C-010

GPU を用いたスクリーニングによる二電子積分の高速化
A GPU implementation of Two-Electron Repulsion Integral Screening

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

1 はじめに

二電子積分とは，シュレディンガー方程式を数値的に解く量子化学の分野において，電子間の相互作用を考慮するための積分計算である．その計算量は膨大で，例えば代表的な分子軌道計算手法であるハートリーフォック法においては二電子積分の計算が実行時間の大半を占める．二電子積分は 4 つの原子軌道のあらゆる組み合わせに対して積分値を計算する問題であり，個々の組み合わせの計算は独立しているため，GPU などを用いた並列計算による高速化が提案されている [1, 2]．さらに，積分値はシュワルツの不等式を用いてその上界を事前に計算でき，その値が十分小さければ積分計算を省略できる．しかし，このスクリーニングでは二電子積分の全組み合わせに対して上界の計算と閾値判定を行うため，計算量のオーダーは変わらない．そこで本研究では，冗長な上界計算や閾値判定を排除したスクリーニングを実現する，GPU アーキテクチャに適した二電子積分計算の並列アルゴリズムを提案する．

2 二電子積分

2.1 定義

二電子積分は，電子の座標を $\boldsymbol{r}$ とした波動関数に関する積分で，4 つの原子軌道 χ_μ, χ_ν と $\chi_\lambda, \chi_\sigma$ にそれぞれ分布する電子 1 と 2 について

$$(\mu\nu|\lambda\sigma) = \iint \chi_\mu(\boldsymbol{r}_1)\chi_\nu(\boldsymbol{r}_1)\frac{1}{r_{12}}\chi_\lambda(\boldsymbol{r}_2)\chi_\sigma(\boldsymbol{r}_2)d\boldsymbol{r}_1 d\boldsymbol{r}_2 \quad (1)$$

と定義される．ここで，原子軌道 $\chi(\boldsymbol{r})$ は原子の周りを分布する電子の軌道であり，一般にガウス型軌道の線形和で表される．

$$\chi(\boldsymbol{r}) = w_1 G_1(r) + w_2 G_2(r) + w_3 G_3(r) + \cdots \quad (2)$$

原子軌道はその形によって s 軌道，p 軌道，d 軌道などに分類され，それぞれ原子の方位量子数が $0, 1, 2$ と対応する．さらに，式 (1) に式 (2) を代入することで，原子軌道に対する二電子積分はガウス型軌道に対する二電子積分（原始積分）の線形和で求めることができる．

$$(\mu\nu|\lambda\sigma) = \sum_a \sum_b \sum_c \sum_d w_{\mu a} w_{\nu b} w_{\lambda c} w_{\sigma d} [ab|cd], \quad (3)$$

$$[ab|cd] = \iint G_a(\boldsymbol{r}_1) G_b(\boldsymbol{r}_1) \frac{1}{r_{12}} G_c(\boldsymbol{r}_2) G_d(\boldsymbol{r}_2) d\boldsymbol{r}_1 d\boldsymbol{r}_2 \quad (4)$$

本研究では，原始積分 $[ab|cd]$ は *Head-Gordon-Pople method* [3] を用いて計算した．また，二電子積分は 4 つの原子軌道 $\chi_\mu, \chi_\nu, \chi_\lambda, \chi_\sigma (1 \le \mu, \nu, \lambda, \sigma \le M)$ の組み合わせの数だけ求める必要があり，その計算量は原子軌道の数 M に対して $O(M^4)$ である．実際には，二電子積分には対称性が存在するため，計算が必要な積分 $(\mu\nu|\lambda\sigma)$ の数は $M(M+1)(M^2+M+2)/8$ となる．M の値は原子の数や原子番号，ガウス型軌道の種類や数を決定する基底関数によって変化する．

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

2.2 スクリーニング

二電子積分の計算は，積分値の上界に基づいたスクリーニングを適用することで十分な精度を維持したまま高速化できる．シュワルツの不等式を用いると，原始積分における絶対値の上界 [4] は

$$|[ab|cd]| \le \sqrt{[ab|ab]}\sqrt{[cd|cd]} = T_{ab}T_{cd} \quad (5)$$

である．前処理として，ガウス型軌道のペアに対する二電子積分の平方根 T_{ab} を $O(M^2)$ で事前計算することで，$[ab|cd]$ を計算する前にその上界はメモリから計算済みの因数 T_{ab}, T_{cd} を読み出すことで求められる．積分値の上界 $T_{ab}T_{cd}$ が設定した閾値 θ を下回れば，$[ab|cd] \approx 0$ と近似して複雑な積分計算をスキップできる．このとき，閾値 θ は精度と計算時間のトレードオフを制御する変数である．

3 GPU を用いた二電子積分計算

3.1 GPU アーキテクチャ

図 1 に示すように，GPU は計算負荷の大きい処理を Block と呼ばれる Thread のグループに分割し，それらを大量の計算コアを抱える複数の SM で独立に並列計算することで高いスループットを実現する．Block 内の Thread は Warp という単位に分割され，Warp 内の各 Thread は同じ命令を並列に実行することができる．しかし，if-else 文などの条件分岐によって Warp 内の Thread が異なる命令を選択した場合に，*Warp divergence* という並列実行の性能低下を引き起こす．

3.2 二電子積分計算の並列化

本研究では，Ufimtsev らが提案した 1T1B と呼ばれる並列化手法 [1] に基づいてスクリーニングを行う．同じ方位量子数や核座標を持つ原子軌道 $\chi(\boldsymbol{r})$ を表すために必要なガウス型軌道 $G(\boldsymbol{r})$ のうち係数 w などが一致したものをひとつの Shell という単位でまとめると，二電子

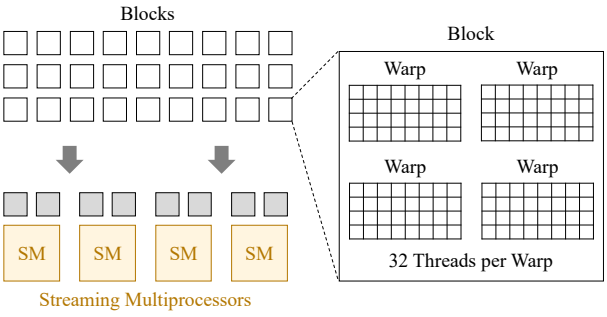

図 1 GPU のプログラミングモデル

積分は 4 つの Shell の全ての組み合わせに対して $[ab|cd]$ を計算する問題に置き換えられる．Shell の 4 つ組に対して GPU の Thread を割り当てて，ひとつの Thread が方位量子数に応じて複数の原始積分をまとめて計算することで，$[ab|cd]$ を高速に求められる．各 Thread は，式 (3) に基づいて a, b, c, d に対応する原子軌道の 4 次元行列 $(\mu\nu|\lambda\sigma)$ に計算した積分値を排他的に加算する．

4 提案手法

GPU のアーキテクチャに適した効率的なスクリーニングを可能にする，Block の動的割り当てを利用した二電子積分計算の並列アルゴリズムを提案する．図 2 に示すように，提案アルゴリズムはスキップされる積分計算にほとんど Block を割り当てることがないため，冗長な上界計算や閾値判定などを削減できる．二電子積分の対称性を利用すると，計算する積分の組み合わせは，Shell のペアをインデックスとした $N(N+1)/2 \times N(N+1)/2$ の上三角行列成分として考えられる．$[ab|ab]$ について事前計算した T_{ab} の最大値をキーとして Shell のペアをソートすることで，閾値判定による Warp divergence を回避できる．各 Block が実行する処理内容を Algorithm 1 に示す．提案アルゴリズムでは，先頭の Thread が代表して Block 毎にスクリーニングすることで閾値判定の回数を削減している．さらに各 Block が，スキップされた Block の最小の列番号をグローバル変数 t として共有・更新することで，不要な Block の割り当てや閾値判定を省略して，有意な積分値を含む行列成分を次々に計算していくことができる．

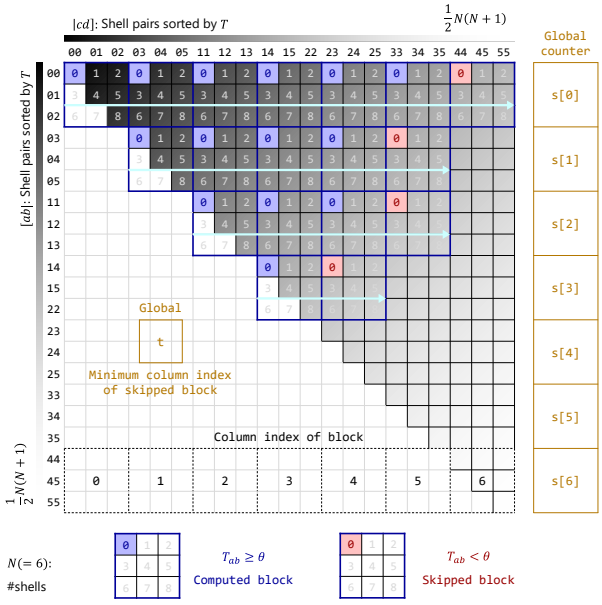

図 2 GPU の Block を動的に割り当てるスクリーニング

Algorithm 1 各 Block が実行する提案アルゴリズム

```
1:  p ← 0
2:  while p < t do
3:      while true do
4:          if i = 0 then
5:              f = 0
6:              q ← p + atomicAdd(s_p, 1)
7:              if q < t then
8:                  if T_ab T_cd < θ then
9:                      atomicMin(t, q)
10:                 else
11:                     f ← 1                    {i: thread index}
12:         if f = 1 then                        {p: row index of block}
13:             Compute [ab|cd]                  {q: column index of block}
14:         else                                 {f: flag shared within a block}
15:             break                            {s_p: global counter initialized with 0}
16:     p ← p + 1                                {t: min. column index of skipped block}
```

5 性能評価

Algorithm 1 の並列実装によるスクリーニングを用いて，s 軌道と p 軌道までの原子軌道を含む系を入力として NVIDIA A100 GPU における二電子積分計算の実行時間を計測し，その積分結果を用いてハートリーフォック法を実行して得られたエネルギーを確認した．図 2 の上三角行列は $[ss|ss], [ss|sp], [ss|pp], \ldots$ などの積分タイプに分けられるが，ひとつの $[ab|cd]$ の計算に最も時間がかかる $[pp|pp]$ タイプの二電子積分の実行時間を表 1 に示す． 積分計算をスキップする閾値には，

表 1 GPU を用いた二電子積分の実行時間とエネルギー

入力分子 (基底関数)	θ	1T1B [1] [ms]	提案手法 [ms]	エネルギー [hartree]
$C_{10}H_{22}$	0	61.17	48.56	−390.545338827
(ANO-DK3)	10^{-12}	27.11	14.63	−390.545338824
$C_6H_{12}O_6$	0	51.78	51.93	−680.948896198
(STO-6G)	10^{-12}	25.65	19.65	−680.948896195
$C_{12}H_{10}N_2$	0	120.19	104.40	−567.593937590
(STO-6G)	10^{-12}	51.20	30.03	−567.593937589

$\theta = 1.0 \times 10^{-12}$ を用いた．提案手法は，積分値の上界の因数 T_{ab} で Shell のペアをソートしない静的な Thread 割り当ての既存手法 1T1B に対して，スクリーニングしたときの実行時間を最大 46%削減した．また，スクリーニングによる高速化効果については，$C_{12}H_{10}N_2$ において既存手法は 2.35 倍の高速化であるのに対して，提案手法は 3.48 倍の高速化を達成している．

6 おわりに

電子間の相互作用を考慮する計算量が大きい二電子積分計算について，動的に GPU の Block を割り当てるスクリーニング手法を提案した．提案手法は，冗長な積分値の上界計算や閾値判定を排除することで，スクリーニングを用いた p 軌道に関する二電子積分計算の実行時間を既存手法に対して最大 46%削減した．

参考文献

[1] 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.

[2] Yingqi Tian, Bingbing Suo, Yingjin Ma, and Zhong Jin. Optimizing two-electron repulsion integral calculations with McMurchie-Davidson method on graphic processing unit. *The Journal of Chemical Physics*, Vol. 155, p. 34112, 07 2021.

[3] Martin Head-Gordon and John A. Pople. A method for two-electron Gaussian integral and integral derivative evaluation using recurrence relations. *The Journal of Chemical Physics*, Vol. 89, No. 9, pp. 5777–5786, 11 1988.

[4] Hans Horn, Horst Weiß, Marco Häser, Michael Ehrig, and Reinhart Ahlrichs. Prescreening of two-electron integral derivatives in SCF gradient and hessian calculations. *Journal of Computational Chemistry*, Vol. 12, No. 9, pp. 1058–1064, 1991.