

二電子 Fock 行列の原始基底計算アルゴリズムの提案

○今村憲亮¹, 本田宏明^{2,3}, 稲富雄一^{1,3}, 南里豪志^{2,3}
(九大シス情¹, 九大情基センター², JST-CREST³)

【はじめに】

京を代表とする大規模な並列計算機は、相互結合網内の多数の計算ノード、各計算ノード内の複数のプロセッサ、プロセッサ内メニーコア、各コア上の SIMD 演算器といった多段の階層から構成されている。そのため、各階層のハードウェア構成を考慮したプログラム実装が必要となるが、本研究では特に SIMD 演算器の有効利用に着目する。

SIMD 演算を利用することで複数の同一演算を単一命令で処理することが可能となる。今後開発されるメニーコアプロセッサにおいて、SIMD 演算器における同時演算可能な数であるウェイ数が 16 から 32 へと拡大することが予想されている。そのため、プログラムの高速化における効率的な SIMD 演算の重要性はより一層高まっている。

しかしながら、現在の分子軌道法計算プログラムでは、そのボトルネックとなる二電子 Fock 行列 (\mathbf{G} 行列) 計算において効率的な SIMD 演算が難しいことが知られている。そこで本研究では、 \mathbf{G} 行列計算における効率的な SIMD 演算の利用法を提案し、SIMD 演算率及び計算高速化に関して性能評価を行うことを目的とする。関連研究としては、GPGPU を利用した高速化に関する研究 [1] や複数入力並列計算による SIMD 演算器の効率的利用に関する研究 [2] がある。

【効率的な SIMD 演算に向けたコード変更】

一般に、コンパイラの自動並列化機能を利用したコード中の多重ループ箇所の SIMD 化オブジェクトコードの生成には、最内ループが容易にアンローリング可能であることならびにループボディ部にデータ依存性が少ないことが必要条件となる。しかしながら、一般的な \mathbf{G} 行列計算コードは 8 重ループから構成され、下位 4 重ループ長が上位 4 重ループの変数に依存して 1~6 で変化し、実行時に決まる。またループボディ部では条件分岐を含むデータ依存関係のある複雑な計算が行われている。

これらの問題に対し本研究では、これまでの縮約基底ではなく原始基底に基づく方法を試み SIMD 演算の効率的利用を図る。この際、下位縮約ループをアンローリングし上位シェルループへの組み込みを行うというアプローチをとる。さらに、ループボディのコードに対し配列表記や専用ディレクトリを利用してコード変更を行うことで、より明確にコンパイラへ SIMD 演算箇所を伝える工夫をする。

【予備的性能評価】

本研究では、九大の稲富らにより開発されているフラグメント分子軌道法プログラム (OpenFMO)[3] の一部として開発された、すべて C 言語にて記述された RHF 法プログラムを使用した。

既存の RHF 法プログラムに対する評価のため、アミノ酸 5 種 (Gly, Ser, Glu, Phe, Trp) に対し上記のプログラムを用いた 1 回の繰返しのための RHF 法テスト計算を行った。Intel Compiler 14.0.0 の-O3 最適化での PAPI ライブラリ [4] によるハードウェアカウンタの取得結果から、 \mathbf{G} 行列計算における SIMD 演算率は 5% 以下であった。次に、下位縮約ループのアンローリングを実行し、同様のテスト計算を行ったところ、SIMD 演算率は 1% 以下へ低下し、浮動小数点演算数は 2 倍程度、実行時間では 6 倍程度の増加がみられた (積分タイプごとの浮動小数点演算数および実行時間の増加内訳のアミノ酸 5 種における平均を図 1, 2 に示す)。いずれの結果に対しても、 s , p 関

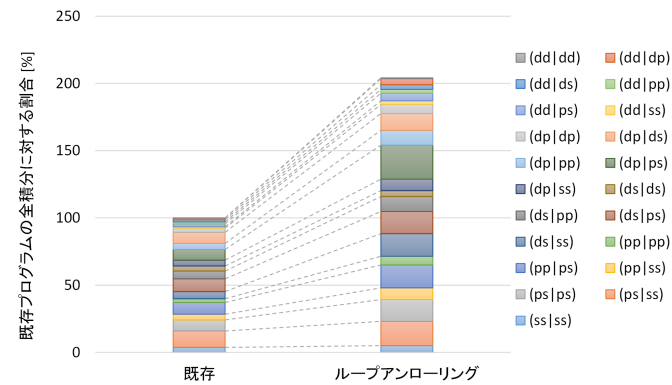


図 1: アミノ酸 5 種における積分タイプ毎の浮動小数点演算数増加内訳平均

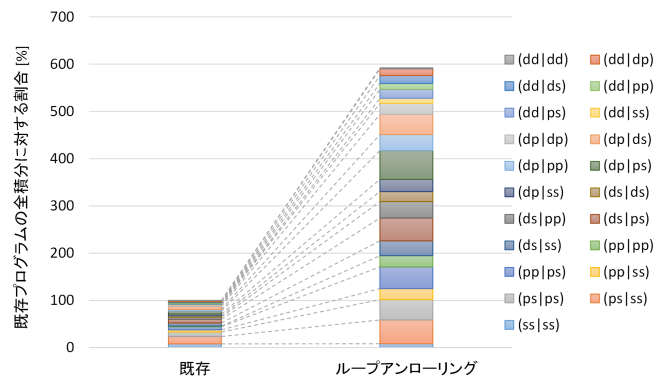


図 2: アミノ酸 5 種における積分タイプ毎の実行時間増加内訳平均

数のみから構成される積分タイプでの増加率は d 関数が含まれる積分タイプに比較し小さいという結果になった。現在のループアンローリングを施したコードに対しさらにループボディの SIMD 向け変更を施すことで、今後開発が予想される 16 ウェイや 32 ウェイの SIMD 演算器を持つプロセッサの利用により性能向上は見込まれるといえるが、現在の 8 ウェイ以下の環境においても可能な限りの性能向上を果たすため、これらの問題に対してより詳細な原因調査を行っている。

当日は、ループアンローリングによる G 行列計算の浮動小数点演算数、実行時間の増加原因の更なる調査の結果ならびにループボディの SIMD 向けコード変更による性能変化の調査結果について報告する予定である。

【参考文献】

[1] 梅田宏明 他, 情報処理学会論文誌 コンピューティングシステム, Vol. 6, No. 4, pp. 26-37 (2013).
 [2] 本田宏明 他, 情報処理学会論文誌 コンピューティングシステム, Vol. 7, No. 4, pp. 15-24 (2014).
 [3] OpenFMO Project (online) available from <http://www.openfmo.org/>.
 [4] Performance Application Programming Interface (online) available from <http://icl.cs.utk.edu/papi/>.