

スクリプト言語を用いた分子軌道法プログラムの開発

(九大情基センター^a, 九大シス情^b, 九州先端研^c, 自宅^d)

本田宏明^a, 稲富雄一^b, 真木淳^c 野呂武司^d

Development of a molecular orbital program using a script language

(RIIT, Kyushu Univ.^a, ISEE, Kyushu Univ.^b, ISIT^c)

Hiroaki Honda^a, Yuichi Inadomi^b, Jun Maki^c, Takeshi Noro

【はじめに】

現在標準的に利用されている種々の分子軌道法計算プログラムは Hartree-Fock 計算から電子相関計算に至るまでの様々な物性計算が可能である。しかしながら全体で数百万行に達するプログラムも有り、内部実装の詳細なドキュメントが公開されていない場合が多い。そのためプログラムソース中にも記述されている各種データのデータ構造や定義箇所、変更箇所やその条件を正確に把握する事が難しく、密度行列やフォック行列といった基本的なデータについても利用方法を独習し自身のアイデアを実装することは困難であることが多い。このような状況において、実装コード量の減少を期待した Python プログラミング言語によるプログラム [1] も既に公開されており、分子軌道法プログラムの実装の学習を促進させる試みも進んではいるが、現状では内部の各種データがユーザーに把握しやすい形で実装されているとまでは言えない。また逐次計算のみに主眼が置かれているようである。

そこで本研究ではスクリプト言語の一つである Ruby を利用し、学習や研究にも利用可能で並列計算も可能な分子軌道法計算プログラムを公開することを目的として、プロトタイププログラムを作成し性能評価を行なった。この際、対話型のプログラム実行もサポートすることで、各種データがプログラム実行中にどのように利用されているかを把握し易くする試みを行なった。

【Ruby 言語を用いたプログラム開発】

Ruby はオブジェクト指向をベースとした動的型付けの汎用プログラミング言語であり、型宣言やデータの動的確保、開放処理を明示的に記述する必要がなく、組み込みの配列クラスや文字列クラス等の数値計算に必要となる定型の処理が予め用意されている。そのため、他のスクリプト言語と同様に Fortran や C 言語のコンパイラ言語に比較し記述に必要なプログラムコード量が大幅に減少するために、より容易に記述することが可能である。

一方、動的型付けのスクリプト言語による実装では通常コンパイラ言語のプログラム実装に比較して種々の原因により実行速度が遅くなる事が知られており、分子軌道法計算の際の性能低下の原因と予想される箇所について、1. 配列利用については数値計算用配列ライブラリ [2] の利用、2. 線型行列計算については GSL [3] や Lapack ライブラリの Ruby からの利用、3. 分子積分計算についてはコンパイル言語による積分ライブラリの作成ならびに Ruby から利用可能なライブラリ化、の工夫を行った。また、RHF 計算における 2 電子フォック行列 (G 行列) 計算では、積分以外の G 行列計算の Ruby 言語での実行時間が計算ボトルネックになる可能性や計算アクセラレータープロセッサの利用を考慮し、G 行列計算全体を C 言語にて実装した G 行列計算ライブラリも用意した。この他、ruby-mpi ライブラリ [4] により Ruby からの MPI 並列計算を可能としている。

現状実装では、デスクサイドコンピュータの RHF/UHF 計算の他、計算機センターの CX400 コンピュータ上における並列の RHF 計算、さらには京コンピュータの商用版である FX10 コンピュータ上においても Ruby による逐次 G 行列計算が可能であることを確認している。

【積分計算ライブラリの開発】

種々の実用的な研究に対応可能とするためには、ガウス型関数の重なり積分や運動エネルギー積分、核引力積分、電子反発積分のみならず、種々の分子積分の実装が必要である。そのため、小原の一般漸化表式の方法 [5] を実装し統一的な方法による積分計算プログラム生成の試みを行った。生成された分子積分プログラムは積分プログラムだけのライブラリとして配布可能である。現状では C 言語実装の積分プログラムを生成しており C や Fortran 言語から利用可能である。これについて Ruby 言語に対するラッパープログラムを作成し、Ruby から呼出し可能な積分ライブラリとした。今後はこの分子積分プログラム生成プログラムを利用して種々の積分プログラムを順次作成する予定である。

【対話型の分子軌道法計算手法の実装】

本プログラムでは分子軌道法プログラムの学習のため対話型処理のサポートを試みた。

コンパイル言語によって記述される RHF 計算を対象としたバッチ処理型プログラムでは、入力や 1 電子フォック行列計算、2 電子フォック行列計算等、計算が連続して実行される。この場合、プログラムを理解するためには、内部の処理の確認に加えて利用しているデータについてもそのデータ構造を理解し、ソースコード中の適切な箇所に出力文を挿入し確認する必要がある。またプログラム実行中にユーザーからデータ内容を変更することは不可能である。

これに対し、プログラムの対話型実行では Mathematica や R 統計解析ソフトウェア等と同様にプロンプトが表示され、入力やフォック行列計算等の各ステップを命令として与え実行することが可能である。また、プログラムの内部を記述している主要なデータをその場でデータ構造を付随して出力することが可能であり、データの変更もプログラム実行中に行う事が可能となるため、各種のデータがどの様に使用されているかの理解が容易になると考える。

この対話型の計算は通常のデスクサイド等のコンピューターにおいて利用可能であり、現状では逐次計算のみサポートしている。

【性能評価テスト】

上述の Ruby 言語にて実装した G 行列計算と C 言語による G 行列ライブラリ実装の 2 種類のプログラムの 1 回の G 行列計算を対象とし MPI による 16 プロセス並列計算までの実行時間を測定した。6-31G 基底関数の (H₂O)₈ をテスト入力とし、Intel Xeon E5-2650 (16 コア) プロセッサ搭載のデスクサイドコンピューターを利用、C 言語プログラムコンパイルには Intel コンパイラ 13.0.1 による O3 最適化ならびに Ruby 1.9.3p429 の測定条件を利用した。今回は静的負荷分散のみの G 行列計算実装である。その結果、逐次計算では Ruby 言語実装が C 言語実装に比較し 7.01 倍遅く、16 並列計算の並列化効率については Ruby 言語 と C 言語実装のそれぞれに対し 85.8% と 92.9% との結果が得られている。

計算機センターでの並列計算等の結果については当日報告する。

【謝辞】

本研究の一部は JST-CREST の研究課題「省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの実装」の支援を受けております。

【参考文献】

- [1] “PyQuante: Python Quantum Chemistry,” [On line]. Available: <http://pyquante.sourceforge.net/>.
- [2] “Numerical Ruby NArray,” [On line]. Available: <http://narray.rubyforge.org/index.html.ja>.
- [3] “GSL - GNU Scientific Library,” [On line]. Available: <http://www.gnu.org/software/gsl/>.
- [4] “Seiya/ruby-mpi,” [On line]. Available: <https://github.com/seiya/ruby-mpi>.
- [5] M.Honda *et al.*, *J.Chem.Phys.*, Vol.94, pp.3790-3804, 1991.