

신속한 오염사고 대응을 위한 입자 분산 모형의 속도 개선 및 평가

Particle Dispersion Model Speed Improvement and Evaluation for Quick Reaction to Pollutant Accidents

신재현*, 성호제*, 박인환**, 이동섭*

한국건설기술연구원 국토보전연구본부*, 서울과학기술대학교 건설시스템공학과**

Jaehyun Shin(jaehyunshin@kict.re.kr)*, Hoje Seong(hoje.seong@kict.re.kr)*,
Inhwan Park(ihpark@seoultech.ac.kr)**, Dong Sop Rhee(dsree@kict.re.kr)*

요약

본 연구에서는 오염물 사고에 대한 신속한 대응을 위하여 입자 분산 모형을 개발 및 개선하고 병렬 프로그램을 적용한 모의 속도 증가와 그 분석을 통하여 속도개선 결과를 평가하였다. 개발된 모형은 전단류 분산이론을 따르면서 수평 혼합 과정은 전단이송, 연직 혼합 과정은 연직배열 알고리즘을 이용한 난류 및 입자 확산을 구현하였다. 오염사고에 신속하게 대응하기 위해 모형 속도 개선을 위하여 OpenMP를 활용한 병렬 프로그래밍으로 멀티코어 적용 알고리즘을 적용하였다. 병렬 프로그래밍 적용 결과, 가상 사행수로에서 기준 소요시간 내로 모의가 가능한 입자 및 활용 코어 개수의 관계를 도출할 수 있었다. 이 연구 결과로 신속한 수질 오염사고 대응을 위한 적절한 모의 조건을 구성할 수 있게 되어 모형의 활용성을 증대할 수 있었다.

■ 중심어 : | 과학기술 | 입자 분산 모형 | 오염물 확산 | 모의 속도 | 입자 | 수질 사고 | OpenMP | 배열 알고리즘 | 병렬 프로그래밍 |

Abstract

This study deals with the development and improvement of a particle dispersion model for quick response to water pollutant accidents. The developed model is based on the shear dispersion theory where vertical mixing is done by step by step mixing by vertical and molecular diffusion algorithm. For the quick response to chemical accidents, an algorithm for multi-core modeling for the particle dispersion model is applied. After the application of multi-core operation using OpenMP directives to the model, the relation for the calculation time and particle size were determined along with the number of cores used for parallel programming to determine the model time for chemical accident responses. The results showed the adequate conditions for the modeling of chemical accidents for quick response and to increase the applicability of the model.

■ keyword : | Science Technology | Particle Dispersion Model | Pollutant Transport | Simulation Speed | Particle | Water Pollutant Accident | OpenMP | Arrangement Algorithm | Parallel Programming |

* 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었습니다(과제번호 20DPIW- C153746-02).

접수일자 : 2020년 10월 26일

심사완료일 : 2020년 11월 29일

수정일자 : 2020년 11월 17일

교신저자 : 이동섭, e-mail : dsree@kict.re.kr

I. 서론

자연하천에서 수질 오염사고 발생 시 오염물의 거동을 해석하기 위하여 수치모형을 적용할 수 있으며, 사고 대응방안 수립을 위한 프로그램으로 사용할 수 있다. 국내에서는 폐물 유출사고와 같은 수백억에 달하는 피해의 수질 오염사고가 발생한 바 있으며, 이 같은 수질 피해에 대한 방재대책 수립과 신속한 대응을 위하여 오염물질 거동 해석을 위한 예측 모델링이 중요하다.

그동안 오염물 거동 분석은 다양한 방식으로 이루어져 왔으며 그 중 상당수는 오염물을 주입하는 실험 및 모의로 구성되었다[1]. 일반적인 자연하천의 경우 수심이 하폭의 크기보다 적기 때문에 오염물이 투입되는 경우 연직 방향으로의 확산은 단기간에 완료되었다. 이러한 이유로 자연하천에서의 오염물 확산 예측 관련 연구는 2차원 모형으로 구현이 충분하다고 볼 수 있다[2]. 오염 확산 모형 중 이송-확산 모형의 경우, 2차원 혼합을 계산하기 위한 분산계수의 결정이 중요하나 시간과 비용이 소모되는 추적자 실험이 동반되어야 하므로 본 연구에서는 분산계수의 입력이 필요 없는 입자 분산 모형을 도입하고자 하였다[3].

이와 같은 2차원 오염물 확산 모형을 적용할 때 필요한 입력 자료로 하천의 흐름과 관련된 유속 및 수심과 하천의 지형자료 등이 있다. 이러한 하천 데이터를 취득하는 방법으로 기존의 연구자들은 유속계나 수위 측정소와 같은 직접 측정 방식을 주로 적용하였다. 최근에는 드론과 같은 무인이동체의 기술 발전과 보급으로 인하여 하천 모형 구동에 필요한 데이터를 원격으로 확보하는 새로운 방식이 발전하였다. 무인이동체에 연결할 수 있는 기존의 RGB 기반 카메라를 비롯하여, 소형화 기술이 발전되어 드론에 장착이 가능해진 초분광 카메라 등의 이미지 처리 장비를 이용하여 데이터를 수집하면서, 기존의 기기를 이용한 점 및 선 단위의 하천 측량과 데이터 수집에서 벗어나 면 단위의 데이터 수집이 가능하게 되었다.

무인이동체를 이용하는 하천 연구 방향이 발전하면서 최근에는 하천 내의 변화를 분석하기 위해 드론 운영 플랫폼에 하천 흐름-수질 예측모형을 탑재하여 드론 계측을 통한 정밀 입력 자료를 바탕으로 모형을 구동하

는 방식이 제시되고 있다. 원격 계측을 통한 입력자료 구축이 완성되면 하천 환경변화 시나리오에 따른 하천 흐름-수질 예측모형 적용을 할 수 있도록 모형이 탑재되고 이를 구동할 수 있게 된다. 이 같은 방식으로 연구가 진행되면 환경사고에 대응 가능한 하천 시공간 정보를 예측할 수 있게 되어 다양한 하천 환경 모니터링과 더불어 하천 오염사고 저감이 원활하게 이루어질 수 있을 것이다.

오염사고 대응에 사용될 수 있는 드론 운영 플랫폼 기기는 데스크탑 형태로 이동식 기지국에 설치되어 활용될 것이다. 이러한 종류의 데스크탑은 과거에는 계산 속도 향상을 위하여 CPU 클럭의 속도를 증가시키는 방향으로 발전이 이루어졌으나, 최근 들어 미세공정이 나노 단위로 이루어지면서 속도 개선에 한계점이 발생하여 그 대신 단일 CPU의 코어의 개수를 증가하는 방향으로 발전하게 되었다[4]. 하천 오염물의 이동을 예측하고 신속하게 대응하려면 계산속도의 개선이 필요하므로 이러한 CPU 코어의 발전 방향인 개수 증가를 활용하는 방안이 앞으로의 연구와 미래 기술 대응에 바람직할 것이다.

본 연구에서는 오염물 확산 해석이 가능한 입자 분산 모형을 개발 및 개선하여 단일 CPU의 멀티코어를 활용한 병렬 프로그래밍이 도입된 수질사고 대응을 위한 모의시간을 기준 소요시간 내로 구현하고자 하였다. 이를 위하여 개발된 입자분산모형의 입자 및 시간 단계(time step iteration)의 개수에 따른 계산시간을 분석하고, 다중 코어 활용에 따른 모의 속도의 개선점을 반영하여 모형의 활용성을 높이도록 하였다.

II. 모형 개요 및 이론

1. 입자분산 모형의 전단류 분산이론

오염물질의 혼합해석은 이송-분산 방정식을 이용한 모형 해석이 주로 이루어지나, 이러한 종류의 모형은 전단이송과 연직확산이 균형을 이루는 Taylor 구간에서만 적용 가능한 단점이 있다. 본 연구에서는 Taylor 구간 외에도 초기혼합구간에 적용이 가능한 입자 분산 모형을 개발하고 개선하여 적용성을 평가하고자 하였

다. 전단류 분산이론에 따라서 전단이송에 의한 수평 혼합과 연직 배열 알고리즘을 이용한 연직확산을 순차적으로 계산하여 개별 오염입자의 혼합거동을 해석하였다[5]. 기존에 주로 사용되는 이송-분산 모형의 경우 분산계수의 산정과 입력이 필요하였으나 현 모형은 개별 하천 현황에 맞는 분산계수의 입력이 요구되지 않아 신속한 오염사고 대응에 적합하다고 볼 수 있다.

2. 수평 및 연직 확산을 위한 모형 특성

본 모형의 수평 확산 단계에서는 종 방향은 로그분포 [6], 횡 방향은 직선분포 형태[7]로 확산하면서 전단류 및 난류확산[8]에 의한 입자의 이동을 계산한다. 연직 확산 단계에서는 전단에 의해 이송된 입자를 연직 알고리즘에 따라 입자의 연직 이송 계산으로 혼합을 모의한다. 이 방식은 Fischer (1979)가 제안한 순차계산모형을 기반으로 하며, 전단류에 의한 오염물질의 이송과 난류확산을 순차적으로 계산하여 분산계수 없이 전단류에 의한 오염물질의 혼합을 계산한다[9].

전단 이송 단계에서는 연직방향으로 균일하게 분포하는 농도가 전단류에 의해 연직방향 유속편차에 따라 종횡방향으로 분리하게 된다. 이후 연직혼합단계에서는 난류 및 분자확산과정에 의해 연직방향으로 오염물질이 균일하게 분포하게 되는 것이다. 이러한 과정을 반복하여 2차원 혼합으로 계산하는 알고리즘으로 구성된다.

이 과정에서 사용되는 공식들은 다음과 같다. 첫 번째로 종방향 연직 유속의 경우 다음 공식을 사용한다 [6].

$$u = \bar{u}_s \left[1 + \sqrt{g} / \kappa C_h (1 + \ln(z/h)) \right] \quad (1)$$

여기서 \bar{u}_s 는 종방향 수심평균 유속, C_h 는 Chezy 계수이며, κ 는 von Karman 상수, g 는 중력가속도, z 는 연직 좌표, h 는 수심이다. 개발된 입자분산모형은 2차원 흐름해석모형과 연계되어 있기 때문에 수심평균 유속을 사용한다[10]. 수심평균 유속으로부터 유속의 연직분포가 생성된 후 전단류를 이용한 입자의 종방향 이송을 통하여 [그림 1]의 가)와 같이 입자를 이동시킨다.

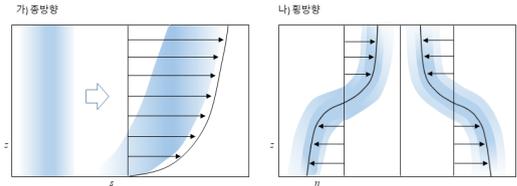


그림 1. 종횡방향 혼합 과정

다음으로 횡방향 공식은 아래와 같다[7].

$$v' = 2\bar{v}_s \left(\frac{z}{h} - \frac{1}{2} \right) \quad (2)$$

여기서 \bar{v}_s 는 횡방향 수심평균 유속이다. 횡방향 분포에 관한 연구는 Rozovskii(1957), de Vriend(1977)와 Kikkawa 등(1976)이 진행하였으며 본 연구에서는 적용이 용이하고 매개변수에 대한 민감도가 커 횡방향 혼합을 계산하는데 유리한 직선 형태의 Odgaard (1986) 공식을 사용하였다[7][11][12]. 이 식을 이용한 입자의 이동 형태는 [그림 1]의 나)와 같다. 이후 연직 혼합 과정은 이송과 난류 및 분자 확산으로 구성되어 있으며 [그림 2]와 같다.

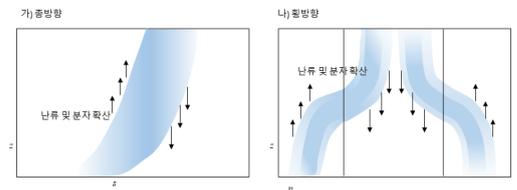


그림 2. 연직 혼합 과정

이송 단계에서는 다음과 같은 방정식을 사용한다.

$$x_i(t + \Delta t) = x_i(t) + u_i(x_i, t) \Delta t + R \sqrt{2\epsilon_i \Delta t} \quad (3)$$

위 식에서 $u_i(x_i, t) \Delta t$ 부분이 전단에 의한 이송이며, $R \sqrt{2\epsilon_i \Delta t}$ 부분이 난류 확산이다. 난류확산계수는 $\epsilon_i = 0.15hu^*$ 로 Fischer et al.(1979)의 공식을 따른다[9].

입자 확산 과정에서는 입자의 거동 뿐만 아니라 오염물 농도의 확인이 중요하므로 각 격자별 입자 개수 확

인 절차를 통하여 결과를 농도장으로 변화시킨다. 농도를 구하는 공식은 다음과 같다.

$$C(x,t) = \frac{mn(x,t)}{h\Delta x\Delta y} \quad (4)$$

여기서 Δx , Δy 는 계산격자의 크기이며, m 은 개별입자의 질량(kg), $n(x,t)$ 는 계산 격자 내 포함되어 있는 입자의 수이다. 계산 격자의 좌표 정보를 이용하여 개별 격자 내에 포함되어 있는 입자의 수를 계산할 수 있으며, 이 방식을 이용하면 입자의 위치를 농도장으로 계산하는 것이 가능하다.

3. 모형속도 개선 방안 및 멀티코어 이룬 적용

모형 속도의 개선은 다양한 연구들을 통해 진행되었으며, 그 중 주로 활용되는 방안으로는 CPU의 다중코어를 이용하거나 그래픽카드를 활용하는 방식 등이 있다. CPU 다중코어의 경우, 기존에 사용되는 수문 모형이나 수치해석 모형에 CPU 다중코어가 적용된 바 있다 [13]. 그 밖의 속도개선 방식으로는 컴파일러를 이용하여 최적화하는 방식이 있으며, 이 방식은 데이터 구조를 단순화하고 벡터화(vectorization)를 적용하여 속도를 향상시킨다[14][15]. 벡터화는 행렬 계산 시 단일 요소를 n번 처리하지 않고, 행렬의 여러 요소를 동시에 처리하도록 루프를 재작성하는 과정으로 이루어진다. 그래픽 카드를 사용하여 속도를 개선하는 방식의 경우, 카드 안의 그래픽 처리장치 GPU(Graphics Processing Unit)의 개별 코어를 활용하는 병렬 프로그래밍으로 속도를 증가시킬 수 있다[16][17]. 최근 딥러닝 관련 분야에서 사용되고 있으나 대표적으로 사용되는 GPU 속도 개선 방식인 CUDA가 NVIDIA 그래픽카드에만 적용되는 한계점이 있다.

이 중에서 CPU의 다중코어를 활용하는 경우 할당 메모리를 구분하여 여러 개의 계산을 동시에 수행할 수 있는 모델 구조를 가지고 있다. 이 중 공유 메모리 방식에서 표준화된 인터페이스로는 OpenMP가 있으며 이를 적용하기 위해서는 대상 플랫폼에서 구현이 가능하도록 변환 컴파일러가 필요하다. 현 모형에서는 OpenMP를 통한 병렬 프로그래밍을 구현하기 위하여

Visual Studio 기반의 Intel Fortran을 사용하여 코드를 구현하도록 하였다. Fortran을 이용한 병렬 프로그래밍 방식으로는 OpenMP 외에도 MPI를 사용하는 방법도 있으며[18], MPI가 워크스테이션과 같은 높은 개수의 병렬 코어에서는 확장성이 더 높으나 현 연구는 하나의 범용적인 데스크탑으로 적용되는 결과를 분석하기 위하여 단일 CPU의 다중 코어를 활용하여 상대적으로 병렬 프로그램 구성이 더 간단한 OpenMP 방식을 사용하기로 하였다[19]. OpenMP의 병렬 프로그래밍을 구체적으로 적용할 부분은 농도장 산정을 위한 각 격자별 입자 개수 확인 절차 부분이다. 이 부분은 입자의 개수에 따라서 계산량이 비례하는 것으로 추정되었기 때문에 우선 사전 모의를 통하여 계산 소요시간을 확인한 후 병렬 프로그래밍을 통하여 모의 필요 시간을 감소시키고자 하였다.

병렬 프로그래밍 방식 중 OpenMP 방식은 기본적으로 C/C++ 과 Fortran 언어로 구성된 지시문을 기반으로 하며, 병렬 프로그램을 비교적 쉽게 작성할 수 있도록 구성되어 있다. 주로 사용하는 병렬 프로그래밍 방식은 루프문인 C 언어의 For 문이나 Fortran 언어의 Do 문 직전에 OpenMP 지시문을 추가하여 병렬방식으로 계산할 부분을 지정하는 것이다[20]. 본 연구에서는 [그림 3]과 같은 방식으로 OpenMP 지시문을 사용하였다. 그 다음 작성한 코드를 컴파일러를 통해 실행 파일로 변환하여 PDM 모형의 병렬 프로그래밍을 구현하였다.

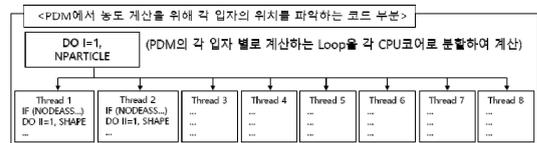


그림 3. OpenMP 병렬 프로그래밍 구성

현 코드에서는 OpenMP 지시문에서 지정한 문구 전에는 하나의 주 단위로 계산을 수행하다가 병렬처리 영역에 도달한 후 다수의 병렬 단위로 나누어 처리하도록 하여 모형의 속도를 개선하도록 하였으며, 컴파일러를 이용한 최적화 방식도 적용하여 서로 비교하도록 하였다.

III. 모형 속도 개선 적용

1. 입자 및 시간 단계 별 모형 속도 비교

본 연구에서는 우선 모의 시간에서 큰 비중을 차지하는 계산 절차를 검증하기 위하여 가상 사행 수로에 모의를 적용하여 입자 수에 따른 필요 모의 시간 결과 및 시간 단계 (Time step iteration) 별 필요 시간을 분석하였다. 가상 사행 수로의 격자 구성은 다음 그림과 같으며 약 4,200여개의 요소로 구성하였다.

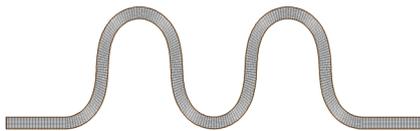


그림 4. 가상 사행수로 격자 구성

우선 입자 분산 모형 구동에 필요한 유속 및 수심을 산정하기 위하여 경계조건을 상류 유량 1.5 cms, 하류 수심 0.75 m 로 지정하여 흐름 모형을 구동하였다. 흐름 모형의 구동 결과 사행수로에서 변화하는 유속을 구현하였으며, 가상 사행수로인 관계로 사행으로 인한 침식 및 퇴적으로 발생하는 지형 변화에 따른 유속의 변화는 적게 반영되고 주로 가상수로의 중앙지점을 따라서 최대 0.6 m/s 정도의 유속이 발생하였다.

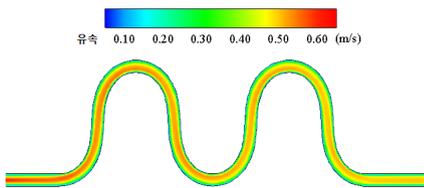


그림 5. 가상 사행수로 유속 모의 결과

이후 입자 분산 모의를 위해 우선 20,000개의 입자를 주입하였으며 연직방향 계층은 200개로 나누어 계산하였다. 위의 유속 결과를 이용하여 입자 분산 모형을 구동한 결과는 다음과 같다.

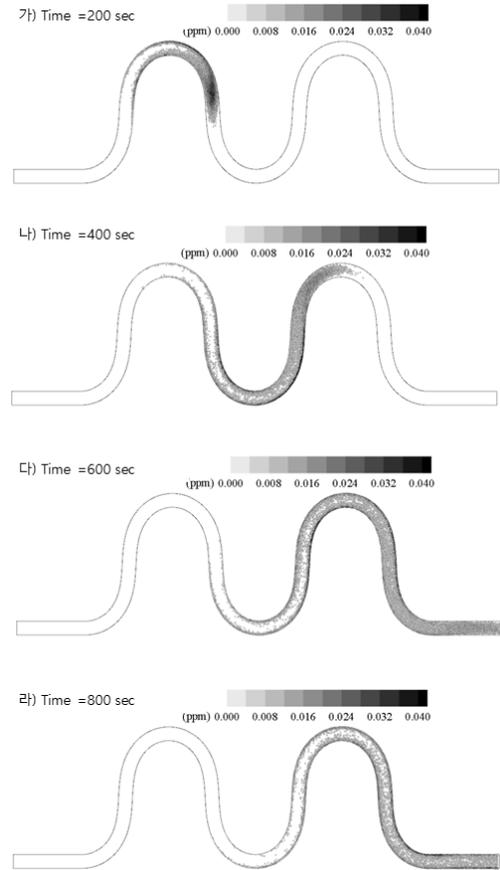


그림 6. 가상 사행수로 입자 분산 모의 결과

결론적으로 가상 사행수를 따라서 입자의 이동이 잘 구현되었으며, 실제 자연 하천의 오염물 확산 거동과 유사한 형태로 오염운이 형성되었다. 다음 그림과 같이 초반 입자의 거동을 확대해보면 최대 유속이 발생하는 지점을 따라서 주로 이동하면서, 상대적으로 느린 양안에서는 오염운 뒷부분의 꼬리 형태가 유속이 늘어지는 부분을 잘 표현했다고 볼 수 있다.

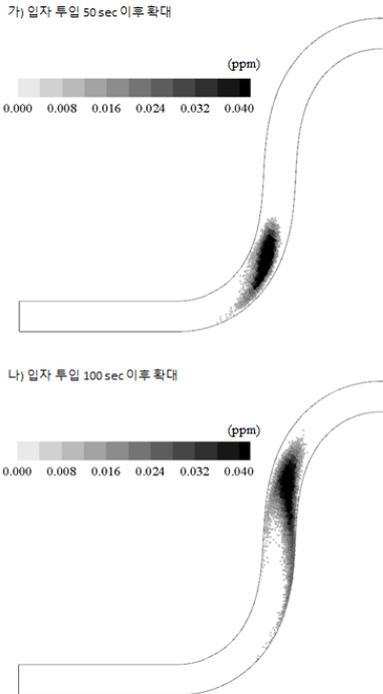


그림 7. 입자 거동 모의 결과

본 연구의 다음 단계로 입자 분산 모형의 속도 분석을 위하여 우선 입자의 개수가 모의 속도에 미치는 영향을 분석하고자 하였다. 입자의 개수를 5,000개에서 40,000개 까지 증가시키면서 모의하였다. 그리고 운영 플랫폼에 탑재되는 컴퓨터 성능에 따른 속도 비교도 중요하므로 CPU 2.5 GHz의 노트북과 CPU 4.6 GHz의 데스크탑과의 계산속도 비교도 수행하였다. 모의 결과 다음 그림과 같이 모의 하는 입자 개수와 모의 필요 시간이 비례함을 확인하였다.

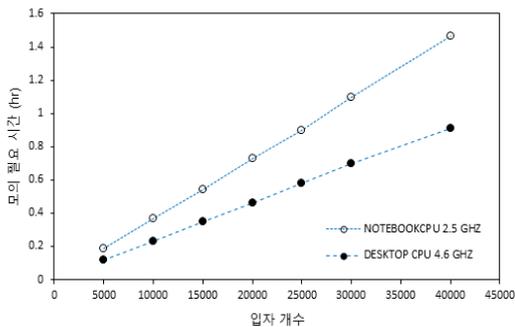


그림 8. 입자 개수 별 모의 필요 시간

다음 비교를 위하여 모의에 필요한 시간 단계 (Time step iteration) 의 개수를 조절한 결과, 모의 필요 시간이 정비례할 것으로 예상되었으나, 초기에는 필요 시간이 지수 형태와 유사하게 증가하다가 입자가 모의 영역을 벗어나는 지점인 time step 800 이상에서 필요 시간 증가폭이 감소함을 확인하였다.

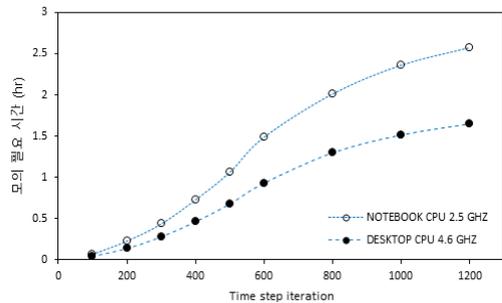


그림 9. 모의 Time step iteration 별 모의 필요 시간

이후 시간 단계가 1,200개로 증가함에 따라서 모의 필요시간 증가폭이 거의 없어지는 것을 볼 수 있는데, 다수의 입자가 모의 영역을 벗어난 상태가 되어 입자 추적과 농도 계산에 필요한 계산이 점차 줄어드는 것을 확인할 수 있었다. 그러므로 농도 계산을 위한 격자 별 입자의 개수 확인 단계가 현 모형에서 가장 계산 부하량이 높은 부분인 점이 증명되었으며, 병렬 프로그래밍의 주요 목표를 이 부분에 중점을 두고 모형의 코드를 구성하였다.

2. 속도 개선 방식 및 코어 개수 별 비교

본 연구의 병렬 프로그래밍에 사용되는 코어의 개수는 기본적인 1 코어 형태로부터 일반적인 데스크탑이 제공하는 물리 8 코어 구성까지 적용하였다. 입자 분산 모형에서 입자를 개별적으로 추적하면서 농도 계산을 위해 각 입자 별로 계산하는 Do-Loop문의 각 인자를 코어별로 할당하였다. 이로써 8 코어 병렬 프로그램의 경우 기본 1 코어에 입자 1, 9, 17.. 등의 계산을 할당하고, 기본 2 코어에는 입자 2, 10, 18.. 등의 계산을 할당하면서 이러한 과정을 반복하고 기존의 높은 계산 부하량을 각 코어별로 분할하여 계산하도록 하였다. 병렬 프로그래밍 코드 적용 예시는 다음과 같다.

```

nthread=2
call omp_set_num_threads(nthread)
!$omp parallel do default(shared) &
private(i,II,III,J,JJ,JJJ,WINDN,XR,XE,XEE,YE,YEE) schedule(static)
DO I=1, NPARTICLE
...
! Count particles in an element code area (code omitted)
!$omp end parallel do
    
```

그림 10. 병렬 프로그래밍 코드 예시

위 코드에서는 nthread = n의 부분으로 코어 개수 n개를 지정한 후 다음 명령어를 통하여 지정된 코어 개수를 할당하도록 하였다. 이후 !\$omp parallel do 명령어를 사용하여 OpenMP 지시문을 도입하면서 지시문 아래의 Do-Loop 문을 각 활용 코어별로 분할하여 계산 시키도록 하였다.

이러한 병렬 프로그래밍의 속도 개선 결과를 확인하기 위하여 우선 가상 사행수에서 입자 개수를 1,000개로 조절하여 모의를 수행하였다. 코어 활용에 따른 모의의 속도 향상 수준은 [그림 11]과 같다. 병렬 프로그래밍 적용이 1 코어에서 2 코어로 증가할 때 속도 개선이 가장 크다는 것을 확인할 수 있었으며, 늘어나는 코어 개수에 따라서 속도 개선이 계속 이루어짐과 동시에 그 효과가 점점 줄어드는 것이 나타났다. 이는 병렬 프로그래밍에서 자주 확인되는 현상으로, 코어의 개수에 따른 의존성이 점점 줄어들면서 나중에는 그 효과가 적은 것을 알 수 있다. 병렬 프로그래밍만 적용하는 경우 8 코어 까지 활용하여 약 1.6 배의 개선 효과를 나타내었다. 한편, 컴파일러 최적화에 의한 속도 개선도 수행하였으며, 이를 통해 단독으로 약 1.1 배 정도로 속도가 증가한 것을 확인하였다. 그리고 두 가지 기법을 동시에 적용한 경우에는 최대 1.7 배의 속도 개선 효과를 볼 수 있었다. 본 연구에서는 8 코어 구성까지 적용하여 일반적인 데스크탑 환경에서는 속도 개선이 계속된 것을 확인하였으나 이후 워크스테이션이나 서버와 같이 코어가 더 활용되는 장비의 속도 개선의 한계를 확인하기 위하여 코어에 따른 모의 필요 시간을 역다항식 형태로 회귀분석하고 외삽하여 [그림 11]과 같이 표기하였다. 결과적으로 약 16 코어에 도달할 때 약 1.8 배의 속도 향상이 이루어지면서 이후로는 속도 향상의 폭이 거의 없는 것을 예상할 수 있었다.

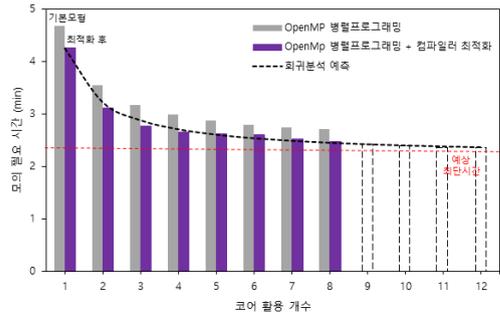


그림 11. 코어 활용 개수 및 최적화에 따른 모의 필요시간과 예측

멀티코어 적용에 따른 모형의 속도 개선 효과와 더불어 기존 소요시간 내로 모의가 가능한 최적 입자 및 코어 개수의 관계를 도출하기 위하여 모의를 진행하였으며 그 결과는 [그림 12]와 같다. 모의 필요 기준 소요시간을 10분으로 지정하는 경우 입자 수 10,000개인 경우, 필요 병렬 프로그래밍 코어 수가 6개 이상임을 확인할 수 있었다. 모의 필요 기준 소요시간을 20분으로 지정하는 경우 활용되는 입자 수 20,000개의 경우 코어 5개 이상, 16,000개의 경우 코어 2개 이상, 12,000개의 경우 코어 1개 이상이 활용되는 등의 관계를 확인할 수 있었다. 물론 현 모의는 가상 사행 수로에 적용되었으며, 입자가 수를 벗어나지 않는 시간 단계 기준 800개를 바탕으로 모의하였기 때문에 [그림 9]에서 확인한 것처럼 일반적인 수로 내에서 오염 확산이 지속되고 시간 단계가 계속 늘어나면 모의 필요 시간이 기하급수적으로 소요될 것이다. 그러나 일단 본 연구의 결과로 일반적인 확산 모의에 필요한 소요 시간을 확인할 수 있었으며, 실제 모의하는 자연 하천의 경우 각 모의 단계별 시간이 현 모의의 1 초보다 큰 단위를 주로 사용하고 오염 사고 대응을 위한 현 모형의 순간 모의 특성상 매우 높은 시간 단계가 필요하지 않기 때문에 본 시간소요 기준 결과가 이후 모형 운영에 있어서 중요한 참조 기준이 될 수 있을 것으로 판단된다.

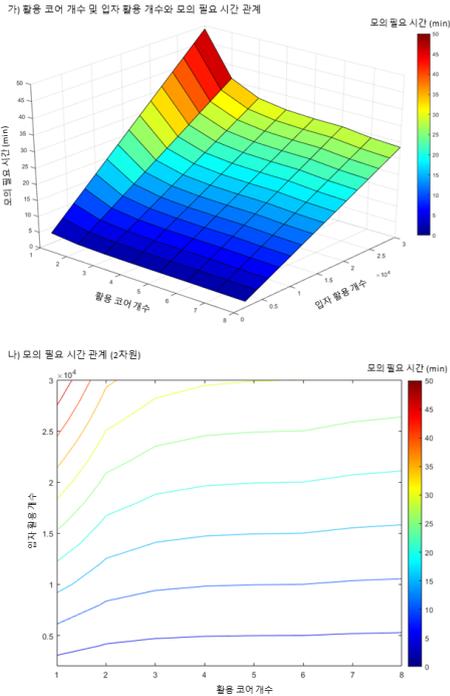


그림 12. 코어 개수 및 입자 별 모의 필요 시간 관계

IV. 결론

본 연구에서는 하천 오염 사고에 신속하게 대응하기 위하여 입자 분산 모형의 속도를 분석하고 개선하여 모형 활용성을 평가하였다. 입자 분산 모형은 전단류 분산이론을 따르면서 수평 혼합 과정은 전단이송, 연직 혼합 과정은 연직배열 알고리즘을 이용한 난류 및 입자 확산을 구현하였다. 하천에서 발생할 수 있는 오염사고에 신속하게 대응하기 위해 모형 속도 개선을 목표로 단일 CPU의 멀티코어를 활용하는 병렬 프로그래밍을 도입하고 컴파일러 최적화를 수행하고자 하였다.

연구 결과, 가상 사행 수로에 모의를 적용하여 입자수에 따른 모의 시간 결과를 분석하고 입자 개수가 모의 시간과 비례하는 것을 확인하였다. 그리고 농도 계산을 위한 과정인 격자 별 입자의 개수 확인 과정이 모의 시간에 큰 비중을 차지하는 것을 확인하였으며, 이를 토대로 병렬 프로그래밍을 이 부분에 중점을 두고 진행하였다. 병렬 프로그래밍에 사용되는 코어 개수에

따라서 계산 속도의 개선을 구현할 수 있었으며 제일 효과가 높은 구간은 1 코어에서 2 코어로 증가하는 구간임을 확인할 수 있었다. 최적화와 병렬 기법의 속도 개선 효과를 비교하고 이를 분석하여 16 코어까지 속도 향상을 예상하였다. 그리고 추가적인 분석으로 사행 가상 수로에서 기준 소요시간 내로 모의가 가능한 최적 입자 및 코어 활용의 관계를 도출할 수 있었다. 기존에 사용되는 수문 모형이나 수치해석 모형에 다중코어를 적용한 것이 아니라 전단류 및 난류확산에 의한 입자 이동을 반영하는 단계적 혼합 과정으로 구동되는 입자 분산 모형을 병렬 및 최적화하게 되어 수질 오염사고에 신속하게 대응하기 위한 적절한 모의 조건을 구성할 수 있게 되었다. 이러한 속도 개선 및 분석을 바탕으로 모형을 개선하고 이와 함께 멀티코어가 향상된 컴퓨터를 사용하면 신속한 오염사고 대응 및 피해 저감 결정 지원의 향상을 기대할 수 있을 것이다.

참고 문헌

- [1] I. W. Seo, H. J. Choi, Y. D. Kim, and E. J. Han, "Analysis of Two-dimensional Mixing in Natural Streams Based on Transient Tracer Tests," *Journal of Hydraulic Engineering*, Vol.142, No.8, p.04016020, 2016.
- [2] M. E. Lee and I. W. Seo, "Analysis of Pollutant Transport in the Han River with Tidal Current Using a 2D Finite Element Model," *Journal of Hydro-Environment Research*, Vol.1, No.1, pp.30-42, 2007.
- [3] I. Park and I. W. Seo, "Modeling Non-Fickian Pollutant Mixing in Open Channel Flows Using Two-dimensional Particle Dispersion Model," *Advances in Water Resources*, Vol.111, pp.105-120, 2018.
- [4] C. G. Kim and J. G. Kim, "Optimizing Image Processing on Multi-core CPUs with Intel Parallel Programming Technologies," *Multimedia Tools and Applications*, Vol.68, No.2, pp.237-251, 2014.
- [5] 박인환, 김형석, 성호제, *수질오염사고 분석 정확도 향*

- 상을 위한 입자분산모형 개발, KICT 2018-116, 한국건설기술연구원 주요사업 보고서, 2018.
- [6] I. L. Rozovskiĭ, *Flow of water in bends of open channels*, Academy of Sciences of the Ukrainian SSR, 1957.
- [7] A. J. Odgaard, "Meander Flow Model. I: Development," *Journal of Hydraulic Engineering*, Vol.112, No.12, pp.1117-1135, 1986.
- [8] G. E. P. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *The Annals of Mathematical Statistics*, Vol.29, pp.610-611, 1958.
- [9] H. B. Fischer, J. E. List, R. C. Y. Koh, J. Imberger, and N. H. Brooks, *Mixing in Inland and Coastal Waters*, Academic Press, 1979.
- [10] C. G. Song, I. W. Seo, and Y. D. Kim, "Analysis of Secondary Current Effect in the Modelling of Shallow Water Flow in Open Channels," *Advances in Water Resources*, Vol.41, pp.29-48, 2012.
- [11] H. J. de Vriend, "A Mathematical Model of Steady Flow in Curved Shallow Channels," *Journal of Hydraulic Research*, Vol.15, No.1, pp.37-54, 1977.
- [12] H. Kikkawa, S. Ikeda, and A. Kitagawa, "Flow and Bed Topography in Curved Open Channels," *Journal of the Hydraulics Division*, Vol.102, No.9, 1976.
- [13] 김수영, 윤광석, "분포형 유역유출모형의 홍수예보시스템 적용을 위한 최적해상도 결정에 관한 연구-GRM 모형을 활용하여 금호강 유역을 중심으로," 한국수자원학회논문집, 제52권, 제2호, pp.107-113, 2019.
- [14] N. G. Dickson, K. Karimi, and F. Hamze, "Importance of Explicit Vectorization for CPU and GPU Software Performance," *Journal of Computational Physics*, Vol.230, No.13, pp.5383-5398, 2011.
- [15] R. Allen and K. Kennedy, *Optimizing Compilers for Modern Architectures: a Dependence-based Approach*, Taylor & Francis US, 2002.
- [16] J. M. Domínguez, A. J. Crespo, and M. Gómez-Gesteira, "Optimization strategies for CPU and GPU implementations of a Smoothed Particle Hydrodynamics Method," *Computer Physics Communications*, Vol.184, No.3, pp.617-627, 2013.
- [17] P. V. Le, P. Kumar, A. J. Valocchi, and H. V. Dang, "GPU-based High-performance Computing for Integrated Surface-sub-surface Flow Modeling," *Environmental Modelling & Software*, Vol.73, pp.1-13, 2015.
- [18] S. J. Noh, J. H. Lee, S. Lee, K. Kawaike, and D. J. Seo, "Hyper-resolution 1D-2D Urban Flood Modelling Using LiDAR Data and Hybrid Parallelization," *Environmental Modelling & Software*, Vol.103, pp.131-145, 2018.
- [19] J. C. Neal, T. J. Fewtrell, P. D. Bates, and N. G. Wright, "A Comparison of Three Parallelisation Methods for 2D Flood Inundation Models," *Environmental Modelling & Software*, Vol.25, No.4, pp.398-411, 2010.
- [20] S. Zhang, Z. Xia, R. Yuan, and X. Jiang, "Parallel Computation of a Dam-break Flow Model Using OpenMP on a Multi-core Computer," *Journal of Hydrology*, Vol.512, pp.126-133, 2014.

저 자 소 개

신 재 현(Jaehyun Shin)

정희원



- 2018년 2월 : 서울대학교 건설환경공학부(공학박사)
- 2018년 3월 ~ 2020년 5월 : 서울대학교 공학연구원 박사후연구원
- 2020년 6월 ~ 현재 : 한국건설기술연구원 국토보전연구본부 박사후연구원

〈관심분야〉 : 수리학, 토목, 수자원

성 호 제(Hoje Seong)

정회원



- 2015년 2월 : 홍익대학교 토목공학
학과(공학석사)
- 2015년 2월 ~ 현재 : 한국건설기
술연구원 국토보전연구본부 전임연
구원

〈관심분야〉 : 수리학, 토목, 수자원

박 인 환(Inhwan Park)

정회원



- 2017년 2월 : 서울대학교 건설환
경공학부(공학박사)
- 2017년 3월 ~ 2017년 6월 : 서울
대학교 공학연구원 박사후연구원
- 2017년 7월 ~ 2019년 8월 : 한국
건설기술연구원 국토보전연구본부
수석연구원

- 2019년 9월 ~ 현재 : 서울과학기술대학교 교수

〈관심분야〉 : 수리학, 토목, 수자원

이 동 섭(Dong Sop Rhee)

정회원



- 2000년 2월 : 서울대학교 토목공
학과(공학석사)
- 2000년 1월 ~ 2002년 12월 : ㈜
금호엔지니어링 대리
- 2002년 12월 ~ 현재 : 한국건설
기술연구원 국토보전연구본부 연구
위원

- 2015년 2월 : 서울대학교 건설환경공학부(공학박사)

〈관심분야〉 : 토목공학, 수공학