

## GPU を用いた核引力積分計算の高速化 A GPU Implementation of Nuclear Attraction Integral Computation

横川 宜矢<sup>1)</sup> 伊藤 靖朗<sup>1)</sup> 藤井 晴斗<sup>1)</sup> 鈴木 寛太<sup>1)</sup> 辻 聡樹<sup>1)2)</sup> 中野 浩嗣<sup>1)</sup> 笠置 明彦<sup>2)</sup>  
Nobuya Yokogawa Yasuaki Ito Haruto Fujii Kanta Suzuki Satoki Tsuji Koji Nakano Akihiko Kasagi

### 1 はじめに

量子化学計算は分子の構造や化学反応を実際に実験することなく解明できる手段として、化学や創薬、製造などの分野で広く用いられている。しかし、量子化学計算は計算コストが非常に大きいという課題が存在する。そのため、本研究では量子化学計算を行う上で事前計算として必要となる核引力積分の計算を GPU の優れた並列処理能力を用いることで高速化を目指す。核引力積分の既存手法として Obara-Saika method [1] や McMurchie-Davidson method [2] が広く用いられており、それらを効果的に計算する GPU 実装を紹介し、さらに本研究ではそれらを組み合わせた Hybrid 法を提案する。

### 2 電子の軌道

量子化学計算において、分子のエネルギーなどを計算する際は電子の軌道について考慮する。電子が原子において従う軌道は原子軌道と呼ばれ、この原子軌道が核引力積分などの分子積分の入力となる。原子軌道には  $s, p, d, f, g, \dots$  といった種類が存在し、これは原子軌道が有する方位量子数 (各方向の角運動量の和) に対応する。この原子軌道を表現するために、式 (1) で表される基底関数  $\chi(\mathbf{r})$  が使用される。

$$\begin{aligned} \chi(\mathbf{r}) &= D_{1k}G_1(\mathbf{r}) + D_{2k}G_2(\mathbf{r}) + \dots + D_{mk}G_m(\mathbf{r}) \\ &= \sum_{i=1}^m D_{ik}G_i(\mathbf{r}) \end{aligned} \quad (1)$$

ここで  $m$  は縮約数、 $D_{ik}$  は係数を表す。また、 $G(\mathbf{r})$  はガウス型軌道と呼ばれ、以下の式で表される。

$$G_i(\mathbf{r}) = N_\chi (x - A_x)^l (y - A_y)^m (z - A_z)^n \exp(-\alpha r_A^2) \quad (2)$$

式 (2) において、 $N_\chi$  は規格化定数、 $A(A_x, A_y, A_z)$  はガウス型軌道の中心、 $l, m, n$  は角運動量、 $\alpha$  は指数部をそれぞれ表している。

### 3 核引力積分

核引力積分は、ある電子がすべての原子核から受ける引力に関する積分である。2つの基底関数  $\chi_p, \chi_q$  に対する核引力積分  $V_{pq}$  は

$$\begin{aligned} V_{pq} &= \sum_{c=1}^{N_c} \int \chi_p(\mathbf{r}) \left( -\frac{Z_c}{r_c} \right) \chi_q(\mathbf{r}) d\mathbf{r} \\ &= \sum_{c=1}^{N_c} \sum_{a=1}^{m_p} \sum_{b=1}^{m_q} D_a D_b \int \phi_a(\mathbf{r}) \left( -\frac{Z_c}{r_c} \right) \phi_b(\mathbf{r}) d\mathbf{r} \\ &= \sum_{c=1}^{N_c} \sum_{a=1}^{m_p} \sum_{b=1}^{m_q} D_a D_b v_{ab}(c) \end{aligned} \quad (3)$$

として定義される。ここで、 $N_c$  は原子核の個数を表す。核引力積分では、あらゆる基底関数の組に対して計算する必要があり、基底関数の個数を  $M$  とすると  $M^2$  回の計算が必要である。

- 1) 広島大学大学院 先進理工系科学研究科
- 2) 富士通株式会社 コンピューティング研究所

式 (3) において  $v_{ab}(c)$  を求める際、既存手法として Obara-Saika method [1] (以下、OS 法) と McMurchie-Davidson method [2] (以下、MD 法) が広く用いられている。

#### 3.1 OS 法

OS 法では、 $v_{ab}(c) = \theta_{iljkmn}^0$  とし、以下の再帰式を用いて計算する。 $j, m, k, n$  についての再帰も同様である。

$$\begin{cases} \theta_{i+1,ljkmn}^N = X_{PA}\theta_{iljkmn}^N - X_{PC}\theta_{iljkmn}^{N+1} + \frac{1}{2p} \left\{ i(\theta_{i-1,ljkmn}^N - \theta_{i-1,ljkmn}^{N+1}) + l(\theta_{i,l-1,jkmn}^N - \theta_{i,l-1,jkmn}^{N+1}) \right\} \\ \theta_{il+1,jkmn}^N = X_{PB}\theta_{iljkmn}^N - X_{PC}\theta_{iljkmn}^{N+1} + \frac{1}{2p} \left\{ i(\theta_{i-1,ljkmn}^N - \theta_{i-1,ljkmn}^{N+1}) + l(\theta_{i,l-1,jkmn}^N - \theta_{i,l-1,jkmn}^{N+1}) \right\} \\ \vdots \\ \theta_{00,00,00}^N = -\frac{2\pi Z_c}{p} \exp\left(-\frac{\alpha\beta}{p}D^2\right) F_N(-pR_{PC}^2) \end{cases} \quad (4)$$

ここでは、 $D = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2}$ 、 $p = \alpha + \beta$  とする。

#### 3.2 MD 法

MD 法では、式 (3) の  $v_{ab}(c)$  は以下の式で求まる。

$$v_{ab}(c) = -\frac{2\pi Z_c}{p} \sum_{t=0}^{i+l} \sum_{u=0}^{j+m} \sum_{v=0}^{k+n} E_t^{i,l} E_u^{j,m} E_v^{k,n} R_{t,u,v}^0 \quad (5)$$

このとき、 $E_t^{i,l}$  は式 (6) の再帰式を用いることで求まり、 $R_{t,u,v}^0$  は式 (7) の再帰式を用いることで求まる。

$$\begin{cases} E_t^{i+1,l} = \frac{1}{2p} E_{t-1}^{i,l} - \frac{\beta}{p} X_{AB} E_t^{i,l} + (t+1) E_{t+1}^{i,l} \\ E_t^{i,l+1} = \frac{1}{2p} E_{t-1}^{i,l} + \frac{\alpha}{p} X_{AB} E_t^{i,l} + (t+1) E_{t+1}^{i,l} \\ E_0^{0,0} = \exp\left(-\frac{\alpha\beta}{p} X_{AB}^2\right) \\ E_t^{i,l} = 0 \quad (t < 0 \quad \text{or} \quad i+l < t) \end{cases} \quad (6)$$

$$\begin{cases} R_{t+1,u,v}^0 = t R_{t-1,u,v}^{N+1} + X_{PC} R_{t,u,v}^{N+1} \\ R_{t,u+1,v}^0 = u R_{t,u-1,v}^{N+1} + Y_{PC} R_{t,u,v}^{N+1} \\ R_{t,u,v+1}^0 = v R_{t,u,v-1}^{N+1} + Z_{PC} R_{t,u,v}^{N+1} \\ R_{0,0,0}^N = (-2p)^N F_N(-pR_{PC}^2) \\ R_{t,u,v}^0 = 0 \quad (t < 0 \quad \text{or} \quad u < 0 \quad \text{or} \quad v < 0) \end{cases} \quad (7)$$

どちらの手法においても点  $C$  は原子核の座標であり、点  $P$  はガウス型軌道の中心を結ぶ線分  $AB$  を  $\beta : \alpha$  に内分する点である。点  $P$  は  $P = \frac{\alpha A + \beta B}{\alpha + \beta}$  で求まる。また Boys 関数  $F_N(X)$  を求める必要があるが、本研究では先行研究 [3] の手法をもとに、近似的に求めた。

### 3.3 shellの使用

核引力積分を計算する上でボトルネックとなるのは Boys 関数の評価であり、この Boys 関数の評価回数を可能な限り減らすことが核引力積分の高速化において重要となる。核引力積分と同様に、二電子積分においても Boys 関数の評価が必要であり、Boys 関数の評価を削減して二電子積分の計算を高速化した研究として、Ivanらの研究がある[4]。Ivanらは、同じ原子核座標と指数部を持ち、なおかつ角運動量の和が等しいガウス型軌道の集合を primitive-shell と定義し、この primitive-shell のデータ構造を用いることで Boys 関数の評価回数を削減した。本研究においても、Ivanらの論文で述べられている primitive-shell のデータ構造を用いて Boys 関数の評価回数を削減した。

### 4 提案手法

本研究では、3.1節と3.2節で紹介した OS 法と MD 法を組み合わせた Hybrid 法を提案する。GPU 実装において、OS 法と MD 法の再帰関係(式(4),(6),(7))を展開すると、対象とする基底関数  $\chi$  の軌道の組が低角運動量の場合は OS 法が高速に求まり、高角運動量の場合は MD 法が高速に求まるということが実験的に分かった。そのため Hybrid 法では、求める核引力積分の軌道の組が低角運動量の場合は OS 法を、高角運動量の場合は MD 法を使用する。図1では、各軌道の組における核引力積分を Hybrid 法で計算する際、どちらのアルゴリズムを使用するかをマッピングしている。なお、本研究では g 軌道までの核引力積分を対象とする。

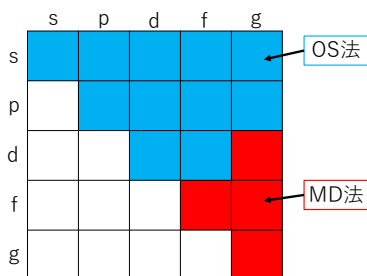


図1: Hybrid 法使用時、各軌道の組に関して OS 法と MD 法のどちらを使用するかをマッピングしている。

GPU 実装時に OS 法と MD 法の再帰関係(式(4),(6),(7))を展開した理由は、計算量の削減と再帰関数呼び出しにかかるオーバーヘッドの削減、さらに GPU 実装時の遅延要因である Warp-divergence を回避するためである。

### 5 性能評価

本章では様々な分子に対して、CPU 実装 (OS 法のみ, MD 法のみ), GPU 実装 (OS 法のみ, MD 法のみ, Hybrid 法) の5つの実装で性能を評価する。原子核の座標は原子単位系で  $[-3.0, 3.0]$  の乱数を用いて設定し、基底関数として Basis Set Exchange[5]にて公開されている STO-6G と cc-pVQZ を使用した。実行環境として、GPU には NVIDIA A100 を、CPU には Intel Xeon Gold 6338 CPU を用いた。

様々な分子に対して、STO-6G を使用した際の実行時間を表1に、cc-pVQZ を使用した際の実行時間を表2に示す。なお、CPU に対する GPU の高速化率は CPU 実

装における最速な手法の実行時間と GPU 実装における Hybrid 法の実行時間から算出した。

表1: 核引力積分 (STO-6G) の実行時間

テストケース	CPU[ms]		GPU[ms]			高速化率 CPU/GPU
	OS 法	MD 法	OS 法	MD 法	Hybrid 法	
(H <sub>2</sub> O) <sub>10</sub>	254.62	506.92	0.153	0.172	0.146	1,743.97
(C <sub>2</sub> H <sub>6</sub> O) <sub>10</sub>	4,468.43	8,761.45	0.993	1.095	0.980	4,559.62
(H <sub>2</sub> SO <sub>4</sub> ) <sub>10</sub>	8,633.35	18,594.58	1.229	1.458	1.221	7,070.72
(Ca(OH) <sub>2</sub> ) <sub>10</sub>	4,155.95	9,262.56	0.660	0.771	0.647	6,423.42

表2: 核引力積分 (cc-pVQZ) の実行時間

テストケース	CPU[s]		GPU[ms]			高速化率 CPU/GPU
	OS 法	MD 法	OS 法	MD 法	Hybrid 法	
(H <sub>2</sub> O) <sub>10</sub>	96.71	116.86	95.75	22.30	20.85	4,638.37
(C <sub>2</sub> H <sub>6</sub> O) <sub>10</sub>	2,554.55	3,110.99	334.90	90.60	87.21	29,291.94
(H <sub>2</sub> SO <sub>4</sub> ) <sub>10</sub>	3,301.10	3,704.70	297.19	87.33	81.81	40,350.81
(Ca(OH) <sub>2</sub> ) <sub>10</sub>	1,178.58	1,493.13	202.09	52.77	50.78	23,209.53

表1からは、STO-6G では提案手法は CPU と比較して最大 7,070.72 倍の高速化を達成したことが分かる。また、提案した Hybrid 法を用いることで OS 法を単独で使用する場合とほぼ同速で計算することができ、MD 法を単独で使用する場合より高速に計算できたことが確認できる。表2からは、cc-pVQZ では提案手法は CPU と比較して最大 40,350.81 倍の高速化を達成したことが分かる。また、提案した Hybrid 法を用いることで OS 法と MD 法のアルゴリズムをそれぞれ単独で使用する場合より高速に計算できたことが確認できる。

### 6 まとめ

本研究では、量子化学計算における核引力積分について、GPU 実装時に Obara-Saika method と McMurchie-Davidson method の再帰関係を展開した。展開した GPU 実装について実行時間を計測すると、求める核引力積分の軌道の組が低角運動量の場合は OS 法が高速に求まり、高角運動量の場合は MD 法が高速に求まるということが分かった。そこで、計算対象の軌道の組の種類によって使用するアルゴリズムを切り替える Hybrid 法を提案し、さらなる高速化を図った。その結果、提案手法は CPU での逐次計算と比較して最大 40,350 倍の高速化を達成した。

#### 参考文献

- [1] S. Obara and A. Saika. General recurrence formulas for molecular integrals over Cartesian Gaussian functions. *The Journal of Chemical Physics*, Vol. 89, No. 3, pp. 1540–1559, 08 1988.
- [2] Larry E McMurchie and Ernest R Davidson. One and two electron integrals over Cartesian Gaussian functions. *Journal of Computational Physics*, Vol. 26, No. 2, pp. 218–231, 1978.
- [3] S. Tsuji, Y. Ito, K. Nakano, and A. Kasagi. Efficient GPU-Accelerated Bulk Evaluation of the Boys Function for Quantum Chemistry. In *2023 Eleventh International Symposium on Computing and Networking (CANDAR)*, pp. 49–58, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society.
- [4] Ivan S. Ufimtsev and Todd J. Martínez. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *Journal of Chemical Theory and Computation*, Vol. 4, No. 2, pp. 222–231, 2008.
- [5] Benjamin P. Pritchard, Doaa Altarawy, Brett Didier, Tara D. Gibson, and Theresa L. Windus. New basis set exchange: An open, up-to-date resource for the molecular sciences community. *Journal of Chemical Information and Modeling*, Vol. 59, No. 11, pp. 4814–4820, 2019. PMID: 31600445.