2P087

電子構造計算プロトタイピング環境の開発 – CSF ベース CASSCF モジュールの実装 –

(九大院・システム情報科学[a], JST-CREST[b])

○本田 宏明[a,b], 野呂 武司

A prototyping environment for electronic structure calculations: Implementation of CSF based CASSCF modules

○ Hiroaki Honda[a,b], Takeshi Noro

(ISEE, Kyushu Univ.[a], CREST-JST[b])

【Abstract】

There are many MO programs which can calculate various chemical properties in many chemical accuracy levels. However, it's common that source code sizes of such programs are larger than one-million steps and detailed code documents are not disclosed. To overcome these difficulties, we have been developing a new MO calculation prototyping environment based on the Ruby scripting language. In the present study, we implemented CASSCF modules in our prototyping environment. CAS-CI wave functions are expanded with CSFs and orbital optimizations are based on the Super-CI method. Measured execution times of CASSCF calculations of a H_2O molecule are less than 40 seconds and essential part of CASSCF modules are implemented by 120 code steps. We expect that our prototyping environment can be utilized for helping to understand CASSCF program implementations.

【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では，SCF から電子相関計算に至るまでの様々な物性計算が利用可能である．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のみ記述されている各種データの構造や定義箇所，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実装 [1,2] も既に公開されており，分子軌道法プログラムの実装学習のための書籍も [3] も出版されている．特に PySCF プログラムは，種々の分子積分ライブラリや SCF, MP, MCSCF, CI, CC などの多様な計算が可能であり有用である．しかしながら，CSF ベースの電子相関計算については公開されていないなどの問題もある．これまで我々はスクリプト言語の一つである Ruby を利用し，逐次実行においては SCF ならびに明示的にハミルトニアン行列を生成する CI 計算が可能であり，並列計算については計算機センターでも利用可能な分子軌道法プログラムを開発してきた [4]．特に電子相関計算では，DRT による CSF 生成ならびに Sasaki による CSF ベースの N 電子ハミルトニアン行列計算のエネルギー表式 [5] を採用しており，ユーザが CSF 対からなる行列要素をすべて直接取得可能な形で CI 計算を実行するよう実装している．しかしながらこれまでの実装では，MCSCF 計算をサポートしていないといった問題が有った．

そこで本研究では，これまでの CSF ベースの Second-order CI 計算プログラムを利用した CASSCF モジュールを実装した．軌道の収束方法としては Roos らによる Super-CI 法 [6] をサポートしている．

表 1: CASSCF モジュール実行時間

	次元数	収束回数	実行時間 [sec.]
4in4	20	70	32.9
6in6	175	60	39.2

【CASSCF 法の実装】

本 CASSCF の実装は Roos らによる Super-CI の方法に基づいている．この方法では，一旦 Active 軌道からなる CAS-CI 計算を実行し CASSCF の候補となる波動関数を求める．次にこの候補となる波動関数から 1 電子励起を考慮した新規の一電子励起波動関数 (SX-State または Brillouin State) を計算する．その上で元の候補となる波動関数の密度行列との差分を計算し，収束していない場合には再びこれらの手続きを繰り返す．この CAS-CI 部分の計算については，Sasaki による Tensor-recoupling 法に基づく CSF ベースのハミルトニアン行列エネルギー表式生成プログラムを利用している．また，SX-State 計算のためのハミルトニアン行列計算では 1 電子積分の項のみを考慮している．本プログラムの入力は，電子相関のための Frozen, Core, Active, External による軌道指定ならびに，1,2 電子積分，初期の軌道 (通常は SCF 軌道) の指定である．

表 1 に，本研究で実装した Super-CI 法による CASSCF モジュールの実行時間を示す．H_2O 分子を対象とし，基底関数は DZ，計算で用いた密度行列の収束のための SQCDF 値は 1.0e-8 とした．また， 使用した計算機環境は Intel Core i5-6260U (1.80GHz), 16GB Mem, GCC 4.9.2, Ruby 2.4.1 であり，逐次計算を行なっている．

この結果から，6in6 までの計算では実行時間が 39.2 秒 と，十分利用可能な実行時間であるといえる．また，MCSCF モジュールプログラムの骨格部分は Ruby プログラムで 120 行, Super-CI のためのライブラリ部分は 1082 行であり，おおよそのプログラム実装を理解することは難しくないと考えている．

現在，本 CASSCF モジュールに対し収束を加速させるため Newton-Raphson 法を実装中である．プログラムの開発の進捗ならびに種々の計算例については当日報告する．

【謝辞】

本 CASSCF モジュールのプログラム開発に際し，中京大学の山本茂義教授から CASSCF プログラム JASON2 [7] のご提供ならびに Super-CI 実装に関する種々のご意見を頂きました．本研究の一部は JSPS の科研費 (16K05660) の助成ならびに 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開発」,「ポストペタスケールシステムのためのための電力マネージメントフレームワークの開発」の支援を受けております．

【参考文献】

[1] “HOLTON,” [On line]. Available: ⟨https://github.com/theochem/horton⟩.

[2] “PySCF: Python-based Simulations of Chemistry Framework,” [On line]. Available: ⟨https://github.com/sunqm/pyscf⟩.

[3] 日野，実践 量子化学計算プログラミング, アドバンスソフト, 2010.

[4] 本田 他, 分子科学討論会 2016, 2P109.

[5] B.O.Roos *et al.*, *J.Chem.Phys.*, Vol.48, pp.157-173,1980 and references therein.

[6] F.Sasaki *et al.*, *MOTECC-90*, IBM, pp.181-234.

[7] 山本 他, CASSCF 計算のためのプログラムシステム JASON2, 北海道大学計算機センター昭和 62 年度ライブラリ開発報告集 (9), pp.1-18,1988.