

미시영역에서 중간역역까지 적용 가능한 범용 분자 시뮬레이션 시스템의 개발

오 광 진[†]

요 약

본 논문에서는 개발된 범용 분자 시뮬레이션 시스템에 대해 기술하고자 한다. 본 연구에서 개발된 분자 시뮬레이션 시스템의 가장 큰 장점은 다른 무엇보다도 Langevin dynamics simulation이나 dissipative particle dynamics simulation 기법을 도입하여 all-atom 모델뿐만 아니라 coarse-grain 모델까지도 다룰 수 있도록 설계하였고 따라서 미시영역은 물론 중간역역에서 일어나는 현상까지도 시뮬레이션할 수 있도록 설계하였다는 점이다. 이를 통해 하나의 통합된 분자 시뮬레이션 시스템으로 생체막 내에서 마취제의 분포, 단백질 접힘 현상, 마이크로 채널 내에서 생체고분자의 구조와 유동 특성 등과 같이 미시영역에서부터 중간역역에 이르는 다양한 현상을 연구할 수 있게 되었다. 개발된 시스템을 이용하면 molecular dynamics simulation에 기반한 분자 시뮬레이션 시스템으로는 불가능한 여러 중요한 바이오/나노 시스템을 시뮬레이션할 수 있을 것으로 기대한다. 마지막으로 벤치마크 결과를 통해 개발된 분자 시뮬레이션 시스템의 성능을 측정하였고 성능 최적화를 위한 병목지점을 조사하였다.

키워드 : 분자 시뮬레이션

Development of a general purpose molecular simulation system from microscopic to mesoscopic scales

Kwang Jin Oh[†]

ABSTRACT

In this paper, a general purpose molecular simulation system, which has been developed by the author, are described. One of the most advantageous features is that the molecular simulation system can handle a coarse-grained model as well as an all-atom model. Therefore, we can simulate mesoscopic phenomena as well as microscopic phenomena with the help of Langevin dynamics simulation and dissipative particle dynamics simulation techniques. Thus we could study anesthesia, protein folding, biopolymer flow in microchannel with single framework, which spans from microscopic to mesoscopic scales. We expect that we can also simulate many other bio/nano systems of technological importance which are not feasible by means of molecular dynamics simulation technique. Finally, performance data are shown and a bottleneck is identified for future optimization.

Key Words : Molecular Simulation

1. 서 론

분자 시뮬레이션은 생체분자의 시뮬레이션에서부터 재료의 시뮬레이션에 이르기까지 자연과학이나 공학 등 여러 분야에 걸쳐 광범위하게 활용되고 있다. 그 범위는 화학, 물리학, 재료공학, 화학공학, 생명공학, 기계공학 등을 망라한다. 지금까지 분자 시뮬레이션의 대상이 되어 왔던 분자들을 나열해 보면 생체분자, 고분자, 유기화합물, 무기화합물 등 다양하다. 이와 같이 분자 시뮬레이션이 광범위하게 활용되고 있는 데에는 몇 가지 이유가 있다. 우선 분자 시뮬레이션은

실험적으로는 얻기 힘든 분자 수준에서의 미시적인 정보들을 제공하여 준다. 따라서 실험으로 설명하기 힘든 현상들을 이해하고 설명하는데 많은 도움을 준다. 그리고 실험이 불가능한 초고온이나 초고압 상태와 같은 극한 상황에서 일어나는 현상에 대해서도 자세한 정보를 제공해 준다. 분자 시뮬레이션에 사용되는 force field가 더욱더 정확해 지고 분자 시뮬레이션 기법이 발전함에 따라 결과에 대한 신빙성도 대폭 증대될 것으로 기대하고 있다. 한편, 분자 시뮬레이션은 주어진 상호작용에 대해 정확한 해를 제공한다. 따라서 분자 시뮬레이션은 제안된 근사이론에 대한 검증도구로서도 사용된다.

분자 시뮬레이션이란 시뮬레이션하고자 하는 대상, 다시

[†] 정 회 원 : 한국과학기술정보연구원 슈퍼컴퓨팅센터 선임연구원
논문접수 : 2005년 7월 20일, 심사완료 : 2005년 12월 6일

말해서, 모델 분자들을 가상의 모의실험 상자 안에 넣은 후 그 모델 분자들의 상태, 다시 말해서, 위치와 속도를 진화시켜 나가는 것을 말한다. 따라서 분자 시뮬레이션을 통해 우리는 분자들의 궤적을 얻게 된다. 여기서 분자는 분자를 구성하는 모든 원자를 포함하도록 표현되기도 하고 또는 연결된 몇 개의 원자를 하나의 가상 입자로 생각하여 표현되기도 한다. 전자의 경우를 all-atom 모델이라 하고 후자를 coarse-grained 모델이라고 한다. 주지하다시피 coarse-grained 모델은 더 적은 수의 입자를 포함하기 때문에 계산 시간이나 메모리 측면에서 상당한 장점을 갖고 있다. 하지만 일부 자유도가 생략되어 있기 때문에 정확도에 있어서는 all-atom 모델에 비해 떨어질 수 있다.

한편 분자들의 상태를 시간에 대해 진화시키는 방법에는 molecular dynamics simulation (MD)[1], Langevin dynamics simulation (LD)[1], dissipative particle dynamics simulation (DPD)[2] 등이 있다. 여기서 MD는 주로 all-atom 모델에 사용되고 LD와 DPD는 coarse-grained 모델에 사용된다. 그 이유는 다음과 같다. 앞서서도 언급하였듯이 coarse-grained 모델에서는 일부 생략된 자유도가 있기 때문에 생략된 부분에 대한 고려를 하여야 하는데 MD는 그러한 방법을 제공하지 못한다. 반면, LD와 DPD는 자체가 생략된 자유도를 간접적으로 고려할 수 있도록 고안된 방법이다.

MD 방법이 갖는 한계는 시간적인 한계와 공간적인 한계에 기인한다. 우선 시간적인 한계에 대해 논하고자 한다. 앞서 MD는 일정한 시간 간격을 단위로 하여 그 시간 단위만큼씩 순차적으로 상태를 진화시켜 나간다. 그 시간 단위는 대상이 되는 분자모델의 특성에 따라 정해진다. 예를 들면, all-atom 모델의 경우 빠르게 일어나는 진동운동이 피코초(10^{-12} 초) 수준에서 일어나기 때문에 시간단위로 보통 1 펨토초(10^{-15} 초)를 사용한다. 만약 시간 단위를 1000 펨토초를 사용하게 되면 분자들의 궤적이 발산하게 될 것이다. 반면, coarse-grained 모델의 경우에는 보통 피코초 이상의 시간 단위를 사용한다. 따라서, 마이크로초(10^{-6} 초) 수준에서 일어나는 현상을 시뮬레이션하기 위해서는 all-atom 모델의 경우 10^9 번을 진화시켜야 하나 coarse-grained 모델의 경우에는 10^6 번 진화를 시키면 충분하다. 보통 MD에서 10^6 번 진화를 시킨다는 사실과 중간영역에서 일어나는 현상이 대부분 마이크로초 수준이라는 것을 감안하면 현실적으로 MD는 적당한 방법이 아니라는 것을 알 수 있다. 다음으로 공간적인 한계에 대해 논하고자 한다. 예를 들면, 랩온어칩(lab-on-a-chip)의 설계에 중요한 마이크로 채널 내에서의 생체 고분자의 유동해석을 한다고 가정하자. 만약 all-atom 모델과 MD를 사용한다면 모의실험 상자 안에는 10^9 수준에 이르는 많은 수의 입자들이 포함된다. 그런데 MD의 경우 보통 10^4 의 입자 수를 취급한다는 것을 주지할 필요가 있다. 따라서 마이크로미터 수준에서 일어나는 현상을 MD를 사용하여 시뮬레이션하기에는 그 규모가 너무 크다고 할 수 있다. 정리하면, MD는 미시영역(microscopic scale)에서 일어나는 현상을 시뮬레이션하기 위한 방법이고 LD와 DPD는

중간영역(mesosopic scale)에서 일어나는 현상을 시뮬레이션하기 위한 방법이라고 할 수 있겠다.

지금까지 여러 분자 시뮬레이션 시스템들이 개발되어 왔다. 대표적인 것으로 CHARMM[3], AMBER[4], NAMD[5] 등이 있다. 하지만 이들은 주로 MD를 이용한 방법만을 제공하고 있다. 따라서 중간영역에서 일어나는 현상을 시뮬레이션하기에는 부족하다. 주지하다시피 중간영역에서 일어나는 현상 중에 기술적으로 중요한 것들이 많이 있다. 예를 들면, 고분자의 유연화적인 거동이나 마이크로 채널 내에서 생체고분자의 거동 등이 있다. 이에 본 연구에서는 미시영역에서부터 중간영역에 이르기까지 넓은 영역에 걸쳐 적용될 수 있는 범용 분자 시뮬레이션 시스템을 구현하고자 한다. 다음 절들에서는 본 연구에서 구현된 분자 시뮬레이션 시스템에 대해 설명하고 그리고 마지막으로 구현된 분자 시뮬레이션 시스템을 이용한 벤치마크 결과와 사례들을 보여주고자 한다.

2. 분자 시뮬레이션 시스템의 구조

어떤 임의의 입자 i 에 대한 운동방정식은

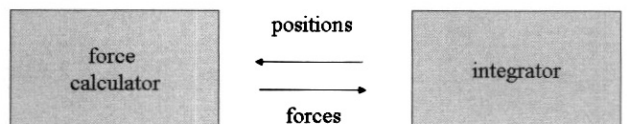
$$\vec{f}_i = m_i \frac{d^2 \vec{r}_i}{dt^2} \tag{1}$$

으로 주어진다. 여기서 m_i 는 입자의 질량이고 \vec{r}_i 는 입자의 위치이며 \vec{f}_i 는 입자에 작용하는 힘이다. 그런데 힘은

$$\vec{f}_i = -\frac{\partial V}{\partial \vec{r}_i} \tag{2}$$

와 같이 퍼텐셜에너지를 통해서 계산된다. 분자 시뮬레이션은 식(2)를 통해 힘을 계산하고 그 힘을 이용하여 식(1)을 수치 적분함으로써 입자들의 움직임을 시뮬레이션하는 방법이다.[1]

분자 시뮬레이션 시스템은 크게 두 개의 모듈로 분리할 수 있다. 하나는 힘을 계산하는 모듈 (힘계산기, force calculator)이고 다른 하나는 운동방정식을 수치 적분하는 모듈(수치적분기, integrator)이다. (그림 1)은 힘계산기와 수치적분기의 관계를 보여준다. 수치적분기는 힘계산기에 입자들의 위치를 제공하고 힘계산기는 그 위치를 이용하여 힘을 계산하고 그 힘을 다시 수치적분기에 전달한다. 여기서 수치적분기는 다시 새롭게 얻어진 힘을 이용하여 입자들의 궤적을 갱신한다. 분자 시뮬레이션은 이 과정의 반복이라고 할 수 있다.



(그림 1) 프로그램의 구조

2.1 힘계산기

계의 퍼텐셜에너지는 일반적으로 다음과 같이 주어진다.

$$V = \sum_i V_1(\vec{r}_i) + \sum_i \sum_j V_2(\vec{r}_i, \vec{r}_j) + \sum_i \sum_j \sum_k V_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) + \dots \tag{3}$$

여기서 첫 번째 항은 일체 퍼텐셜(one-body potential)의 합이고 두 번째 항은 이체 퍼텐셜(two-body potential)의 합 등등이다. 식으로부터 알 수 있듯이 퍼텐셜에너지는 입자들의 위치의 함수이다. 그런데 식(1)은 다시

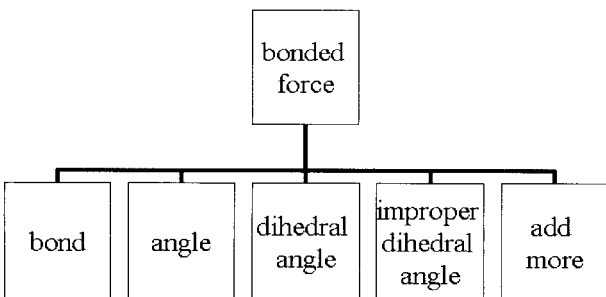
$$V = V_{bonded} + V_{non-bonded} \tag{4}$$

로 나타낼 수 있다. V_{bonded} 는 분자 내에 존재하는 결합길이, 결합각, 이면각 등의 변화에 기인하여 발생하는 퍼텐셜에너지로서 결합력(bonded force)을 야기한다. 결합력은 분자의 골격을 유지시켜 주는 역할을 한다. 다시 말해서, 결합력은 분자의 골격이 평형상태에서의 골격과 벗어나면 원래상태로 복귀시켜 주는 역할을 한다. $V_{non-bonded}$ 는 V_{bonded} 에 관련되지 않은 원자들 사이의 상호작용 즉, van der Waals 상호작용이나 정전기적 상호작용으로 인해 발생하는 퍼텐셜에너지로서 비결합력(non-bonded force)을 야기한다. 이것을 바탕으로 해서 본 연구에서는 힘계산기를 결합력 계산기와 비결합력 계산기로 나누어 구성하였다.

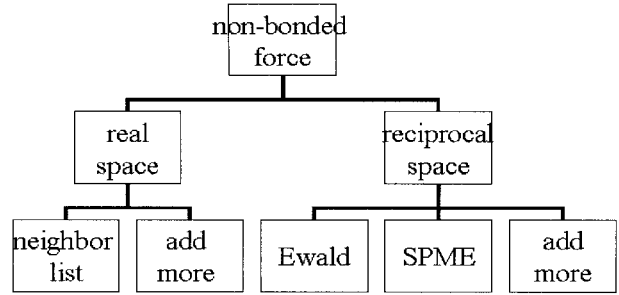
(그림 2)에서와 같이 결합력 계산기에는 결합길이의 변화, 결합각의 변화 이면각의 변화에 기인한 상호작용만이 고려되었다. 물론, 결합력을 야기하는 상호작용에는 앞에서 언급한 상호작용 외에도 여러 가지가 있을 수 있다. 예를 들면, 결합길이의 변화와 결합각 변화 사이의 coupling에 기인한 상호작용이 있다.[6, 7] 하지만 만약 새로운 형태의 상호작용이 추가될 경우에는 해당되는 서브루틴을 첨가하면 된다.

한편, 비결합력 계산기는 real space에서의 비결합력 계산기와 reciprocal space에서의 비결합력 계산기로 구성되어 있다 (그림 3 참조). 이것은 long-range 상호작용을 계산하는 방법이 비결합력을 real space와 reciprocal space로 나누어 계산하기 때문이다.

분자 시뮬레이션은 유한한 크기의 시뮬레이션 상자 안에 유한한 숫자만큼의 입자들만을 다룬다. 만약 상호작용의 범위가 시뮬레이션 상자의 길이보다도 작다면 아무런 문제가



(그림 2) 힘계산기 내 결합력 계산기의 구조



(그림 3) 힘계산기 내 비결합력 계산기의 구조

발생하지 않는다. 하지만 정전기적 상호작용과 같은 long-range 상호작용의 경우에는 보통 시뮬레이션 상자 길이보다 먼 거리만큼 작용하기 때문에 오차가 생기게 된다. 이러한 어려움을 극복하기 위해 개발된 방법이 Ewald summation 방법이다.[1] Ewald summation 방법에서는 상호작용을 real space와 reciprocal space로 나누어 계산한다.

reciprocal space에서의 비결합력은 앞서서도 언급하였듯이 Ewald summation 방법을 사용하면 된다. 그런데 Ewald summation 방법은 최적화된다 할지라도 계산시간이 $N^{3/2}$ 에 비례하는 것으로 알려져 있다.[8] 여기서 N은 입자의 수이다. 이에 본 연구에서는 Ewald summation 방법보다는 fast Fourier 변환(FFT)을 이용한 smooth particle mesh Ewald (SPME) 방법을[9] 구현하였다. SPME 방법은 계산시간이 $N \log N$ 에 비례하는 것으로 알려져 있다. 따라서 SPME 방법은 Ewald summation 방법보다도 훨씬 더 효율적이다.

real space 비결합력 계산은 reciprocal space 비결합력 계산보다도 힘계산기의 성능에 더 큰 비중을 차지한다. 왜냐하면 그 계산시간이 N^2 에 비례하기 때문이다. 따라서 컴퓨터의 성능이 과거에 비해 크게 향상되었다 하더라도 입자들의 수가 증가하게 되면 원하는 결과를 얻기에는 현실적으로 어려움이 따른다. 따라서 본 연구에서는 그 계산시간을 단축하기 위한 효과적인 방법들을 채택하였다. 열거하면 potential truncation 방법[1], Verlet neighbor list 방법[10], cell-linked list 방법[11], 그리고 그 둘을 혼합한 방법[12]이 그것이다.

1) potential truncation 방법

potential truncation 방법에서는 모든 가능한 입자 쌍에 대해 힘을 계산한다기보다는 어느 일정거리 r_{cut} 내에 존재하는 입자 쌍에 대해서만 힘을 계산한다. 다시 말해서 r_{cut} 이상에서는 입자들이 상호작용하지 않는다고 가정한다. 따라서 potential truncation 방법을 사용하면 입자들의 밀도를 ρ 라고 했을때 평균적으로 $4\pi r_{cut}^3 \rho N/3$ 의 입자 쌍에 대해서만 힘을 계산하면 된다. 물론 입자가 상호작용을 하는지 아닌지, 다시 말해서, 입자 쌍이 r_{cut} 내에 존재하는지 여부를 결정하기 위한 검색은 모든 $N(N-1)/2$ 의 입자 쌍에 대해서 거리를 계산하여 행해야 한다.

2) Verlet neighbor list 방법

Verlet neighbor list 방법에서는 (그림 4)에서 보여주는

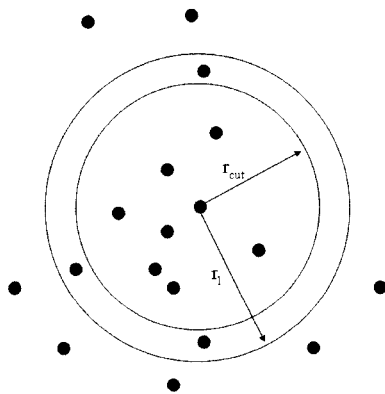
바와 같이 각 입자에 대해 r_{cut} 보다 크게 설정된 거리 r_i 내에 있는 모든 입자들의 목록을 저장해 두었다가 나중에 힘을 계산할 때 그 목록을 사용한다. 물론 목록을 만들 때는 총 입자 쌍의 수인 $N(N-1)/2$ 만큼 검색을 하여야 한다. 하지만 매번 목록을 만들 필요가 없다. 목록의 갱신 여부는 매번 검사하게 되는데 다음의 조건을 만족하면 목록을 갱신하고 그렇지 않으면 저장되어 있는 목록을 그냥 사용한다.

$$2\sqrt{3r_{max}^2} > r_i - r_{cut} \quad (5)$$

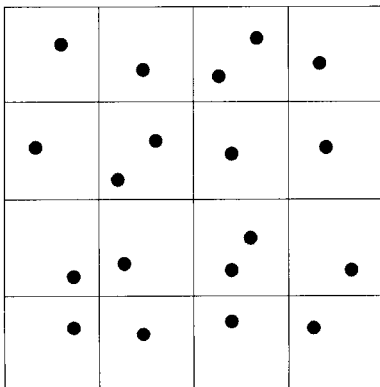
여기서 r_{max} 는 가장 최근 목록이 갱신되었을 때 입자의 위치와 현재 입자의 위치로부터 각 성분(x, y, 및 z 방향)의 차를 구하고 그 중에서 가장 차이가 큰 성분의 차를 선택한다. 보통 목록의 갱신은 매 10-20 스텝마다 이루어진다. 목록을 사용하면 평균적으로 $4\pi r_{cut}^3 \rho N/3$ 의 입자 쌍에 대해서만 검색을 하면 되기 때문에 결과적으로 많은 계산 시간을 단축할 수 있게 된다.

3) cell-linked list 방법

cell-linked list 방법은 (그림 5)에서와 같이 시뮬레이션 상자를 작은 셀들로 나누어 입자들을 해당되는 셀에 저장한다. 다시 말해서 셀의 일련번호를 알게 되면 그 셀 내에 있는 모든 입자들의 일련번호를 알 수 있도록 한다. 보통 셀



(그림 4) Verlet neighbor list 방법



(그림 5) cell-linked list 방법

의 크기는 r_{cut} 에 가깝게 설정하기 때문에 검색을 할 때 어느 특정 셀 주위에 있는 26개의 셀에 있는 입자들과의 원자 쌍만 고려하면 된다. 셀에 입자들을 저장하는 과정은 N에 비례하고 검색시간은 $N_c(N_c-1)/2+26N_c^2$ 에 비례한다. 여기서 N_c 는 셀 내에 있는 평균적인 입자의 수로 V_c 를 셀의 부피라고 했을 때 ρV_c 로 주어진다. $N_c(N_c-1)/2$ 는 같은 셀 내에 있는 입자 쌍의 수이고 $26N_c^2$ 은 주위에 있는 26개의 셀에 있는 입자들과의 입자 쌍 수이다. 지금까지 설명한 모든 과정은 매 스텝마다 이루어 진다.

4) Verlet neighbor list 방법과 cell-linked list 방법을 혼합한 방법

이 방법은 Verlet neighbor list 방법과 근본적으로 같다. 오직 다른 점은 목록을 만들 때 cell linked list 방법을 사용한다는 점이다. 따라서 목록을 만드는 과정은 $N+N_c(N_c-1)/2+26N_c^2$ 하에 비례하게 된다. 앞서 설명한 방법들을 정리하면 다음과 같다.

<표 1>에서는 앞에서 설명한 방법들에 대한 이론적인 계산시간을 비교 정리하였다.

<표 1> neighbor list 방법들의 계산시간 비교

	목록작성	검색	힘계산
none	-	$N(N-1)/2$	$N(N-1)/2$
trun	-	$N(N-1)/2$	$4\pi r_{cut}^3 \rho N/3$
verlet	$N(N-1)/2$	$4\pi r_{cut}^3 \rho N/3$	$4\pi r_{cut}^3 \rho N/3$
cell	N	$N_c(N_c-1)/2+26N_c^2$	$4\pi r_{cut}^3 \rho N/3$
verlet_cell	$N+N_c(N_c-1)/2+26N_c^2$	$4\pi r_{cut}^3 \rho N/3$	$4\pi r_{cut}^3 \rho N/3$

none : 아무런 기법이 적용되지 않은 경우
 trun : potential truncation 방법을 적용한 경우
 verlet : Verlet neighbor list 방법을 사용한 경우
 cell : cell-linked list 방법을 사용한 경우
 verlet_cell : Verlet neighbor list 방법과 cell-linked list 방법을 혼합한 경우

2.2 수치적분기

통계역학의 관점에서 보면, 분자 시뮬레이션으로 얻어진 입자들의 궤적은, 다시 말해서, 위치와 속도는 어느 특정한 모집단(앙상블)으로부터 추출된 것이라고 생각할 수 있다.[1] 그런데 분자 시뮬레이션으로부터 얻어진 궤적은 계의 총 에너지가 보존되기 때문에 보통의 경우에는 NVE 앙상블에서 추출된 것이라고 볼 수 있다. 물론 NVE 앙상블과는 다른 앙상블에서 궤적을 추출한다 해도 얻어지는 평균값에는 변화가 없다. 하지만 때때로 온도가 일정하게 유지되는 앙상블(NVT 앙상블)이나 압력과 온도가 일정하게 유지되는 앙상블(NPT 앙상블)에서 궤적을 추출하는 것이 편리하거나 효율적일 때가 있다. 예를 들면, 생체막(biomembrane)을 시뮬레이션한다든지[13] 고체상태에서 구조적인 상변이(phase transition)를 시뮬레이션 한다든지[14, 15] 하는 상황에서는 NPT 앙상블이 더 효율적이다. 현재 NVT 앙상블이나 NPT 앙상블에서 분자 시뮬레이션을 할 수 있게 하는 많은 방법

들이 개발되어 있다.[1] 본 연구에서는 그 중에서도 범용 분자 시물레이션 시스템의 개발에 적당한 extended system 방법을[16-19] 채택하였다.

〈표 2〉 운동방정식에 사용된 운동주체 별 운동변수

운동주체	운동변수
입자	\vec{r}_i (질량중심 위치)
	\vec{v}_i (질량중심 속도)
	\vec{r}'_{ia} (질량중심으로부터 입자의 위치)
	\vec{v}'_{ia} (질량중심으로부터 입자의 속도)
시물레이션 상자	\vec{h} (9개의 성분을 갖는 시물레이션 상자 행렬)
	\vec{v}_g (시물레이션 상자의 속도)
온도계	η_i (입자와 연결된 온도계의 위치)
	v_{η_i} (입자와 연결된 온도계의 속도)
	ξ_i (시물레이션 상자와 연결된 온도계의 위치)
	v_{ξ_i} (시물레이션 상자와 연결된 온도계의 속도)

분자 시물레이션에서 운동의 주체는 입자, 시물레이션 상자, 온도계(thermostat)라고 할 수 있다. 시물레이션 상자와 온도계의 운동을 허용하거나 금지함으로써 우리는 NVE 앙상블, NVT 앙상블, 또는 NPT 앙상블에서 분자 시물레이션을 수행할 수 있다. 예를 들면, 입자의 운동만 허용한다면 NVE 앙상블이 모집단의 형태가 되고 입자와 온도계의 운동을 허용한다면 NVT 앙상블이 모집단의 형태가 된다. 만약 입자, 시물레이션 상자, 온도계의 운동이 모두 허용된다면 NPT 앙상블이 모집단의 형태가 된다. 각 운동의 주체에 대한 운동변수는 <표 2>에 나타나 있으며 각 주체별 운동방정식은 다음과 같이 주어진다.

$$\frac{d\vec{r}'_{ia}}{dt} = \vec{v}'_{ia} \quad (6)$$

$$\frac{d\vec{v}'_{ia}}{dt} = \frac{\vec{f}_{ia}}{m_{ia}} - \frac{\vec{f}_i}{m_i} - v_{\eta_i} \vec{v}'_{ia} \quad (7)$$

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i + \vec{v}_g \vec{r}_i \quad (8)$$

$$\frac{d\vec{v}_i}{dt} = \frac{\vec{f}_i}{m_i} - \vec{v}_g \vec{v}_i - \frac{1}{N_w} Tr[\vec{v}_g] \vec{v}_i - v_{\eta_i} \vec{v}_i \quad (9)$$

여기서 특이할 만한 사항은 <표 2>에 나타난 바와 같이 입자들의 운동을 그룹(atomic group)에 기반으로 하여 기술한다는 점이다.[19] 참고로 그룹은 서로 연결된 입자들의 집합체를 말한다. 좀 더 구체적으로 서술하면 제약조건(cons-

traint)을 서로 공유하는 입자들의 집합체를 말한다. 위 방정식에서 N_w 는 그룹들의 총 운동자유도이다. 마지막으로 Tr는 행렬의 대각성분의 합을 나타낸다. 한편 시물레이션 상자에 대한 운동방정식은 다음과 같이 주어진다.

$$\frac{d\vec{h}}{dt} = \vec{v}_g \vec{h} \quad (10)$$

$$\frac{d\vec{v}_g}{dt} = \vec{G}_g - v_{\xi_i} \vec{v}_g \quad (11)$$

여기서 \vec{G}_g 는 다음과 같이 주어진다.

$$\vec{G}_g = \frac{V(\vec{P}_{int} - \vec{I}P_{ext}) + \vec{P}_K}{W_g} \quad (12)$$

이고

$$\vec{P}_K = \frac{2K_w}{N_w} \vec{I} \quad (13)$$

이다. V 는 시물레이션 상자의 부피이고 \vec{P}_{int} 는 내부 압력이고 P_{ext} 는 외부에서 가하는 압력이다. \vec{I} 는 단위 행렬이고 W_g 는 시물레이션 상자의 질량이다. 위의 식들로부터 알 수 있는 바와 같이, 시물레이션 상자가 변하는 것은 외부에서 가한 압력에 기인한다. 다시 말해서, 시물레이션 상자는 외부에서 가한 압력과 시물레이션 상자 내부의 압력이 같아지도록 변화한다. 한편 입자들과 연결된 온도계의 운동방정식은 다음과 같이 주어진다.

$$\frac{d\eta_i}{dt} = v_{\eta_i} \quad (14)$$

$$\frac{dv_{\eta_i}}{dt} = G_{\eta_i} - v_{\eta_i} v_{\eta_i} \quad (15)$$

$$\frac{dv_{\eta_j}}{dt} = G_{\eta_j} - v_{\eta_j+1} v_{\eta_j} \quad (16)$$

$$\frac{dv_{\eta_M}}{dt} = G_{\eta_M} \quad (17)$$

이고 여기서

$$G_{\eta_i} = \frac{2K_f - N_f k_B T}{Q_{\eta_i}} \quad (18)$$

$$G_{\eta_j} = \frac{Q_{\eta_{j-1}} v_{\eta_{j-1}}^2 - k_B T}{Q_{\eta_j}} \quad (19)$$

$$G_{\eta_M} = \frac{Q_{\eta_{M-1}} v_{\eta_{M-1}}^2 - k_B T}{Q_{\eta_M}} \quad (20)$$

이다. N_f 는 입자의 총 운동자유도이고 K_f 는 입자들의 운동 에너지 합이며 Q_{η_i} 는 온도계의 질량이다. 마지막으로 k_B 는 볼츠만 상수이고 T 는 온도이다. 시뮬레이션 상자와 연결된 온도계의 운동방정식도 다음과 같이 비슷한 형태로 주어진다.

$$\frac{d\xi_i}{dt} = v_{\xi_i} \quad (21)$$

$$\frac{dv_{\xi_i}}{dt} = G_{\xi_i} - v_{\xi_i} v_{\xi_i} \quad (22)$$

$$\frac{dv_{\xi_j}}{dt} = G_{\xi_j} - v_{\xi_j+1} v_{\xi_j} \quad (23)$$

$$\frac{dv_{\xi_M}}{dt} = G_{\xi_M} \quad (24)$$

이다. 여기서

$$G_{\xi_i} = \frac{2K_f^{box} - N_f^{box} k_B T}{Q_{\xi_i}} \quad (25)$$

$$G_{\xi_j} = \frac{Q_{\xi_{j-1}} v_{\xi_{j-1}}^2 - k_B T}{Q_{\xi_j}} \quad (26)$$

$$G_{\xi_M} = \frac{Q_{\xi_{M-1}} v_{\xi_{M-1}}^2 - k_B T}{Q_{\xi_M}} \quad (27)$$

이고 N_f^{box} 는 시뮬레이션 상자의 운동자유도이다. 그리고 K_f^{box} 는 시뮬레이션 상자의 운동에너지이며 Q_{ξ_i} 는 온도계의 질량이다. 참고로 본 연구에서 온도계와 관련하여 사용한 방법은 Nose-Hoover chain 방법이다. Nose-Hoover chain에는 여러 개의 온도계가 연결될 수 있다. 앞의 운동방정식에서는 Nose-Hoover chain에 M개의 온도계가 연결되었다.

보통 분자 시뮬레이션에서 빠른 진동운동은 수치적분기의 효율성과 안정성이라는 측면에서 문제가 되어 왔다. 진동운동은 빠르게 진행되기 때문에 아주 짧은 간격을 두고 수치적분을 행해야 한다. 따라서 똑같은 현상을 재현하고자 하면 더 오랜 시간이 걸리게 된다. 반대로 상대적으로 큰 간격을 두고 수치적분을 하게 되면 궤적이 발산하게 된다. 따라서 상대적으로 큰 시간 간격을 두고 수치적분을 하면서 안정적인 궤적을 얻을 수 있는 방법을 모색하게 되었다. 그중 하나가 SHAKE/RATTLE 방법이다.[20, 21] SHAKE/RATTLE 방법은 현재 분자 시뮬레이션에서 널리 이용되고 있다. 그런데 SHAKE/RATTLE 방법은 NPT 앙상블에서

사용될 경우 변형된 버전이 사용되어야 하고 시간이 더 걸릴 수 있는 단점을 갖고 있다.[17] 뿐만 아니라 SHAKE/RATTLE 방법은 병렬프로그램에서 입자를 기반으로 하여 분배되었을 경우 프로세서 간에 불필요한 통신을 야기하게 된다.[22, 23] 한편, 그룹의 형태로 입자들의 운동방정식을 기술하게 되면 이러한 모든 어려움을 극복할 수 있게 된다.[19]

Langevin dynamics simulation에서 사용되는 운동방정식은 앞서 설명한 힘에 입자의 운동량에 비례하는 damping 항과 그리고 stochastic 항을 더함으로써 다음과 같이 쉽게 얻어진다.

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i \quad (28)$$

$$\frac{d\vec{v}_i}{dt} = \frac{\vec{f}_i}{m_i} - \gamma \vec{v}_i + \vec{R}_i \quad (29)$$

여기서 γ 는 마찰계수이고 \vec{R}_i 는 다음의 조건을 만족하는 난수벡터이다.

$$\langle R_{i\alpha}(t) \rangle = 0 \quad (30)$$

$$\langle R_{i\alpha}(t) R_{j\beta}(t') \rangle = 2k_B T \gamma \delta_{ij} \delta_{\alpha\beta} \delta(t-t') \quad (31)$$

dissipative particle dynamics simulation에서 사용되는 운동방정식도 비슷하게 주어진다.

$$\frac{d\vec{r}_i}{dt} = \vec{v}_i \quad (32)$$

$$\frac{d\vec{v}_i}{dt} = \frac{\vec{f}_i}{m_i} \quad (33)$$

여기서

$$\vec{f}_i = \sum_{j \neq i} \vec{F}_{ij}^C + \vec{F}_{ij}^D + \vec{F}_{ij}^R \quad (34)$$

로 주어지는데 각각은 다음과 같다. 우선 분산력으로서

$$\vec{F}_{ij}^D = -\gamma \omega^D(r_{ij})(\hat{r}_{ij} \cdot \vec{v}_{ij}) \hat{r}_{ij} \quad (35)$$

으로 주어지고 여기서 $r_{ij} = |\vec{r}_i - \vec{r}_j|$, $\hat{r}_{ij} = (\vec{r}_i - \vec{r}_j)/r_{ij}$, $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ 이며 γ 는 마찰계수이다. 다음으로 무작위하게 작용하는 힘으로서

$$\vec{F}_{ij}^R = \sigma \zeta_{ij} \omega^R(r_{ij}) \hat{r}_{ij} \quad (36)$$

으로 주어지는데 여기서 σ 는 요동의 크기를 나타내는 상수

이고 ζ_{ij} 는 균일한 분포로부터 추출되는 난수이다. 마지막으로 보존력으로서

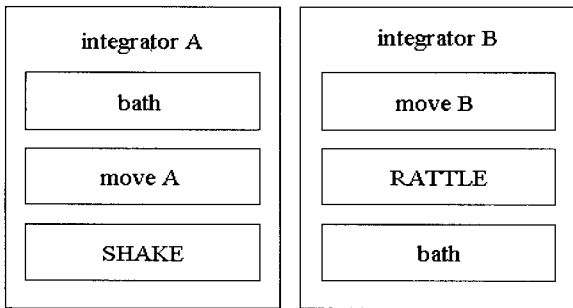
$$\vec{F}_{ij}^C = a\omega^C(r_{ij})\hat{r}_{ij} \quad (37)$$

으로 주어지는데 여기서 a 는 상수이고 다음과 같은 관계식을 갖는다.

$$\omega^R(r_{ij}) = \omega^C(r_{ij}) = 1 - \frac{r_{ij}}{r_c} \quad (38)$$

여기서 $r_{ij} < r_c$ 인 경우에만 값을 갖고 나머지 경우에는 영이다.

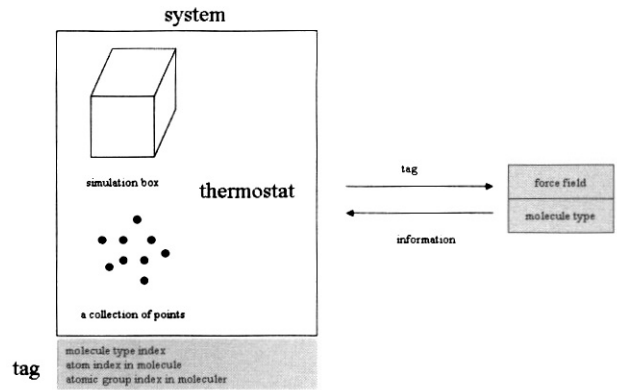
(그림 6)는 수치적분기의 구조를 나타낸다. 수치적분기는 두개의 부분으로 구성되는데 하나(수치적분기 A)는 힘계산기 이전에 사용되는 것이고 다른 하나(수치적분기 B)는 힘계산기 이후에 사용되는 것이다. 각 수치적분기의 구성은 (그림 6)에 나타나 있고 각 구성성분이 참조가 되는 순서에 따라 나타나 있다. 예를 들면, 수치적분기 A의 경우에는 압력이나 온도 (환경변수, bath)에 기인해서 입자들의 속도가 변하는 부분이 먼저 실행되고 나서 NVE 앙상블에서 사용되는 일반적인 수치적분기 (move A)가 실행된다. 그런 다음 SHAKE 부분이 실행된다. 하지만 수치적분기 B의 경우에는 NVE 앙상블 수치적분기(move B)가 먼저 실행되고 RATTLE 부분이 실행된다. 그리고 마지막으로 환경변수에 기인한 수치적분기 부분이 실행된다.



(그림 6) 수치적분기의 구조

3. 자료 구조

(그림 7)에서 보이는 바와 같이 계(system)의 구성성분은 입자, 시뮬레이션 상자, 온도계이다. 각각의 운동을 기술하기 위한 운동변수는 <표 2>에 제시되어 있고 배열의 형태로 메모리에 저장된다. 그런데 힘계산기나 수치적분기에 필요한 질량이나 상호작용과 관련된 상수 등 분자와 상호작용에 대한 정보들은 각 입자들에 할당된 tag를 통해 추출된다. tag는 시뮬레이션 상자 안에 있는 분자의 색인(molecule type index), 분자 내 원자의 색인(atom index in molecule), 분자 내 그룹의 색인(atomic group index in molecule)을 포



(그림 7) 자료구조

함한다. 분자에 대한 정보는 원자의 이름, 종류, 질량, 전하, 그리고 원자가 포함된 결합, 각, 이면각 등에 관련된 원자들의 번호를 포함하고 상호작용에 대한 정보는 상호작용의 종류와 관련된 상수들을 포함한다. 이 모든 정보는 시뮬레이션이 수행되기 전에 입력파일을 읽어 들임으로써 일괄적으로 저장된다.

4. 벤치마크 결과

힘계산기는 분자 시뮬레이션에서 계산시간 상 절대적인 비중을 차지한다. 따라서 본 절에서는 힘계산기와 관련된 벤치마크 결과를 보여주고자 한다. 소개된 벤치마크 결과는 모두 KISTI 슈퍼컴퓨팅센터 하멜 클러스터에서 얻어졌다. 하멜 클러스터는 256개의 IBM x355 서버로 구성되어 있고 프로세서는 Intel Pentium IV Xeon DP 2.8GHz로 이루어져 있으며 Myrinet2000 Network으로 연결되어 있다.

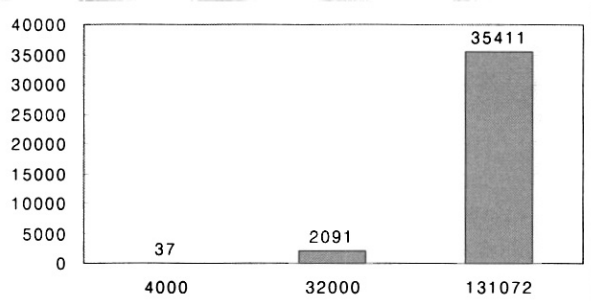
<표 3>은 한 스텝 당 힘계산기의 계산시간을 보여준다. 사용된 분자는 물속에 있는 단백질(PDB ID: 5DHFR)이고 시뮬레이션 상자 안에 있는 원자의 총수는 23558이다. 단백질과 물을 위해 사용된 force field는 각각 CHARMM22[24]와 TIP3P[25]이다. Verlet neighbor list와 cell-linked list 방법을 혼합한 방법을 사용하였고 r_{cut} 은 9.0Å이며 $n_1 = 10.5Å$ 이다. 정전기적 상호작용은 SPME 방법을 사용하였는데 이때 사용된 FFT로는 FFTW[26]를 이용하였다. 여기서 mesh의 dimension은 64×64×64이다. <표 3>이 보여주는 바와 같이 real space에서의 비결합력 계산시간은 2.57초로서 가장 오래 걸렸으며 결합력 계산시간은 0.03초로서 가장 짧게 걸렸다. 한편 reciprocal space 비결합력 계산시간은 0.3초로서 real space 비결합력 계산시간에 비해 약 10배정도 빠르게 나왔다.

<표 3> 힘계산기의 계산시간 (초)

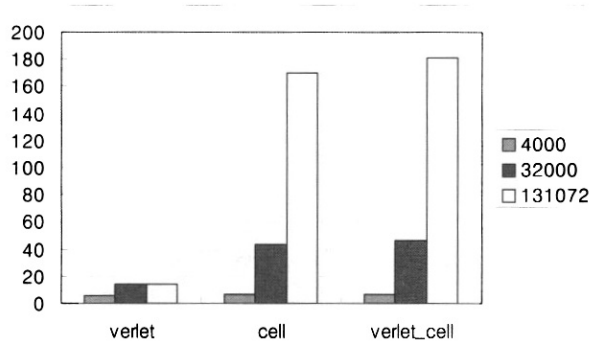
	결합력	real space 비결합력	reciprocal space 비결합력
계산시간 (초)	0.03	2.57	0.3

<표 4> real space에서의 비결합력 계산시간 (초)

	N=4000	N=32000	N=131072
trun	37	2091	35411
verlet	7	143	2544
cell	6	48	209
verlet_cell	6	45	195



(그림 8) potential truncation 방법에 의한 입자수 변화에 따른 real space에서의 비결합력 계산시간의 변화



(그림 9) potential truncation 방법과 비교한 neighbor list 방법들의 계산시간 향상률

다음으로 real space 비결합력 계산에 사용된 neighbor list 방법들의 계산시간을 비교하기 위해 reciprocal space 비결합력이 존재하지 않는 아르곤 분자를 선택하였다. 사용된 입자의 수는 각각 N=4000, N=32000, N=131072이다. <표 4>는 100스텝 당 real space에서의 비결합력 계산시간을 보여준다.

(그림 8)은 potential truncation 방법을 사용했을 경우 입자수 변화에 대한 real space에서의 비결합력 계산시간의 변화를 보여준다. N=4000인 경우 37초가 걸렸으나 N=131072인 경우 35411초로 약 1000배가 느려졌다. 입자의 수가 증가함에 따라 그 계산시간이 기하급수적으로 늘어남을 쉽게 알 수 있다.

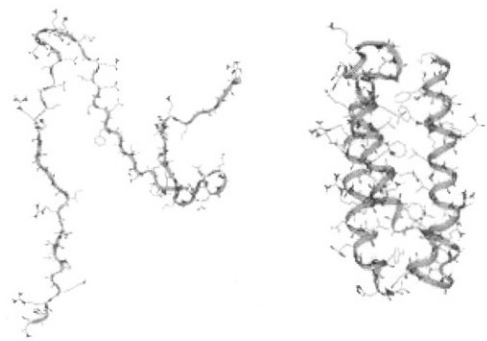
(그림 9)는 potential truncation 방법을 사용하였을 경우와 비교하여 Verlet neighbor list 방법, cell-linked list 방법, 둘을 혼합한 방법을 사용했을 경우 얻어지는 계산시간 향상률을 보여준다. 입자의 수가 131072인 경우, Verlet neighbor

list 방법과 cell-linked list 방법을 혼합한 방법을 사용하면 180배에 이르는 성능 향상을 보이고 있음을 알 수 있다. 그리고 입자의 수가 증가할수록 그 효과가 더욱더 크게 나타남을 알 수 있다. 그리고 Verlet neighbor list 방법, cell-linked list 방법, 그 둘의 방법을 혼합한 방법의 순서로 효율적임을 알 수 있다.

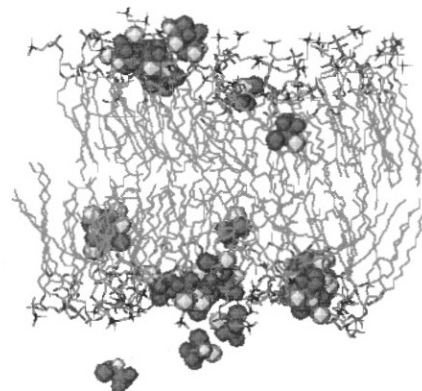
하지만 각 방향으로 셀의 숫자가 3보다 크지 않을 경우에는 cell-linked list를 사용함에 따른 효과가 적어지게 되므로 Verlet neighbor list 방법을 사용하는 것이 낫을 수도 있다.

5. 사례

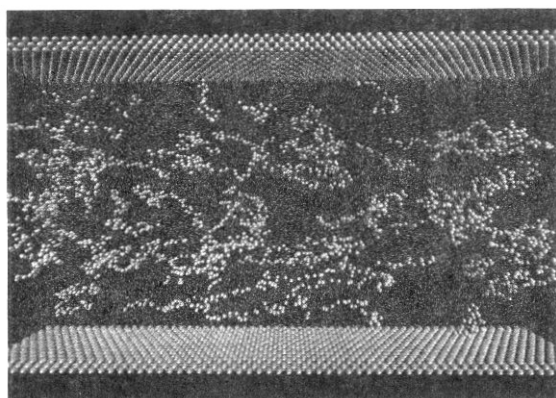
본 절에서는 개발된 시스템을 이용하여 연구한 몇몇 사례들로부터 얻은 결과를 보여주고자 한다. 특히, 미시영역에서의 현상에서부터 중간영역에서의 현상까지를 고르게 포함하도록 사례들을 보여주고자 하였다. 그리고 본 시스템에서 제공하고 있는 다양한 기능들을 사용한 사례들을 보여주고자 하였다. (그림 10)은 Korea@Home[27] 상에서 implicit solvent 모델인 SASA 모델과 LD를 이용하여 단백질의 접힘 현상을 재현한 결과를 보여주는 그림이다. 왼쪽은 단백질이 접히지 않은 임의의 초기구조를 보여주고 오른쪽 그림은 일정 시간이 흐른 후 제대로 접힌 단백질의 구조를 보여준다. (그림 11)은 NPT양상틀에서 분자 시뮬레이션을 이용



(그림 10) 단백질 접힘



(그림 11) 생체막 내에서 마취제의 분포



(그림 12) 마이크로채널 내에서 생체고분자 및 유체의 포아제 흐름

한 결과를 보여 주는 것으로서 생체막 내에서 마취제가 어떻게 분포하고 있는지를 보여준다. (그림 12)는 coarse-grained 모델과 DPD를 이용하여 마이크로채널 내에서 생체고분자 및 유체의 포아제 흐름을 시뮬레이션한 결과를 보여준다.

6. 결론 및 향후 계획

본 논문에서는 개발된 범용 분자 시뮬레이션 시스템에 대하여 기술하였다. 본 연구에서 개발된 분자 시뮬레이션 시스템의 가장 큰 장점은 다른 무엇보다도 all-atom 모델뿐만 아니라 coarse-grain 모델까지도 쉽게 다룰 수 있도록 설계되었고 미시영역에서부터 중간영역에 이르기까지 다양한 영역에서 일어나는 현상까지도 시뮬레이션할 수 있도록 설계되었다는 점이다. 앞으로는 개발된 분자 시뮬레이션 시스템의 성능 향상을 위한 최적화를 계획하고 있다. 본 논문에서 제시한 두가지 경우에 대해 행한 벤치마크 결과를 통해 real space에서의 비결합력 계산시간이 분자 시뮬레이션의 성능을 좌우하는 중요한 요소임을 알 수 있었다. 따라서 시스템의 성능 최적화 시 real space에서의 비결합력 계산 루틴의 최적화가 무엇보다도 선행되어야 할 것으로 예상된다.

참고 문헌

- [1] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*, Clarendon, Oxford (1987).
- [2] P.J. Hoogerbrugge and J.M.V.A. Koelman, "Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics", *Europhysics.Lett.* **19**, 155-160 (1992).
- [3] CHARMM (<http://www.charmm.org>).
- [4] The Amber Molecular Dynamics Package (<http://amber.scripps.edu>).
- [5] NAMD-Scalable Molecular Dynamics (<http://www.ks.uiuc.edu/Research/namd/>).
- [6] A. T. Hagler, E. Huler, and S. Lifson, "Energy Functions for Peptides and Proteins: I. Derivation of a Consistent Force Field Including the Hydrogen Bond from Amide Crystals", *J. Am. Chem. Soc.* **96**, 5319-5327 (1974).
- [7] A. T. Hagler and S. Lifson, "Energy Functions for Peptides and Proteins: II. The Amide Hydrogen Bond and Calculation of Amide Crystal Properties", *J. Am. Chem. Soc.* **96**, 5327-5335 (1977).
- [8] N. Karasawa and W. A. Goddard, "Acceleration of convergence for lattice sums", *J. Phys. Chem.* **93**, 7320-7327 (1989).
- [9] U. Essmann, L. Perera, M.L. Berkowitz, T. Darden, H. Lee, and L.G. Pedersen, "A smooth particle mesh Ewald method", *J. Chem. Phys.* **103**, 8577-8593 (1995).
- [10] L. Verlet, "Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules", *Phys. Rev.* **159**, 98-103 (1967).
- [11] R. W. Hockney and J. W. Eastwood, *Computer simulations using particles*, McGraw-Hill, New York (1981).
- [12] D. J. Auerbach, W. Paul, C. Lutz, A. F. Bakker, W. E. Rudge, and F. F. Abraham, "A special purpose parallel computer for molecular dynamics: motivation, design, implementation, and application", *J. Phys. Chem.* **91**, 4881-4890 (1987).
- [13] K. Tu, M. Tarek, M. L. Klein, and D. Scharf, "Effects of Anesthetics on the Structure of a Phospholipid Bilayer: Molecular Dynamics Investigation of Halothane in the Hydrated Liquid Crystal Phase of Dipalmitoylphosphatidylcholine", *Biophys. J.* **75**, 2123-2134 (1998).
- [14] M. Parrinello and A. Rahman, "Crystal structure and pair potentials: A molecular-dynamics study", *Phys. Rev. Lett.* **45**, 1196-1199 (1980).
- [15] M. Parrinello and A. Rahman, "Polymorphic transitions in single crystals: A new molecular dynamics method", *J. Appl. Phys.* **52**, 7182-7190 (1981).
- [16] M. E. Tuckerman and G. J. Martyna, "Understanding modern molecular dynamics: Techniques and application", *J. Phys. Chem B*, **104**, 159-178 (2000).
- [17] Glen J. Martyna, Mark E. Tuckerman, Douglas J. Tobias, and Michael L. Klein, "Explicit reversible integrators for extended systems dynamics", *Mol. Phys.* **87**, 1117-1157 (1996).
- [18] Massimo Marchi and Piero Procacci, "Coordinates scaling and multiple time step algorithms for simulation of solvated proteins in the NPT ensemble", *J. Chem. Phys.* **109**, 5194-5202 (1998).
- [19] K. J. Oh and M. L. Klein, "A parallel molecular dynamics simulation for a molecular system with bond constraints in NPT ensemble", accepted for publication in *Comp. Phys. Comm.*.
- [20] J.-P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, "Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes",

- J. Comp. Chem. **23**, 327-341 (1977).
- [21] H.C. Andersen, "RATTLE: A velocity version of the SHAKE algorithm for molecular dynamics calculations", J. Comp. Phys. **52**, 24-34 (1983).
- [22] D. Brown, J.H.R. Clarke, M. Okuda, and T. Yamazaki, "A domain decomposition parallel processing algorithm for molecular dynamics simulations of polymers", Comp. Phys. Commun. **83**, 1-13 (1994).
- [23] D. Brown, H. Minoux, B. Maigret, "A domain decomposition parallel processing algorithm for molecular dynamics simulations of systems of arbitrary connectivity", Comp. Phys. Commun. **103**, 170-186 (1997).
- [24] MacKerell et. al., "All-atom empirical potential for molecular modeling and dynamics studies of protein", J. Phys. Chem. B, **102**, 3586-3616 (1998).
- [25] W.L. Jorgensen, J. Chandrasekhar; J.D. Madura, R.W.

Impey, M.L. Klein, "Comparison of simple potential functions for simulating liquid water", J. Chem. Phys. **79**, 926-935 (1983).

[26] FFTW Home Page (<http://www.fftw.org>).

[27] Korea@Home (<http://www.koreaathome.org/>).



오 광 진

e-mail : koh@kisti.re.kr

1992년 고려대학교 화학과(학사)

1996년 고려대학교 대학원 화학과(석사)

2000년 네브라스카 주립대학교 화학과
(박사)

2000년~2003년 펜실베이니아 대학교 연구원

2003년~현재 한국과학기술정보연구원 슈퍼컴퓨팅센터 선임연구원

관심분야 : 슈퍼컴퓨팅, 분자시뮬레이션