 현상 평가에 적합하다. LBM에 기반한 일반 진화 방정식(general evolution equation)은 입자 밀도 분포 함수인 f_i 를 이용하여 다음과 같이 표현된다.

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = \Omega_i(\vec{x}, t) \quad (3)$$

D2Q7 모델의 경우 격자 구성이 육각 형태로 정의되기 때문에 관심 영역과 경계 조건을 다양하게 표현하기에는 다소 부족하다. 따라서 본 논문에서

는 다양한 형태의 경계 조건과 관심 영역을 표현하기 위해 Fig. 1(c)의 D2Q9 모델을 사용하여 공간을 분할하였다.

3.2 이류-확산 방정식을 위한 Lattice Boltzmann Formulation

Lattice Boltzmann method에서 충돌 연산자 (collision operator) $\Omega_i(\vec{x}, t)$ 는 기본적으로 충돌 시 질량, 운동량 및 에너지 보존의 법칙을 따른다. 충돌은 입자 간의 상호작용에 의한 운동의 변화이기 때문에 $\Omega_i(\vec{x}, t)$ 는 입자 밀도 분포 함수 f_i 를 이용하여 정의되어야 한다. Qian 등은 충돌 연산자를 간략히 정의하기 위해 기존에 Bhatnagre 등이 볼츠만 방정식에 사용한 알고리즘을 기반으로 한 BGK(Bhatnagre-Gross-Krook) 연산자^[16]를 다음과 같이 제안하였다^[17].

$$\Omega_i(\vec{x}, t) = -\frac{\Delta t}{\tau} [f_i(\vec{x}, t) - f_i^{(0)}(\vec{x}, t)] \quad (4)$$

여기서, τ 는 완화 시간이며, i 는 인접한 격자로의 방향, $f_i^{(0)}$ 는 평형 상태에서의 입자 밀도 분포 함수를 의미한다. BGK 연산자를 식 (3)에 대입하면 다음과 같다.

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{\Delta t}{\tau} [f_i - f_i^{(0)}] \quad (5)$$

위 식은 입자 밀도 분포 함수 f_i 가 $\Delta t/\tau$ 의 속도로 평형 상태에서의 입자 밀도 분포 함수 $f_i^{(0)}$ 에 접근하는 것을 의미한다. 식 (5)은 단일 완화 시간에 의해 영향을 받으므로 single relaxation time lattice Boltzmann model이라고도 불린다. 외부에서 유입되는 원유가 있을 경우 이를 생성(source) 항 F_i 으로 고려하여 다음과 같이 식을 수정할 수 있다.

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) - f_i(\vec{x}, t) = -\frac{\Delta t}{\tau} [f_i - f_i^{(0)}] + \Delta t F_i(\vec{x}, t) \quad (6)$$

평형 상태에서의 입자 밀도 분포 함수 $f_i^{(0)}$ 는 지배 방정식의 해와 직접적인 관계를 가진다. 따라서 원유의 확산 과정을 예측하기 위한 lattice Boltzmann 모델을 정의하기 위해서는 지배 방정식인 이류-확산 방정식을 고려하여 평형 상태에서의 입자 밀도 분포 함수 $f_i^{(0)}$ 를 정의해야 한다.

$f_i^{(0)}$ 는 예를 들어 다음과 같은 함수형을 사용할 수 있다.

$$f_i^{(0)} = \Phi \omega_i [A + B \vec{c}_i \cdot \vec{u} + C (\vec{c}_i \cdot \vec{u})^2 + D |\vec{u}|^2] \quad (7)$$

여기서 u 는 평형 상태에서의 유체의 속도이다. Φ 는 상수로서 지배 방정식에 따라 밀도나 온도의 값을 가질 수 있으며 다음과 같이 정의된다.

$$\Phi = \sum_{i=0}^{i=n} f_i^{(0)} = \rho \quad (8)$$

n 은 lattice 의 개수로써 D2Q9의 경우 $n=9$ 이다. 식 (7)에서 A~D의 계수는 질량, 운동량, 에너지 보존 법칙을 만족하면서 주어진 지배 방정식이 재현되도록 결정할 수 있다. Deng 등은 이류-확산 방정식을 위해 다음과 같은 평형 상태에서의 입자 밀도 분포 함수 $f_i^{(0)}$ 를 제시하고, 이를 사용한 lattice Boltzmann model이 이류-확산 방정식을 만족함을 확인하였다.

$$f_i^{(0)} = \omega_i \rho \left[1 + \frac{\vec{c}_i \cdot \vec{u}}{c_s^2} + \frac{0.5 (\vec{c}_i \cdot \vec{u})^2}{c_s^4} - \frac{0.5 |\vec{u}|^2}{c_s^2} \right] \quad (9)$$

이 때 c_i 는 인접한 이웃 lattice로의 방향 벡터, $c_s = c/\sqrt{3}$ 이며, c 는 격자의 단위 길이이다. D2Q9에서 ω_i 는 다음과 같다.

$$\omega_i = \begin{cases} \frac{4}{9}, & i = 0 \\ \frac{1}{9}, & i = 1 \sim 4 \\ \frac{1}{36}, & i = 5 \sim 8 \end{cases} \quad (10)$$

식 (6)에서 F_i 는 i 방향으로 작용하는 외력으로써 다음과 같이 주어진다.

$$\omega_i = \omega_i F \left[1 + \frac{c_s^{-2} (\vec{c}_i \cdot \vec{u}) (\tau - 0.5)}{\tau} \right] \quad (11)$$

Lattice gas automata로부터 발전한 lattice Boltzmann method는 공간을 Fig. 1과 같이 균일한 격자로 분할하여 국소적인 입자들의 충돌(collision)과 이동(propagation)으로 시스템의 특성을 정의하는 것이 특징이다. 각 격자는 식 (6)과 같이 동일하게 정의된 일반 진화 방정식(general evolution equation)을 이용하여 매 단위 시간마다 자신의 상태를 동시에 갱신한다. 따라서 lattice Boltzmann

method는 동일한 알고리즘을 이용하여 수많은 격자의 상태를 한꺼번에 갱신하기에 별도의 수정 없이 쉽게 병렬 연산을 적용할 수 있다는 장점이 있다. 본 논문에서는 3장에서 설명한 lattice Boltzmann 모델에 GPU 기반의 병렬 연산을 적용하여 시뮬레이션 속도 향상을 도모하였다.

4. GPU 기반 병렬 연산

4.1 병렬 처리 방법의 종류

본 연구에서는 lattice Boltzmann method의 계산 속도 향상을 위해 매 단위 시간마다 수행하는 계산을 동시다발적으로 수행할 수 있는 병렬 처리 방법을 도입하였다. 병렬 처리 방법은 순차 처리 방법과 달리 주어진 반복 연산을 서로 다른 프로세스에 분배한 후 동시에 처리하는 방식으로 단순하면서도 반복 횟수가 많은 연산을 보다 빠른 시간 내에 처리할 수 있다.

병렬 처리 방법으로는 클러스터 컴퓨팅(Cluster Computing), 그리드 컴퓨팅(Grid Computing), 그리고 GPGPU(General-Purpose computing on Graphics Processing Units) 방법 등이 있다. 본 논문에서는 다른 방법에 비해 쉽게 접근할 수 있고 가용비가 뛰어난 GPGPU 방법을 이용하였다.

GPGPU 방법은 프로세서가 부담을 갖는 작업을 그래픽 프로세서가 대신 처리함으로써 시스템의 효율을 높이는 것을 기본으로 한다^[18].

CPU와 GPU의 가장 큰 차이점은 연산구조이다. 기존의 연산은 하나의 명령에 의해 하나의 DATA를 얻는 SISD(Single Instruction Single Data) 구조이다. GPU의 경우 다수의 형상 정보(vertices, edges, faces 등)를 기반으로 하는 수많은 연산을 처리해야 하기에 하나의 명령을 통해 많은 data를 동시에 처리할 수 있는 SIMD(Single Instruction Multi Data) 구조를 가지고 있다. 따라서 GPU를 이용하면 Host에서 하나의 명령(instruction)을 내리면 data를 동시에 처리할 수 있는 병렬 연산이 가능하다.

본 논문에서는 nVIDIA에서 제공하는 GPGPU 방법인 CUDA(Compute Unified Device Architecture)라는 기술을 도입하여 병렬 처리 방법을 적용하였다. 특히 기존 CPU를 이용하여 순차 처리 방식으로 연산을 수행하던 방법에서 벗어나 GPU의 hardware resource를 이용한 병렬 처리 방법을 이용함으로써 계산 속도를 향상시킬 수 있었다.

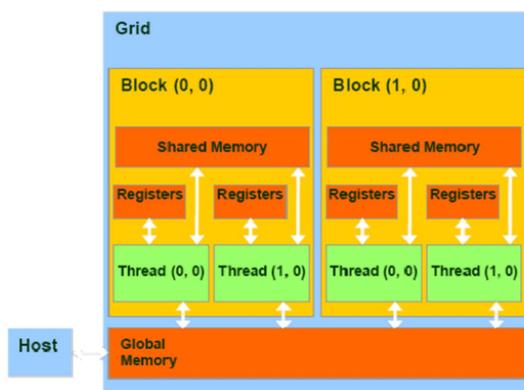


Fig. 2 Configuration of GPU^[19]

CUDA(Compute Unified Device Architecture)는 그래픽 처리 장치(Graphic Processing Unit; GPU)에서 수행하는 (병렬 처리) 알고리즘을 C 프로그래밍 언어를 비롯한 산업 표준 언어를 사용하여 작성할 수 있도록 하는 GPGPU 기술이다. CUDA를 통해 개발자들은 CUDA GPU 안 병렬 계산 요소 고유의 명령어 집합과 메모리에 접근할 수 있으며, 최신 nVIDIA GPU를 효과적으로 사용할 수 있다.

GPU는 기본적으로 컴퓨터의 메모리와는 독립적으로 존재하는 device로서 CPU로부터 데이터와 명령을 전달받아 연산을 수행한다. 따라서 GPU 상에서 연산을 수행하기 위해 데이터를 저장하는 메모리가 별도로 필요하다. CUDA에서는 컴퓨터를 ‘호스트’, GPU를 ‘디바이스’라고 부르며, 컴퓨터 상의 메모리를 ‘호스트 메모리’, GPU의 메모리를 ‘디바이스 메모리’라 부른다(Fig. 2 참조).

4.2 GPU를 이용한 Lattice Boltzmann Method의 병렬 처리 과정

CUDA를 이용하여 GPU를 이용한 lattice Boltzmann method의 병렬 처리 과정은 Fig. 3과 같다. 병렬 처리 과정에 대한 상세한 설명은 다음과 같다.

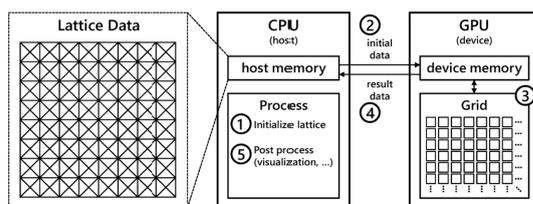


Fig. 3 Sequence of parallel process by using lattice Boltzmann method

격자(lattice) 초기화

계산 수행을 위해 정의한 격자(lattice)를 초기화한다(Fig. 3①). 이 때 초기화된 데이터는 CPU에서 관리하는 메모리인 호스트 메모리에 저장된다. 격자를 초기화 하는 과정에 대한 pseudo code는 다음과 같다.

```
// allocate 호스트 메모리
host_lattices = AllocateLattices(x_size, y_size)

// initialize lattices
For each lattice in host_lattices
    lattice.Initialize() // initialize lattice
End For each
```

CPU에서 GPU로의 데이터 복사

초기화가 완료되면 GPU에서 연산을 수행하기 위해 호스트 메모리에 저장한 데이터를 GPU의 디바이스 메모리로 복사한다(Fig. 3②). 디바이스 메모리로 데이터를 복사하기 위해 CUDA에서 사전에 정의된 함수를 이용한다. 이에 대한 pseudo code는 다음과 같다.

```
// allocate 디바이스 메모리
device_lattices = cudaMalloc(x_size * y_size)

// copy from host to device
cudaMemcpy(host_lattices, device_lattices, x_size * y_size)
```

GPU에서의 병렬 연산 수행

GPU는 복사한 데이터를 이용하여 다수의 산술 연산장치의 집합인 Grid를 이용하여 병렬 연산을 수행한다(Fig. 3③). Lattice Boltzmann method는 충돌(collision)과 이동(propagation)의 2가지 연산을 포함한다. 충돌과 이동 연산을 동시에 수행할 경우 충돌 연산이 이웃 격자에서의 이동 연산에 영향을 받을 수 있다. 따라서 병렬 처리를 위해 충돌 연산과 이동 연산을 다음과 같이 분리하였다.

$$f_i(\vec{x}, t) = f_i(\vec{x}, t) - \frac{\Delta t}{\tau} [f_i - f_i^{(0)}] + \Delta t F_i(\vec{x}, t) \quad (12)$$

$$f_i(\vec{x} + \vec{c}_i \Delta t, t + \Delta t) = f_i(\vec{x}, t) \quad (13)$$

병렬 연산을 수행하기 위해서 먼저 GPU상의 그리드를 분할하여 이를 다수의 블록으로 구분한다. 그리고 블록 별로 데이터를 분할한 후 이를 이용하여 병렬 연산을 수행한다. 병렬 연산은 먼저 생성(source)항에 의한 외력과 평형 상태에서의 입자 밀도 분포 함수를 계산하고 식 (12)을 이용하

여 충돌 연산을 수행한다. 병렬 처리로 실행되는 충돌 연산의 종료는 CUDA에서 제공하는 시간 동기화 기능을 이용하여 확인한다. 충돌 연산이 종료되면 다음으로 식 (13)를 이용하여 이동 연산을 수행한다. 이에 대한 pseudo code는 다음과 같다.

```
// setup grid property
SetGridProperty(threads, blocks)

// calculate external force (source)
CalcExternalForce<<<blocks, threads>>>(device_lattices)

// calculate equilibrium distribution
CalcEqDistribution<<<blocks, threads>>>(device_lattices)

// collision
CollideLattice<<<blocks, threads>>>(device_lattices)

// time synchronize
__syncthreads()

// propagation
Propagate<<<blocks, threads>>>(device_lattices)
```

GPU에서 CPU로의 데이터 복사 및 후처리

단위 시간만큼 진행 후 GPU에서의 연산이 종료되면 계산 결과를 CPU의 호스트 메모리로 다시 복사한다(Fig. 3④). CPU에서는 호스트 메모리에 저장된 데이터를 이용하여 가시화 등의 후처리 작업을 진행한다(Fig. 3⑤). 이에 대한 pseudo code는 다음과 같다.

```
// copy from device to host
cudaMemcpy(device_lattice, host_lattice, x_size * y_size)

// post process
Visualize(host_lattices)
```

5. 원유 확산 시뮬레이션

3, 4장에서 정의한 lattice Boltzmann 모델의 기능을 확인하기 위해 다음과 같이 가정하여 시뮬레이션을 구성하였다.

- ① 300(m) × 300(m) 크기의 영역에 대해 900개의 격자를 생성한다. (격자의 크기는 사용자에게 임의로 조정 가능하다.)
- ② 초기에 유출된 원유의 양은 10,000 kL로써 임의 위치의 격자에 입력한다.

시뮬레이션 실행 환경은 Table 1과 같다.

Table 1 Test environment for the oil spill simulation

항목	실행 환경
CPU	Intel® Core™ i7-2600K Processor (8M Cache, 3.40@4.50 GHz)
VGA	nVIDIA GeForce GTX 460 - CUDA Cores: 336 Units - Graphics Clock: 675 MHz - Processor Clock: 1,350 MHz
SSD	Intel G2 160GB x 2 (RAID 0)
OS	Microsoft Windows 7 64-bit with Service Pack 1

Table 2 Comparison of the executing time between CPU and GPU

항목	계산 시간 (단위: 초)
CPU 이용	0.530
GPU 이용	0.084

동일 예제에 대해 CPU를 이용한 연산 처리 속도와 GPU를 이용한 병렬 연산 처리 속도의 비교 결과는 Table 2와 같다. Table 2에서의 계산 시간은 lattice Boltzmann 모델을 이용하여 단위 시간만큼 진행할 때의 계산 소요 시간을 측정하는 것이다. 계산 속도 비교 결과 약 6배 정도의 속도 향상이 있었으며, CPU와 GPU 간의 데이터 교환, 후처리 연산 등의 부가적인 계산 시간을 고려한다면 약 2~3배 정도의 속도 향상이 예상된다.

6. 결론 및 향후 연구 계획

본 논문에서는 원유 유출에 따른 확산 과정을 예측하기 위해 lattice Boltzmann 모델을 제안하고 이를 이용하여 원유 확산 시뮬레이션을 수행하였다. 제안한 lattice Boltzmann 모델은 이류-확산 방정식을 기반으로 한다. 계산 속도 향상을 위해 lattice Boltzmann 모델에 GPU를 이용한 병렬 처리 기법을 적용하였다. 제안한 lattice Boltzmann 모델의 기능을 검증하기 위해 가상의 공간에 대해 원유 확산 시뮬레이션을 수행하였다. 향후에는 다양한 경계 조건을 추가하고 태안 반도 원유 유출 사고와 같은 실제 사례를 대상으로 시뮬레이션을 수행한 후 비교 검증할 예정이다.

감사의 글

본 연구는 (a) 지식경제부 산업원천기술개발사

업 (10035331, 시뮬레이션 기반의 선박 및 해양플랜트 생산 기술 개발), (b) 서울대학교 공학연구소, (c) 서울대학교 해양시스템공학연구소의 지원으로 이루어진 연구 결과의 일부임을 밝히며, 이에 감사 드립니다.

참고문헌

- Inan, A. and Balas, L., 2010, An Application of 2D Oil Spill Model to Mersin Coast, *WSEAS Transactions on Environment and Development*, 6(5), pp.345-354.
- Halil, K., 2007, A Third-order Upwind Scheme for the Advection-diffusion Equation Using Spreadsheets, *Advances in Engineering Software*, 38(9), pp.688-697.
- Nakano, T., Kasegawa, J. and Morishita, S., 1998, Coastal Oil Pollution Prediction by a Tanker Using Cellular Automata, *OCEANS '98 Conference Proceedings*, 3, pp.1324-1328.
- Sirakoulis, G., Karafyllidis, I., Thanailakis, A. and Tsalides, P., 2003, A Methodology for Modeling Ecological Systems Based on Cellular Automata, *WSEAS Transactions on Computers*, 2(4), pp.982-990.
- Rusinovic, Z. and Bogunovic, N., 2006, Cellular Automata Based Model for the Prediction of Oil Slicks Behavior, *Proceedings of 28th International Conference on Information Technology Interfaces*, pp.569-574.
- Shyue, S., Chung, J.S., Hong, S.W., Nagata, S., Sarmento, A. and Koterayama, W., 2007, Oil Spill Modeling Using 3D Cellular Automata for Coastal Waters, *Proceedings of the International Society of Offshore and Polar Engineers Conference*, pp.546-553.
- Ha, S., Ku, N.K. and Lee, K.Y., 2011, Battle Space Model Based on Lattice Gas Automata for Underwater Warfare Simulation, *Proceedings of Asia Simulation Conference 2011*, pp.361-376.
- Deng, B., Shi, B.C. and Wang, G.C., 2005, A New Lattice Bhatnagar-Gross-Krook Model for the Convection-diffusion Equation with a Source Term, *Chinese Physics Letters*, 22(2), pp.267-270.
- Chopard, B., Falcone, J.L. and Latt, J., 2009, The Lattice Boltzmann Advection-diffusion Model Revisited, *The European Physical Journal Special Topics*, 171(1), pp.245-249.
- Shi, B. and Guo, Z., 2009, Lattice Boltzmann Model for Nonlinear Convection-diffusion Equations, *Physical Review E*, 79(1), p.016701.
- Tubbs, K., 2010, *Lattice Boltzmann Modeling*

- for Shallow Water Equations Using High Performance Computing, Doctoral Thesis, Louisiana State University.
12. Zhou, J.G., 2011, Lattice Boltzmann Method for Advection and Anisotropic Dispersion Equation, *Journal of Applied Mechanics*, 78(2), pp.021007-5.
 13. Tubbs, K.R. and Tsai, F.T., 2011, GPU Accelerated Lattice Boltzmann Model for Shallow Water Flow and Mass Transport, *International Journal for Numerical Methods in Engineering*, 86(3), pp.316-334.
 14. Hardy, J., Pomeau, J. and de Pazzis, O., 1973, Time Evolution of a Two-dimensional Classical Lattice System, *Physical Review Letters*, 31(5), pp.276-279.
 15. Frisch, U., Hasslacher, B. and Pomeau, Y., 1986, Lattice-gas Automata for the Navier-Stokes Equation, *Physical Review Letters*, 56(13), pp.1505-1508.
 16. Bhatnagar, P.L., Gross, E.P. and Krook, M., 1954, A Model for Collision Processes in Gases, *Physical Review*, 94(3), pp.511-525.
 17. Qian, Y.H., D'Humieres, D. and Lallemand, P., 1992, Lattice BGK Models for Navier-Stokes Equation, *Europhysics Letters (EPL)*, 17(5), pp.479-484.
 18. Nguyen, H., 2007, *GPU Gems 3: Programming Techniques for High Performance Graphics and General-Purpose Computation*, Addison-Wesley Education Publishers, Inc.
 19. Nvidia, C., 2007, *Compute Unified Device Architecture Programming Guide*, NVIDIA: Santa Clara, CA.



하 솔

2003년 서울대학교 조선해양공학과 학사
 2013년 서울대학교 조선해양공학과 박사
 2013년 3월~ 서울대학교 공학연구소 선임연구원
 관심분야: 해양 플랜트 설계 최적화, 시스템 신뢰도 분석, 모델링 & 시뮬레이션, 수중운동체(잠수함, 어뢰) M&S, 격자 기반 시뮬레이션(Cellular Automata, Lattice Gas Automata, Lattice Boltzmann Method)



구 남 국

2004년 서울대학교 조선해양공학과 학사
 2012년 서울대학교 조선해양공학과 박사
 2012년~2013년 2월 서울대학교 해양기술인력양성사업단 연수연구원
 2013년 3월~ 서울대학교 공학연구소 선임연구원
 관심분야: 해양 플랜트, 다물체계 동역학 및 제어, 용접/전처리 로봇



노 명 일

1998년 서울대학교 조선해양공학과 학사
 2000년 서울대학교 조선해양공학과 석사
 2005년 서울대학교 조선해양공학과 박사
 2005년~2007년 서울대학교 공학연구소/해양시스템공학연구소/BK21해양인력양성사업단 연구원
 2007년~2013년 울산대학교 조선해양공학부 조교수
 2013년~현재 서울대학교 조선해양공학과 부교수
 관심분야: 전산선박설계, 시뮬레이션 기반 설계, 최적 설계, 해양 구조물 설계, CAD/CAM/CAE, CAGD