

グラフィックカードを用いた 水表面張力の高速分子動力学シミュレーション

坂 牧 隆 司^{†1} 成 見 哲^{†1} 泰 岡 顕 治^{†2}

古典分子動力学シミュレーションは界面現象を定量的に観察できる手法である。しかし高い精度の界面系の分子動力学シミュレーションに膨大な計算を必要とする。本研究では、まず高速な分子動力学シミュレーションを行うために、近年発達したグラフィックカードを用いた計算環境を構築した。テスト計算の結果、グラフィックカードを用いて単精度で力の結果を行っても計算精度に問題がなかった。また通常のPCに比べて、 $\mathcal{O}[N^2]$ のアルゴリズムで約70倍、 $\mathcal{O}[N \log N]$ の高速なアルゴリズムで13.4倍の高速化が確認できた。本計算環境を用い水気液界面の計算を大規模な系で行い、精度の高い界面張力の値を得た。結果、SPC/Eモデルは実験値よりも10 mN/m程度低い値を示すことが確認された。

Fast Molecular Dynamics Simulation of Water Surface Tension Using Graphic Processing Unit

RYUJI SAKAMAKI,^{†1} TETSU NARUMI^{†1}
and KENJI YASUOKA^{†2}

Classical Molecular Dynamics (MD) simulation is a powerful method to investigate interfacial phenomena. However, huge amount of computation is required to perform precise simulation on the interface. In this paper, we used a Graphic Processing Unit (GPU), for accelerating MD simulations. The single precision calculation of forces with the GPU showed sufficient accuracy. The acceleration ratio between a PC and our GPU code is about 70 with $\mathcal{O}[N^2]$ algorithm and 13.4 with $\mathcal{O}[N \log N]$ algorithm. We performed large scale simulation of water interface with GPUs, and got with high accuracy its interfacial tension, which is about 10 mN/m lower than experimental result.

1. はじめに

1.1 界面の分子動力学シミュレーション

物理化学の分野において、物質の構造を予測することは大きな目標である。たとえば気体の物性は分子間の相互作用がきわめて小さいため、理想気体のように簡単に扱うことができる。また固体は分子間で強い相関があるが、周期的な構造を持っている点が理論構築の足がかりとなり、様々な物性の予測を可能にしている。だが液体は、分子間に強い相互作用があるうえに構造が時々刻々変化しているので、分子論的立場から定量的に考察することが困難である。このため液体のような凝集系の理論構築にあたってコンピュータシミュレーションは非常に有効な手段となってきた。特に古典分子動力学シミュレーション（以下、MD）は、第一原理的な方法よりもはるかに大きな計算領域を扱うことができるため、分子が多数集合したときの性質を議論するのに向いている。

凝集系の中でも気液¹⁾・固液²⁾などの界面現象は、特に厳密な理論的考察や実験的観察が難しい。これらの界面現象は吸着・脱着^{3),4)}や核生成^{1),5)}、濡れ^{6),7)}などの現象の基礎となっている。だが界面現象は、分子スケールのミクロな構造に大きく依存しているなどの理由により、解析が困難になっている。

たとえば、低温高圧下のメタン–水界面では急速にクラスレート水和物の結晶が成長することが知られているが⁸⁾、クラスレート水和物の生成現象には界面のマクロな性質と個々の分子の振舞いの両方が深く関わっており、様々な実験的事実を検証することがきわめて困難である。しかし最近の研究ではメタン–水界面のMDにより、界面におけるモル分率の変化や分子の配向の様子などが明らかにされた⁹⁾。このように、実験では危険がともなう高圧な状態も計算上のパラメーターを変化させるだけで解析できるため、実験を専門とする研究者もMDに注目を集めている。

だが界面系のMDは非常に計算量が多いことが問題で、そのため十分な解析がなかなか行われていないのも現状である。計算量が多くなる原因は、系が平衡状態に達するまでに長いシミュレーション時間が必要で、かつバルクの領域を各相ごとに十分に確保するために大きい計算領域が必要となる点にある¹⁰⁾。また、高精度で負荷の大きい計算条件を用いな

^{†1} 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University

^{†2} 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

かった場合、界面における物理量の計算結果が大きく異なるという報告もある¹¹⁾。よって、界面系の MD には高速なシミュレーション環境が必要不可欠である。

1.2 シミュレーションの高速化

MD に限らず重力多体問題や連続体の粒子ベースの解法 (SPH, 渦糸法, 境界要素法など) などの粒子多体問題は、粒子ペアごとに発生する計算が全体の計算時間の大半を占めている。しかしこの種の計算は並列化することが可能で、さらにスレッドごとに共有できる情報があるため、低速なメモリと多数のパイプラインを持つ専用設計のハードウェアを用いると効果的に高速化できることが示されている^{12),13)}。MD の分野ではこれまでに MDGRAPE シリーズや MDEngine などのアクセラレータが開発されてきた¹⁴⁾⁻¹⁹⁾。これらはコスト・設置面積・電力的に優れたシミュレーション環境を提供し、シミュレーション可能な時空間スケールを拡大することで様々な新しい物理現象を解明してきた。

一方で専用計算機は開発コストの高さから単価が高額になってしまい、広く用いられていないのが現状である。専用計算機が台頭してきた初期のころはそれこそ非常に低価格で汎用的な計算機よりも十分に速いものが開発できた。しかし、次第にプロセッサの微細・高集積化技術が発達するにつれてプロセッサの開発そのものが複雑になり、高速なプロセッサの開発に多大なコストを必要とするように変化してきた。そのため、市場規模が大きい汎用的なプロセッサは急速に発展し、反対に専用計算機は開発そのものが困難になってきた。

しかし現状では、汎用的なプロセッサと専用計算機で同じ粒子多体シミュレーションを行った場合、設置面積や消費電力的に汎用的なプロセッサを用いた方が劣っているため²⁰⁾、シミュレーションに向けたアーキテクチャを持つ計算機もいまだに不可欠である。そこで現在、これらの大規模物理シミュレーション分野において必要とされているハードウェアは、以下の2点を備えたものであるといえる。

- (1) 市場規模が大きく開発が頻繁に行える。
- (2) プロセッサ内に多数のパイプラインを有し、効果的に物理シミュレーションを高速化できる。

本研究ではそのような可能性を秘めたハードウェアとしてグラフィックカードを選択した。

グラフィックカード (Graphic Processing Unit, 以下 GPU) は、コンピュータの中でグラフィックス処理を専門に担当するハードウェアとして開発されてきたものである。この GPU は最近の一般的なパソコンにはほとんど搭載されているため、専用計算機と比べて圧倒的に大きな市場規模を持っており、頻繁にアップデートを繰り返している。また近年のコンピュータでは、ゲームや OS などにおいて膨大な計算量を必要とするグラフィック処理が

求められており、それにとまって GPU もパイプライン数の増加という形でこれに対応してきた。最近ではこれらのパイプラインにおいてグラフィックス処理以外の汎用的な計算をするための開発環境も整い出し、まさに上にあげた今後求められるハードウェアの条件を満たしているといえる。

特に GPU には、以下のような特徴があるだろう。

- (1) 1枚あたりで理論ピーク値で数百 GFlops から TFlops の高速な性能を実現している。
- (2) 1枚あたり数万円で入手でき、他の専用・準専用計算機に比べて安価である。

そのため少しずつではあるが、GPU を物理シミュレーションに応用する研究が最近登場してきた^{21),22)}。これらには、重力多体問題に関するものが数例と^{23),24)}、MD に関するものなどがある²⁵⁾。

しかし、このように GPU を用いて MD を高速化した例はあるものの、計算精度の評価などをきちんと行い示した例はない。これまでに専用計算機などにおいて苦労した点は、実際に MD シミュレーションにおいて十分な精度を保ちつつ高速化する、という点であり、実用化に向けてここを避けて通るのは難しいだろう。また、それらの過去の研究で GPU を用いた高速計算の使用例として取り上げられた結果は、他の研究例と比較できるようなものではなく、実際に GPU が MD シミュレーションにおいて本当に使えるものなのか、いまだに疑問を残したままである。

1.3 本研究の目的

本研究ではまず、GPU を用いて MD の高速化を行うことを目的とした。そしてこれまで議論されてこなかった計算精度の評価を行い、GPU を用いた MD 計算が示す結果の妥当性について検討を行った。また、本研究では実際に GPU を用いて水気液界面の MD を実行し、他の水気液界面の MD に関する論文との比較による計算結果の検証および、それらの論文よりも大きな時空間スケール・幅広い温度圧力条件における計算結果の提示とその考察を行った。

2. 方 法

2.1 古典分子動力学シミュレーションの方法

古典分子動力学シミュレーションは、ニュートンの運動方程式を数値積分し実在する物質の様々な性質を観察する手法である。分子間に働く相互作用は、経験的・平均的な分子間の相互作用を表現する有効ポテンシャル関数 U_{inter} を用いて計算している。

$$U_{\text{inter}} = U_{\text{vdW}} + U_{\text{elec}} \quad (1)$$

U_{inter} は, van der Waals 力に相当する部分 (U_{vdW}) と極性分子間に働く強い静電相互作用に相当する部分 (U_{elec}) に分けて計算される.

$$U_{\text{vdW}} = \frac{1}{2} \sum_{i,j} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (2)$$

$$U_{\text{elec}} = \frac{1}{2} \sum_{i,j} \frac{Q_i Q_j}{r_{ij}} \quad (3)$$

U_{elec} の係数 $1/4\pi\epsilon_0$ は省略してある. Lennard-Jones (以下, LJ) パラメータ σ_i, ϵ_i は Lorentz-Berthelot 則 $\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}, \sigma_{ij} = (\sigma_i + \sigma_j)/2$ によって結合される. 一方, 運動方程式の数値積分には速度ベレ法などが用いられ, 適切な条件で解いた場合には系のハミルトニアン $H = \sum p_i^2/2m_i + U$ が保存することが分かっている. また, 境界条件は限られたケースを除き主に 3 次元周期境界条件が用いられる.

2.2 相互作用ポテンシャルの計算方法

有効ポテンシャル (二体ポテンシャル) の計算量は MD の計算時間全体の大半 (> 99%) を占めるため, 様々な計算方法が考案されている. また専用ハードウェアによる高速化もこの部分的に的を絞ったものが多い²⁶⁾. これは本研究における GPU を用いた高速化でも同様である.

二体ポテンシャルの計算アルゴリズムの中で, 最も単純かつ基礎的なものがカットオフ法である. 二体ポテンシャル u_{ij} は通常 $r_{ij} \rightarrow \infty$ で $u_{ij} \rightarrow 0$ となるため, r_{ij} がある有限の距離 r_c を超えた場合に $u_{ij} \approx 0$ とする近似計算法をカットオフ法, r_c をカットオフ半径と呼ぶ.

カットオフ法における粒子間距離 r_{ij} の計算は, $\mathcal{O}[N^2]$ の計算量となる (N は粒子数). そのため大きな計算領域では $\mathcal{O}[N]$ の Cell Index 法などが用いられる. Cell Index 法では, 計算領域を等しい大きさの直方体セルに分割する処理を行う. まず全粒子間の距離を探索する前に, あらかじめ各粒子がどのセルの中に入るかを計算する. そして二体ポテンシャルを計算する際には, 近傍のセル間で互いに粒子を参照し合い, そこでカットオフ法と同様の計算を行う.

LJ ポテンシャルは 0 への収束が早いいためカットオフ・Cell Index 法がよく用いられるが, 式 (3) で表される Coulomb ポテンシャルは収束が遅いため, Ewald 法などの特殊な方法が使われる. Ewald 法は周期境界条件下で無限遠からのポテンシャルの寄与を高精度で計算する方法である. Ewald 法ではまず, 0 への収束が早い補誤差関数 $\text{erfc}(r) (= 1 - \text{erf}(r))$

を用いて Coulomb ポテンシャルを近距離成分と遠距離成分に分解する. そして近距離成分 U_{real} はカットオフ法と同様の手順で, 遠距離成分 U_{wave} はフーリエ変換したうえで高周波成分をカットして計算する.

$$U_{\text{elec}} = \underbrace{\text{erfc}(r) U_{\text{elec}}}_{U_{\text{real}}} + \underbrace{\text{erf}(r) U_{\text{elec}}}_{U_{\text{wave}}} \quad (4)$$

$$U_{\text{real}} = \frac{1}{2} \sum_{i,j} Q_i Q_j \frac{\text{erfc}(\alpha r_{ij})}{r_{ij}} \quad (5)$$

$$U_{\text{wave}} = \underbrace{\frac{1}{2\pi V} \sum_{\substack{m < m_c \\ m \neq 0}} \frac{\exp(-\pi^2 m^2 / \alpha^2)}{m^2} |S(m)|^2}_{U_{\text{recip}}} - \frac{1}{2} \sum_{i=1}^N Q_i^2 \frac{\text{erf}(\alpha r)}{r} \Big|_{r=0} \quad (6)$$

α は $U_{\text{real}}, U_{\text{wave}}$ の比率を変化させるパラメータで, カットオフ半径付近において U_{real} が十分 0 に収束するように選択される. m, m, m_c はそれぞれ逆格子ベクトルとその絶対値およびカットオフ半径, V は計算領域の体積である. また $S(m)$ は Structure factor と呼ばれ以下の形で表される ($\iota = \sqrt{-1}$).

$$S(m) = \sum_i^N Q_i \exp(2\pi \iota m \cdot r_i) \quad (7)$$

オリジナルの Ewald 法は計算量が $\mathcal{O}[N^{3/2}]$ となり特に波数空間の項 (U_{recip}) が計算のホットスポットとなる. そこで, 波数空間の計算に高速フーリエ変換 (FFT) に応用させた方法が Particle Mesh Ewald (PME) 法である²⁷⁾. PME 法は FFT を用いるために原理的には計算量が $\mathcal{O}[N \log N]$ となり, 大規模系の MD では広く用いられている. PME 法ではまず計算空間を等間隔に分割したメッシュ上に粒子が持つ電荷を外挿し, メッシュ上で求めたポテンシャルエネルギーを粒子に内挿する, という方法をとる. この内挿・外挿に B-Spline 曲線

$$M_n(u) = \frac{1}{(n-1)!} \sum_{j=0}^n (-1)^j \frac{n!}{j!(n-j)!} \max(u-j, 0)^{n-1} \quad (8)$$

を用いた場合, 式 (6) の (U_{recip}) は以下のように書き換えられる²⁸⁾.

$$U'_{\text{recip}} = \frac{1}{2} \sum'_{m \neq 0} Q(\mathbf{m})(\mathcal{F}(\theta') \star Q)(\mathbf{m}) \quad (9)$$

$$\theta'(\mathbf{m}) = \frac{1}{\pi V} \frac{\exp(-\pi^2 m^2 / \alpha^2)}{m^2} \frac{|b_1(m_1)|^2 |b_2(m_2)|^2 |b_3(m_3)|^2}{\exp\left(\frac{2\pi i(n-1)m_\beta}{K_\beta}\right)} \quad (10)$$

$$b_\beta(m_\beta) = \frac{\sum_{j=0}^{n-2} M_n(j+1) \exp\left(\frac{2\pi i m_\beta j}{K_\beta}\right)}{(\beta = 1, 2, 3)} \quad (11)$$

K_1, K_2, K_3 は各方向のメッシュ分割数, m_1, m_2, m_3 は \mathbf{m} の各成分, $\sum'_{m \neq 0}$ は $m = 0$ を除くすべての \mathbf{m} に対する和を表す. また n は B-Spline 曲線の補間次数, $\mathcal{F}(A)$ は行列 A のフーリエ変換, $(A \star B)$ は行列 A, B の畳み込み, $Q(\mathbf{m})$ は (m_1, m_2, m_3) 番目のメッシュに割り振られた電荷の値を表す. 計算の手順としては, まずあらかじめ $\mathcal{F}(\theta')$ を計算しておく必要がある. 次に $Q(\mathbf{m})$ を求め FFT した後, $\mathcal{F}(\theta')$ を乗じて逆 FFT し, $Q(\mathbf{m})$ を掛けてメッシュ上のポテンシャルエネルギーを得る. そして最後に各粒子のポテンシャルエネルギーおよび力を, メッシュからの内挿により計算する.

2.3 グラフィックカードを用いた高速化方法

本研究では NVIDIA 社の GPU である GeForce8800 GTX および GeForce8800GT を用いて MD の高速化を行った²⁹⁾. 開発言語には CUDA を使用した²⁹⁾. 各 GPU のスペックを表 1 に, また例として 8800GTX のハードウェア構成を図 1 に示してある. これらの GPU は SIMD Block 単位で複数の Processors が高速な Shared memory を共有しており, このようなアーキテクチャは二体ポテンシャルの並列計算に都合が良い. また, グラフィック用途ではビットマップデータを格納するための Texture cache は, あらかじめ用意した配列の要素間を一次補間して値を返す関数テーブルとして使用でき, 補誤差関数や指数関数などを含む荷の大きい関数を, 直接計算するよりも高速に見積もることができる.

二体ポテンシャルの計算は, 実空間 (LJ ポテンシャルおよび Ewald 法近距離成分) と波数空間に分けて実行される. 本研究では CUDA を用いて実空間の計算を以下のように GPU で並列化した.

- (1) 粒子数分のスレッドを立て, 各スレッドはそれぞれ異なる粒子の座標, 相互作用パラメータの情報を保持する (参照側の i 粒子と呼ぶ).
- (2) Shared memory に数百粒子程度の座標, 相互作用パラメータを読み込む (被参照側の j 粒子と呼ぶ).
- (3) 各スレッド (i 粒子) は Shared memory から他の粒子 (j 粒子) の情報をいっせいに読み出し, そこで二体ポテンシャル u_{ij} を計算する.
- (4) 最終的に各スレッドから $U_i = \sum_j^N u_{ij}$ を得るのが目的なので, 各スレッドは計算した u_{ij} を足し込んでいく.
- (5) すべての二体ポテンシャルを計算するまで手順 (2) ~ (4) を繰り返す.

ここでは U_i の計算方法しか示さなかったが, 力 $F_i = -\partial U_i / \partial r_i$ や圧力テンソルのピリ

表 1 使用した GPU のスペック²⁹⁾
Table 1 Specifications of each GPUs²⁹⁾.

	number of processors	processors clock, GHz	ideal peak performance, GFlops
8800GTX	128	1.35	518.0
8800GT	112	1.50	504.0

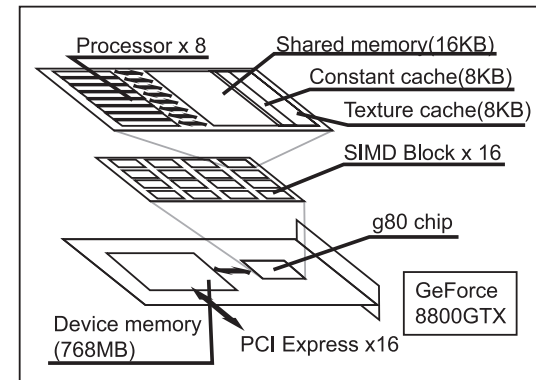


図 1 GeForce8800GTX のハードウェア構成²⁹⁾
Fig. 1 Architectures of GeForce8800GTX²⁹⁾.

アル項 $W_i = F_i \cdot r_i$ も同様の手順で計算される。

波数空間の計算には PME 法を用いた。PME 法は実空間の計算のように単純ではなく、粒子-メッシュ間の補間、畳み込み、FFT などの様々な処理が必要になる。本研究ではこのうち FFT のみを GPU で実行し、他の処理は通常どおり CPU で実行した。GPU での FFT の実行には、CUFFT ライブラリを使用した²⁹⁾。

2.4 界面張力の計算方法

本研究では界面系の MD において界面張力の算出を行った。界面張力は MD から計算される圧力テンソル $P_{\alpha\beta}(\alpha, \beta = x, y, z)$ と次のように結び付けられる。

$$\gamma = \left\{ P_{zz} - \frac{P_{xx} + P_{yy}}{2} \right\} \frac{L_z}{2} \quad (12)$$

$$VP_{\alpha\beta} = \left\langle \sum_{i=1}^N \frac{p_{i,\alpha} p_{i,\beta}}{2m_i} \right\rangle + \left\langle \sum_{i=1}^N \sum_{j>i}^N r_{ij,\alpha} F_{ij,\beta} \right\rangle$$

ここでは z 方向が界面に垂直であるとし、 L_z は z 方向の計算領域幅である。

本研究では水 (SPC/E モデル) の気液界面における界面張力の計算を行い、過去の研究と比較した。運動の時間発展には等温剛体シンプレクティック差分法³⁰⁾を採用し、相互作用の計算には GPU を用いた。時間刻みは 2 fs とし、水分子を 10,000 個含むバルク系を立方体三次元周期境界、275K、 $1\text{g}/\text{cm}^3$ で 50 ps 計算した後、 z 方向に 4 倍の長さの空き領域を挿入することで気液界面系を作成した。その後 1 ns 平衡化計算し、2 ns の間 100 ステップごとに物理量の平均をとった。実空間のカットオフ半径は 9.8\AA 、Ewald 法の収束パラメータは $\alpha = 0.31\text{\AA}^{-1}$ とし、PME 法の補間次数は 4 次、メッシュ数は $64 \times 64 \times 256$ とした。図 2 には本研究で計算した系のスナップショットを載せてある。また界面張力の計算では、式 (12) で得られた γ に以下の式で表される LJ ポテンシャルのカットオフ補正項を加えた¹⁰⁾。

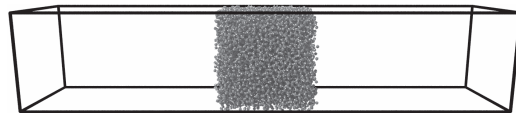


図 2 計算した水気液界面のスナップショット
Fig.2 Snapshot of simulated water surface.

$$\gamma_{\text{cor}} = 12\pi\epsilon\sigma^6(\rho_L - \rho_V)^2 \int_0^1 ds \int_{r_c}^{\infty} dr \coth\left(\frac{rs}{d}\right) \frac{3s^3 - s}{r^3} \quad (13)$$

d は 10-90 厚さ t を用いて $t = 2.1972d$ と表される界面の厚さ、 ρ_L 、 ρ_V はそれぞれ液相・気相の数密度である。

3. 結 果

3.1 グラフィックカードを用いた高速化方法

まず、Intel Core2 Quad Q6600³¹⁾ (2.4 GHz) を搭載したホストコンピュータ (CPU は 1 コアのみ使用) に GeForce8800GTX を接続し、LJ ポテンシャルのみが働く系の MD を実行した。そして最初にシミュレーションの妥当性となるハミルトニアン保存の様子を調べた。図 3 は、二体ポテンシャルを CPU で倍精度 (四角) および単精度実数 (白丸) で計算した場合と GPU で (単精度実数で) 計算した場合 (黒丸) のハミルトニアン揺らぎの大きさを示している。CPU で単精度実数で二体ポテンシャルを計算することは、MD の汎用ソフトウェアでも高速化のためにしばしば使われている技法である。今回の計算では GPU において実行してもハミルトニアンは保存し、時間刻みを小さくしたときの挙動は CPU で単精度実数を用いた場合とほぼ同じであった。通常用いられる時間刻みは 1 fs 程度であり、以上のことから GPU を用いて二体ポテンシャルを計算しても実用上問題がないこ

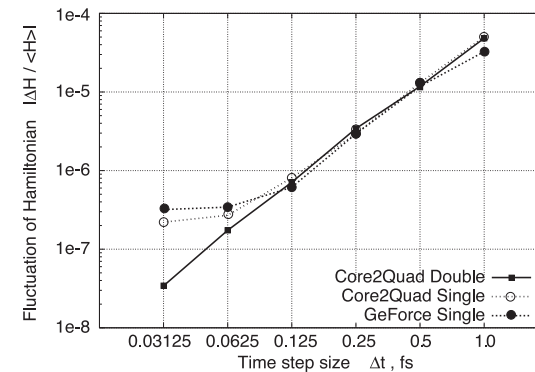


図 3 時間刻みとハミルトニアンの揺らぎの関係
Fig.3 Time step size dependency of Hamiltonian fluctuations.

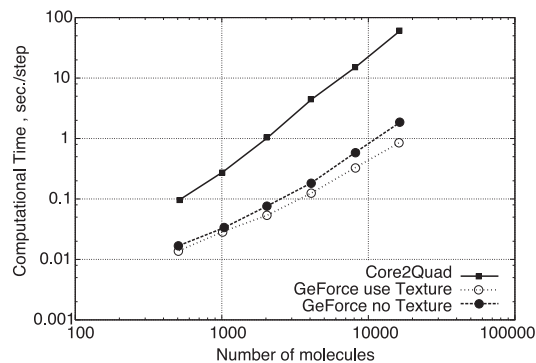


図 4 SPC/E 水分子の計算速度の比較

Fig. 4 Comparison of computational speeds for SPC/E water molecules.

とが分かった。

次に SPC/E 水分子の MD を実行し計算速度を評価した。図 4 は粒子数にともなう計算時間の変化を表し、実空間は計算量が $O[N^2]$ となるカットオフ法、波数空間は PME 法により計算している。そして分子数 16,384 のときに Texture Unit を使わずに約 32.1 倍、Texture Unit を使用して約 69.8 倍高速化した。しかし粒子数の少ない場合には 10 倍程度の高速化にとどまり、スレッド数が少ないときに並列化効率が低下することが確認された。これは、GPU コンピューティングにおいてよりスレッド数を消費するアプリケーションほど高速化が期待できることを意味する。

次に、カットオフ法よりも高速なアルゴリズムである Cell Index 法を用いて、水分子 32,768 個の系を計算した。波数空間は PME 法を用いた。全体としては GPU を用いることで 13.4 倍高速化した。表 2 は実空間および波数空間の計算時間が全体の中でそれぞれどれだけ占めているかを詳細に示している。GPU を用いない通常の計算であれば、一般には実空間の計算量が全体の大半を占めている。だが GPU を用いるとホットスポットが逆転し、波数空間の計算量が全体の半分を上回った。よって、今後さらなる計算時間の短縮を行うためには、波数空間のさらなる最適化が必要となることが分かった。

また、実空間の計算に着目し計算速度の変化の様子をより詳細に調べた。カットオフ法の場合は、全粒子間距離の探索を N^2 回行い、カットオフ半径内のペアについてのみ相互作用の計算を行う。理論的には粒子数が多くなれば、全体のペア数に対して相互作用計算を行うペア数が小さくなるので、カットオフ法の計算時間は距離の探索だけを行った時間に漸近す

表 2 Cell Index 法および PME 法を用いた 32,768 水分子の計算時間の内訳, sec./step

Table 2 Breakdown of computational time for 32,768 water molecules using Cell Index method, and PME method, sec./step.

	Real Space	Reciprocal Space	Other	Total
without GPU	13.23	1.53	0.20	14.96
with GPU	0.39	0.53	0.20	1.12

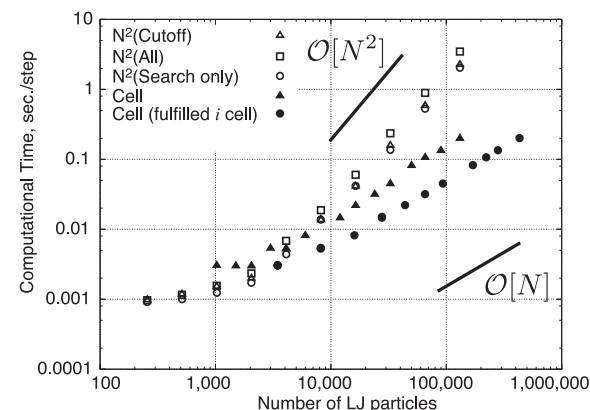


図 5 実空間計算アルゴリズムの計算速度の検証

Fig. 5 Verifications of computational algorithms for real space.

る。図 5 には、全粒子間の相互作用計算を行った場合（白四角）、カットオフ半径内に粒子間で相互作用計算を行った場合（白三角）、距離の探索のみ行った場合（白丸）の計算時間が示されている。ここでカットオフ半径は 10\AA 、数密度は 0.035\AA^{-3} である。カットオフ法の計算（白三角）は 7,000~8,000 粒子以上で距離のみを計算したケースの速度（白丸）に近づいているが、距離の探索コストが $O[N^2]$ に比例しているために大規模系（数万粒子以上）では $O[N]$ の Cell Index 法などの使用が望まれる。

Cell Index 法についても検証を行った。CUDA を用いて計算を行う場合は、全スレッド数がある数（1つの SIMD Block が同時に並列処理するスレッド数）の整数倍になっている必要があるため、粒子数も擬似的にそれらの整数倍に揃えなければならない。図 5 ではまた、GPU を用いた Cell Index 法において、それらが原因となるロスがある場合とない場

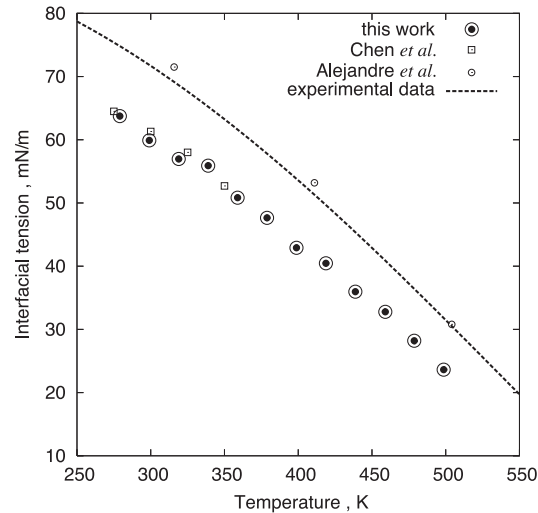


図6 SPC/E 水分子の界面張力

Fig. 6 Interfacial tensions of SPC/E water molecules.

合を比較している。図5には、実際に相互作用計算を行った場合（黒三角）、 i 粒子に無駄がないと想定した場合（黒丸）の計算時間が示されている。セルは1辺の長さが 10\AA を下回らずにできるだけたくさん分割できるように配置された。ここで、Cell Index 法の効率が i 粒子を擬似的に増やすことで約3倍低下し、実・波数空間の合計時間が実空間の計算時間の倍程度であるとすると、カットオフ法よりも Cell Index 法で高速化効率が5, 6倍程度低下していることの説明がつく。つまり、Cell Index 法の実装で高速化効率が低下している原因は、余分な i 粒子を計算している点であることが分かった。しかしこの弱点は、効率を出すためには大量の（数万以上～）スレッドが必要となる現在のGPUの仕様からくる。そのため、今後物理シミュレーションの高速化におけるGPUの利用が拡大するためには、この部分を改善する必要があるだろう。

3.2 水界面張力の計算

ここでは同じ SPC/E モデルを用いた水気液界面 MD の過去の計算結果^{(10),(11)} と実験値から求められた IAPWS の経験式⁽³²⁾ および本研究の計算結果を比較した。図6はそれぞれの界面張力を示している。それぞれの計算条件は表3に示してあり、本研究のシミュレーションはより大きい領域で行われ、さらに波数空間も十分高い解像度で計算していることが

表3 界面張力を計算した過去の研究との計算条件の比較^{(10),(11)}
Table 3 Comparisons of computational conditions^{(10),(11)}.

	Number of reciprocal vectors	Number of molecules
Alejandre, et al. ⁽¹⁰⁾	$\approx 5 \times 5 \times 20$ (Ewald)	1,000
Chen, et al. ⁽¹¹⁾	$\approx 16 \times 16 \times 83$ (PME)	512
this work	$\approx 64 \times 64 \times 256$ (PME)	10,000

分かる。

本研究の結果は分子間相互作用の計算に GPU を使用しており、実空間はカットオフ法、波数空間は PME 法を用いた。1 温度条件での MD シミュレーションに対し 1 GPU (GeForce8800GT) を使用し、計算が終了するまでそれぞれ約 1 週間かかった。

図6より、実験値から約 10 mN/m 低い値が SPC/E モデルの精度の高い界面張力の値であることがうかがえる。ここで提示されたデータを用いると、たとえば MD から求めた界面の高次の物理量（気泡核生成速度など）を、界面張力をパラメーターとして含む理論式（古典核生成理論など）と厳密に対比させることが可能になる¹⁾ ハードウェアアクセラレータを用いずにこのような計算を行うのは現実的に不可能であるため、今後はさらに多くの温度条件、異なるポテンシャルモデルに対して、Cell Index 法などの高速なアルゴリズムを用いて界面張力の計算を行っていく予定である。

4. 結 論

GPU を用いて高速な MD の計算環境を構築し、それが計算精度・計算速度ともに実用的であることを示すことができた。そして水気液界面の MD を過去の計算例よりも大規模な系で精密に行い、今回計算した SPC/E ポテンシャルモデルに関して高精度な界面張力の値を算出することができた。今後は、複数の GPU を用いた大規模計算に応用していく予定である。

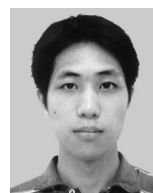
謝辞 理化学研究所の泰地真弘人博士には MD の高速化に関する助言をいただいた。また長崎大学の濱田剛博士には GPU を用いた並列計算に関する助言をいただいた。使用したクラスタに関しては科学技術振興機構 (JST) 戦略的創造研究推進事業 (CREST) の支援を受けた。

参 考 文 献

- 1) Matsumoto, M., Yasuoka, K. and Kataoka, Y.: *J. Chem. Phys.*, Vol.101, pp.7912–7917 (1994).
- 2) Andoh, Y., Kurahashi, K., Sakuma, H., Yasuoka, K. and Kurihara, K.: *Chem. Phys. Lett.*, Vol.448, pp.253–257 (2007).
- 3) Andoh, Y. and Yasuoka, K.: *J. Phys. Chem. B*, Vol.110, pp.23264–23273 (2006).
- 4) Andoh, Y. and Yasuoka, K.: *Langmuir*, Vol.21, pp.10885–10894 (2005).
- 5) Matsubara, H., Koishi, T., Ebisuzaki, T. and Yasuoka, K.: *J. Chem. Phys.*, Vol.127, pp.214507 1–11 (2007).
- 6) Koishi, T., Yasuoka, K., Ebisuzaki, T., Yoo, S. and Zeng, X.C.: *J. Chem. Phys.*, Vol.123, pp.204707 1–7 (2005).
- 7) Koishi, T., Yoo, S., Yasuoka, K., Zeng, X.C., Narumi, T., Susukita, R., Kawai, A., Furusawa, H., Suenaga, A., Okimoto, N., Futatsugi, N. and Ebisuzaki, T.: *Phys. Rev. Lett.*, Vol.93, 185701 (2004).
- 8) Ohmura, R., Matsuda, S., Uchida, T., Ebinuma, T. and Narita, H.: *Cryst. Growth Des.*, Vol.5, No.3, pp.953–957 (2005).
- 9) Reed, K.S. and Westacoot, R.E.: *Phys. Chem. Chem. Phys.*, Vol.10, pp.4614–4622 (2008).
- 10) Alejandre, J., Tildesley, D. and Chapela, G.: *Mol. Phys.*, Vol.85, pp.651–663 (1995).
- 11) Chen, F., Smith, P.E.: *J. Chem. Phys.*, Vol.126, pp.221101 1–3 (2007).
- 12) Hut, P. and Makino, J.: *Science*, Vol.283, pp.501–505 (1999).
- 13) Komeiji, Y. and Uebayasi, M.: *CBI J.*, Vol.2, pp.102–118 (2002).
- 14) Ito, T., Fukushige, T., Makino, J., Ebisuzaki, T., Okumura, S.K., Sugimoto, D., Miyagawa, H. and Kitamura, K.: *PROTEINS*, Vol.20, pp.139–148 (1994).
- 15) Fukushige, T., Makino, J., Ito, T., Okumura, S.K., Ebisuzaki, T. and Sugimoto, D.: *Publ. Astronom. Soc. Jpn.*, Vol.45, pp.361–375 (1993).
- 16) Susukita, R., Ebisuzaki, T., Elmegreen, B.G., Furusawa, H., Kato, K., Kawai, A., Kobayashi, Y., Koishi, T., McNiven, G.D., Narumi, T. and Yasuoka, K.: *Comput. Phys. Commun.*, Vol.155, pp.115–131 (2003).
- 17) Narumi, T., Ohno, Y., Okimoto, N., Koishi, T., Suenaga, A., Futatsugi, N., Yanai, R., Himeno, R., Fujikawa, S., Ikei, M. and Taiji, M.: A 185 TFLOPS simulation of amyloid-forming peptides from Yeast prion Sup35 with the special-purpose computer system MDGRAPE-3, In *Proc. SC2006*, Tampa, (2006). CD-ROM
- 18) Oda K., Miyagawa and H., Kitamura, K.: *Mol. Simul.*, Vol.16, pp.167–177 (1996).
- 19) Toyoda, S., Miyagawa, H., Kitamura, K., Amisaki, T., Hashimoto, E., Ideda, H., Kusumi, A., Miyakawa, N.: *J. Comput. Chem.*, Vol.20, pp.185–199 (1999).
- 20) Narumi, T., Kameoka, S., Taiji, M. and Yasuoka, K.: *SIAM Journal on Scientific Computing*, Vol.34, pp.3108–3125 (2008).
- 21) <http://www.gpgpu.org/>
- 22) Hamada, T., Ohno, Y., Morimoto, G., Taiji, M., Iitaka, T. and Nitadori, K.: Internals of the CUNBODY-1 library: particle/force decomposition and reduction, *AstroGPU* (2007).
- 23) Hamada, T. and Iitaka, T.: *astro-ph/0703100* (2007).
- 24) Nyland, L., Harris, M. and Prins, J.: *Proc. ACM Workshop on General-Purpose Computation on Graphics Processors* (2004).
- 25) Stone, J.E., Philips, J.C., Freddolino, P.L., Hardy, D.J., Trabuco, L.G. and Schulten, K.: *J. Comput. Chem.*, Vol.28, pp.2618–2640 (2007).
- 26) Makino, J. and Taiji, M.: *Scientific Simulations with Special-Purpose Computers – The GRAPE Systems*, John Wiley & Sons, Chichester (1998).
- 27) Darden, T.A., York, D.M. and Pedersen, L.G.: *J. Chem. Phys.*, Vol.98, pp.10089–10092 (1993).
- 28) Essmann, U., Perera, L., Berkowitz, M.L., Darden, T., Lee, H. and Pedersen, L.G.: *J. Chem. Phys.*, Vol.103, pp.8577–8593 (1995).
- 29) <http://www.nvidia.com/>
- 30) Okumura, H., Itoh, S.G. and Okamoto, Y.: *J. Chem. Phys.*, Vol.126, pp.84103 1–17 (2007).
- 31) <http://www.intel.com/>
- 32) IAPWS Release on Surface Tension of Ordinary Water Substance, IAPWS (1994). <http://www.iapws.org/>

(平成 20 年 10 月 3 日受付)

(平成 21 年 2 月 3 日採録)



坂牧 隆司

昭和 59 年生。平成 19 年慶應義塾大学工学部卒業。同年から現在まで慶應義塾大学大学院理工学研究科に所属。GPU を用いた分子動力学シミュレーションの高速化，界面現象等の研究に従事。第 21 回分子シミュレーション討論会にて学生ポスター賞受賞。分子シミュレーション研究会，分子科学会各会員。



成見 哲 (正会員)

昭和 45 年生。平成 10 年東京大学大学院総合文化研究科広域科学専攻博士後期課程修了。平成 13 年から 19 年にかけて理化学研究所ゲノム科学総合研究センター研究員。平成 19 年から慶應義塾大学大学院理工学研究科特別研究講師。分子動力学シミュレーション専用計算機の開発、GPU や PS3 による高速化等の研究に従事。学術博士。平成 8 年に日経サイエンス社主催第 2 回 Computer Visualization Contest 最優秀賞，平成 12 年，18 年に IEEE Gordon Bell Prize 各受賞。



泰岡 顕治

昭和 43 年生。平成 9 年名古屋大学大学院応用物理学専攻博士後期課程修了。同年から平成 10 年まで理化学研究所基礎科学特別研究員。平成 10 年から慶應義塾大学工学部機械工学科助手。平成 12 年から慶應義塾大学工学部機械工学科専任講師。平成 16 年から慶應義塾大学工学部機械工学科助教授。平成 19 年から慶應義塾大学工学部機械工学科准教授。分子動力学シミュレーションを用いた界面現象，核生成現象等の研究に従事。博士(工学)。平成 12 年に IEEE Gordon Bell Prizes，平成 18 年に分子シミュレーション研究会学術賞各受賞。日本機械学会，日本物理学会，日本伝熱学会，日本熱物性学会，分子シミュレーション研究会各会員。