計算機アーキテクチャ 117－16
（1996. 3. 5）

超高速分子軌道計算専用機ＭＯＥの開発

長嶋 雲兵[†]　小原 繁[††]　村上和彰[†††]　吉井 卓[†††]　白川 暁[†††]　網崎孝志[††††]
北村一泰[†††††]　高島 一[†††††]　田辺和俊[††††††]
[†]お茶の水女子大学理学部 [††]北海道教育大学釧路校教育学部 [†††]九州大学総合理工学研究科 [††††]
島根大学総合理工学部 [†††††]大正製薬株式会社 [††††††]物質工学研究所

分子軌道計算は、材料化学や医薬品開発のために欠くことのできない手法であり、現在本方法は化学工業においても広く利用されはじめている。分子軌道計算は、基底関数の数Ｎの４乗に比例する演算量および、補助記憶量を必要とするため、実際のタンパク質等の巨大な分子の計算は、事実上不可能であった。そこで、演算時間の大幅な短縮と補助記憶量の削減を目的として、分子軌道計算のための専用計算機MOEとそれを用いた分子軌道計算プログラムの開発を試みている。本報告では、分子軌道計算方法の紹介と、MOEの設計方針について概要を述べる。

Special purpose computer for high-performance molecular orbital calculations: Molecular Orbital calculation Engine(MOE)

Umpei Nagashima[†], Shigeru Obara [††], Kazuaki Murakami [†††],
Satoru Shirakawa [†††], Takashi Yoshii [†††], Takashi Amisaki [††††],
Kunihiro Kitamura [†††††], Hajime Takashima [†††††]
and Kazutoshi Tanabe[††††††]
[†]Ochanomizu University, [††]Hokkaido University of Education, [†††]Kyushu University,
[††††]Shimane University, [†††††]Taisho Pharmaceutical Co. Ltd., [††††††]National Institute of Materials and Chemical Reactions

Moleculer Orbital calculations play an important role for the design of new materials. Today, the method is widely used also in Chemical industry. However, CPU time and Disk space requirements increase proportionally to N^4 where N is the number of basis fanctions. Therefore, it is quite diffiicult to carry out MO calculations for realistic molecules such as proteins. In oder to reduce CPU time and Disk space requirements, we are developing a special purpose computer for MO calculations: the MO calculation Engine (MOE), and new program based MOE. In this paper, the design concept of the MOE is described along with the algorithm of the calculation.

1. はじめに

計算機と情報ネットワークの飛躍的な発展を背景に、計算機シミュレーションによる機能性材料や医薬の設計は、極めて学際的であるにもかかわらず１つの研究分野を形成しつつある。一方、計算機シミュレーションは理論や科学的知識に立脚した分子や物質の設計を可能にするため、化学や医薬品産業の側も計算機シミュレーションの有効性への期待から、その普及とさらなる発展を切望している。数ある分子シミュレーション手法の中でも非経験的分子軌道法は、分子個々の物理的化学的性質と分子間相互作用という物質科学の基礎的知見を提供する。そして、分子動力学シミュレーション等で用いるポテンシャル関数に関する知見を与えて分子集合系の巨視的物性算出を可能にする。従って、非経験的分子軌道法は分子シミュレーション各手法の基盤といっても過言ではない。そのため、国内外において数多くの非経験的分子軌道法プログラムが開発され、海外のいくつかはすでに商品化されて実際の分子設計に利用されるに至っている。非経験的分子軌道法が格段に実行しやすくなったため、分子の電子状態の精密な波動関数を誰もが容易に得ることができるようになった。この波動関数を解析し種々の物理量の期待値を求めることを通して分子系の物理的・化学的な性質や現象について理解を一層深めていくことが広く行われる状況になっている。研究者が独自の解析を行なうことは理論化学者ばかりではなく実験化学者にも可能になって来ている。しかしながら、非経験的分子軌道計算は、基底関数の数ｎの４乗に比例する演算量と補助記憶量を必要とする。そのため、現在のスーパーコンピュータシステムを使っても、演算量およ

びディスク容量の点からたかだか100原子程度の分子系に適用されているにすぎず、まだまだ生命あるいは化学現象の素過程の分子論的解明とそれに立脚した材料や医薬の設計には不十分といわざるを得ない。物質科学のさらなる発展を考えると、非経験的分子軌道計算の飛躍的な高速化と補助記憶容量等の計算コストの低減化は不可欠である。これの解決策は、カスタムLSIの開発を含む専用計算システムの開発が最も現実的であると考えられる。非経験的分子軌道計算の飛躍的な高速化と補助記憶容量等の計算コストの低減化の実現は、物質科学の各分野へ大きなインパクトをあたえる。本研究の目的は、このような非経験的分子軌道計算の超高速化専用デバイスと計算機システムMOEの開発に加えて、MOEを利用することを念頭においた分子軌道計算プログラムの開発である。戦略的には計算優位性を確立して、わが国の材料設計技術等、物質科学研究のレベルを欧米のそれより格段に高めることにある。

2. 非経験的分子軌道計算[1] とMOE

非経験的分子軌道法計算としては最も基本的であり、世界的に最も広く利用されているハートリーフォック(HF)法を用いる。この方法は、1) データ入力、2) 分子積分計算、3) フォック行列の作成、4) フォック行列の対角化、5) 全エネルギー計算からなる。解くべき方程式が非線形方程式であるため、ステップ2)で計算された分子積分をDisk等の補助記憶に蓄積し、答えが一定になるまでステップ3)、4)を反復する。このうち、ステップ2)の演算量は基底関数の数nの4乗に、ステップ3)、4)はnの3乗に比例する。最も計算時間を要するのはステップ2)であり、現状では全実行時間の90%以上を占める。ワークステーションクラスタによる並列計算の経験から、ステップ4)の対角化はホストで行なうこととし、ホストとMOE間の通信量をn^2のオーダにおさえるため、MOEではステップ2)と3)を高速化することを目的とした。つまり、Fock行列の計算式(r,s,t,u = 1 ~ n)は、

$$F_{rs} = T_{rs} + V_{rs} + \sum_t \sum_u P_{tu} \left\{ g_{rstu} - \frac{1}{2} g_{rtsu} \right\} \quad (1)$$

であり、ここでT_{rs}は、電子の運動エネルギー積分、V_{rs}は核と電子の引力積分と呼ばれ、第3項は、電子間の反発を表わす2電子反発項である。(以降F'_{rs}と表わす。)F'_{rs}は密度行列の要素P_{tu}と2電子積分g_{rstu}, g_{rstu}の和の積となる。MOEで計算するのは上式のF'_{rs}である。F'_{rs}のg_{rstu}が分子積分と呼ばれ、n^4個の計算量と補助記憶量が必要となる。計画中のMOEのシステム構成は図1に示すとおりであるが、計算のほぼ全てが積和演算であるので、図中の汎用プロセッサを、分子軌道計算に特化した積和演算器を多重に実装した専用プロセッサとする計画である。MOEが充分高速であれば、分子積分g_{rstu}を反復の度に計算し、Disk容量を必要としない計算の実行が可能となり、Disk容量のみならずDisk I/O時間の削減等、大幅な計算機資源の節約が可能となる。

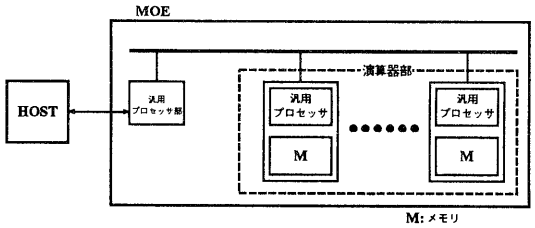


図1 システム構成

まずg_{rstu}の計算方法について説明する。

3. 分子積分

分子積分g_{rstu}の計算方法としては、データ並列性が極めて高く、ベクトル並列計算機のみならず逐次型計算機でも非常に高速に分子積分を計算でき、世界中の非経験的分子軌道計算プログラムで採用されている小原の方法[2]を用いることとした。まず、基底関数について簡単に説明する。

3.1 ガウス関数

通常の非経験的分子軌道計算ではガウス関数φを基底関数に使用する。φの一般形は

$$\varphi = r_x^{n_x} r_y^{n_y} r_z^{n_z} \exp\left[-\zeta \mathbf{r}^2\right]. \quad (2)$$

と書くことができる。

ここで、n_x、n_y、および、n_zは負でない整数でありこれらの和$n_x + n_y + n_z$が軌道量子数になる。これらの整数の組をnと表記し和を|n|と表わすことにする;

$$\mathbf{n} \equiv (n_x, n_y, n_z), \quad (3)$$

$$|\mathbf{n}| \equiv n_x + n_y + n_z. \quad (4)$$

軌道量子数が|n|である関数の成分数$N_{成分}$は

$$N_{成分} = \frac{(|\mathbf{n}|+1)(|\mathbf{n}|+2)}{2} \quad (5)$$

になる。|n| = 0の時、これをs関数と呼び、1の時p、2の時dとそれぞれ呼ぶ。

以上の関数はいずれも座標系の原点の周りの電子を記述するものである。原点以外の座標R

$$\mathbf{R} = (R_x, R_y, R_z) \quad (6)$$

の場合にはガウス関数中の電子座標rをr − Rに置き換えるだけでよい;

$$\mathbf{r} = (r_x, r_y, r_z)$$
$$\Downarrow$$
$$\mathbf{r} - \mathbf{R} = (r_x - R_x, r_y - R_y, r_z - R_z). \quad (7)$$

従って、ガウス関数の一般形φ_Cは

$$\varphi_C(\mathbf{r} - \mathbf{R}; \mathbf{n}; \zeta) \equiv (r_x - R_x)^{n_x} (r_y - R_y)^{n_y} \cdot (r_z - R_z)^{n_z} \exp\left[-\zeta(\mathbf{r} - \mathbf{R})^2\right] \quad (8)$$

になる。

3.2 分子積分計算方式

ワークステーションHP7000/715を用いたトリメチ

	MIDI-4	MIDI-4+P
	117 基底、243pri.	225 基底、351pri.
合計時間 (sec)	2072.720	59345.044
標準 (n^2)注1	113.360(5.5%)	5437.888(9.2%)
変数設定 (n^4)	442.276(21.3%)	1714.726(2.9%)
初期積分	388.124(18.7%)	1966.598(3.3%)
漸化計算	558.980(27.0%)	40873.052(68.9%)
縮約積分	235.132(11.3%)	4101.430(6.9%)
ラベル	334.848(16.2%)	5251.350(8.8%)

注1 n^2 データの計算の他に n^4 ループの空回りの時間も含む。

表1 電子反発積分の計算時間
分子：trimethylcyclohexane(C_9H_{18})

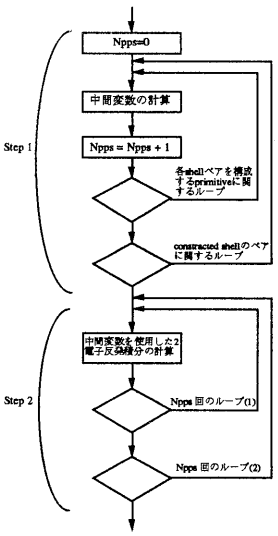


図2 積分計算のアルゴリズム

ルシクロヘキサン $(CH_3)_3C_6H_9$ の分子積分の計算の内訳を表1に示す。トリメチルシクロヘキサンは、ほぼ、タンパク質を構成するアミノ酸1つに対応する大きさの分子である。

表1の左側の基底関数の数nは117基底であり、243プリミティブガウス関数となっている。右側はn=225であり、351プリミティブガウス関数である。nが約2倍に増加することにより、総計算時間が約30倍に増加していることが判る。また、それに伴い、積和計算が大部分を占める漸化計算のステップの演算時間が約30%から約70%へと飛躍的に増加している。

計算アルゴリズムの概要を図2に示す。Step1とは、表1中の標準、変数設定のステップであり、Step2ではそれ以降、初期積分の生成、漸化計算、縮約積分の計算及びラベルの付与のステップが実行される。

MOEでは、全体の計算時間の90%の時間を占める分子積分の計算のうちの70%を占める漸化計算のみならず、本表に表われる全てのステップに加え、残りの10%のうちの3%以上を占めるフォック行列生成の一部をも実行する。つまり、計算時間のうちの約95%程度のステップをMOEで計算する。

パラメータ	パラメータ数	積和演算回数
$1/\zeta, 1/\eta,$ $1/(\zeta+\eta)$	3 (割算3個)	43
$\mathbf{W}=(\zeta\mathbf{P}+\eta\mathbf{Q})/(\zeta+\eta)$	3	9
P−A, P−B, Q−C, Q−D	12	12
W−P, W−Q	6	6
$(\zeta+\eta)^{-1/2}$	1*	32
$\rho=\zeta\eta/(\zeta+\eta)$	1*	2
$T=\rho(\mathbf{P}-\mathbf{Q})^2$	1*	7
Fm(T)	5*	(21 × 5)
計		119+21 × 5 =216

*(2)で使用

表2 漸化計算のために必要なパラメータと積和演算回数

次に図2のStep2における分子積分の計算を例を用いて説明する。

3.3 計算の例

(p_x, p_x, p_x, p_x) を例にとり、2電子積分を計算する場合について具体的に説明する。小原の方法に従って (p_x, p_x, p_x, p_x) を展開すると、以下のような計算が必要になる。

(1) パラメータの計算

Step1で出来るラベルや変数設定のための計算は、すでに行われているとする。必要となるパラメータを表2に示す。表2では、$1/x$、$x^{-\frac{1}{2}}$ が表われているが、これらの計算は、Newton法を用いて積和計算で求めることができる。MOEでは、パラメータ全てを積和演算器で計算することとした。各々の計算に必要な積和演算回数も表2に示してある。

(2) 漸化計算の初期値 $(ss,ss)^m$ の計算

$(ss,ss)^m$ は以下の式を用いて計算される。(詳細は文献[2]を参照)

$$
\begin{aligned}
&[(\mathbf{a}+\mathbf{1i})\mathbf{b},\mathbf{c},\mathbf{d}]^{(m)} = \\
&(Pi-Ai)\times[\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}]^{(m)} \\
&+(Wi-Pi)\times[\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}]^{(m+1)} \\
&+Ni(\mathbf{a})\cdot(1/(2\zeta)\times[(\mathbf{a}-\mathbf{1i})\mathbf{b},\mathbf{c},\mathbf{d}]^{(m)} \\
&+Ni(\mathbf{a})\cdot(1/(2\zeta)\cdot(-\rho/\zeta)\times[(\mathbf{a}-\mathbf{1i})\mathbf{b},\mathbf{c},\mathbf{d}]^{(m+1)} \\
&+Ni(\mathbf{b})\cdot(1/(2\zeta)\times[\mathbf{a}(\mathbf{b}-\mathbf{1i}),\mathbf{c},\mathbf{d}]^{(m)} \\
&+Ni(\mathbf{b})\cdot(1/(2\zeta)\cdot(-\rho/\zeta)\times[\mathbf{a}(\mathbf{b}-\mathbf{1i}),\mathbf{c},\mathbf{d}]^{(m+1)} \\
&+Ni(\mathbf{c})\cdot(1/(2\zeta+2\eta))\times[\mathbf{a},\mathbf{b},(\mathbf{c}-\mathbf{1i}),\mathbf{d}]^{(m+1)} \\
&+Ni(\mathbf{d})\cdot(1/(2\zeta+2\eta))\times[\mathbf{a},\mathbf{b},\mathbf{c},(\mathbf{d}-\mathbf{1i})]^{(m+1)}
\end{aligned}
$$

ここで必要となるパラメータ数と積和演算回数は以下の通りである。

パラメータ	パラメータ数	積和演算回数
m=0 ～ 4	5	3 × 5=15 回

(3) $(pp,pp)^0$ の計算

$(ss,ss)^m$ の計算表式を用いて、$(ss,ss)^{(0\sim4)}$ から $(pp,pp)^{(0)}$ までを求めるときに現れる積和演算の回数は、(1),(2)を使用して、44である。

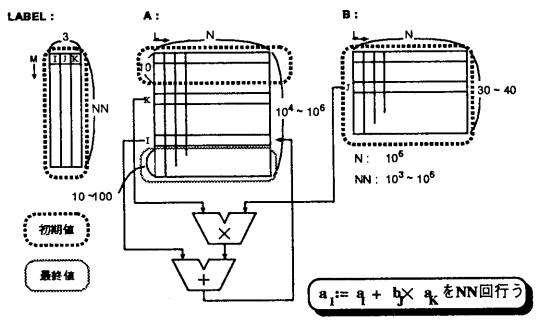


図3 2電子積分

仮数部のビット幅 (bw)	$Fn(T)$の相対誤差(*1)	テーブル容量(*2)
32 ビット	0.12×10^{-4}	256KB
64 ビット	0.29×10^{-14}	512KB
128 ビット	0.16×10^{-33}	1MB

(*1) 桁落ち 12 ビットを仮定し、局所相対誤差 $\epsilon = 2^{-(bw-12)}$ に対して $(1+\epsilon)^{13}$ で計算。テイラー展開による近似誤差は含まない。
(*2)n は 1 刻みで 0 ～ 16、T は 0.05 刻みで 0 ～ 80 の範囲をとるとする

表3 Fn(T) の浮動小数計算の精度と必要なテーブル容量

3.4 積和表による 2 電子積分計算方式

例えば (pp,ss),(ps,ps),(pp,pp) といった積分のタイプ毎に、積和の手順を示すラベルや、初期値などのパラメータの計算も積和演算器で行なうが、積和表方式とは、これら積和の手順を示す LABEL をもとにして、順次中間積分を計算しながら、2 電子積分を求める方式である。図 3に概要を示した。LABEL が積分表であり、1 つの積分タイプの計算で必要なステップ数 NN 個の $a_i := a_i + b_j \times a_k$ という積和演算のための index: i, j, k が格納されている。A,B はそれぞれの分子積分の漸化計算のための初期値とパラメータである。同じ積分タイプでも、2 電子積分に含まれている座標、指数ζの 4 つのペアが異なる N 個の積分計算は並列に実行されるが、1 つの 2 電子積分に関する計算は LABEL に示された手順に従って逐次的に計算される。

$(p_x p_x, p_x p_x)$ の計算時、m は 0 ～ 4 であるから、M=5 であり、積和計算は計 216+15+44=275 個となる。仮に 1 サイクルで 1 回の積和計算が行なわれるとすると、275 サイクルかかることになる。

3.5 回路規模

データのビット幅を 128 ビットと仮定したとき、$F_n(T)$ の計算に必要なメモリ量と演算器のハードウェア規模は表 3のとおりとなる。この他に必要となるメモリは約 75KB であり、内訳はパラメータ +WORK AREA で 1KB、中間積分約 20 個で 320B、テーブル約 73KB である。演算器は約 41 万トランジスタを用いる事で、128 ビット幅の乗算器、加算器 1 個ずつを作ることができる。

この規模の LSI の開発には莫大な資金を必要とするので、データビット長を短くするために、現在必要最小なデータビット長の見積りを行なっている。

次に Fock 行列の行列要素の計算方式について説明する。

4. Fock 行列要素の計算方式

MOE で計算するのは式 1の第 3 項 F'_{rs} で、F'_{rs} を計算する際に密度行列 $P_{tu}(O(n^2))$ は HOST から転送される。また先に説明したように g_{rstu}(2 電子積分) は MOE で計算する。式 1からわかるように F'_{rs} は P_{tu} および g_{rstu} から計算するが、並列計算を行う場合式 1で g_{rstu} の添字が入れ替わっていることを考慮して計算方式を考えねばならない。

例えば図 1の演算器部にある汎用プロセッサ (以下、演算プロセッサ) が n^2 個あるとする。この場合ある F'_{rs} を 1 つの演算プロセッサで計算するのが自然であろう。図 4に示すように、単純にある演算プロセッサが P_{tu} および特定の rs に対する 2 電子積分 g_{rs**} を計算して持っているとする時、1 つの演算プロセッサだけでは F'_{rs} を計算することは不可能で、そのため別の演算プロセッサから必要な g_{rtsu} を送信してもらう必要がある。(図 5) この場合演算プロセッサどうしの転送が頻繁に起こってしまう。これは効率の良い F'_{rs} 並列計算には望ましくなく、また 1 演算プロセッサ当たりに必要となるメモリも大きくなってしまう。

したがって演算プロセッサ間の通信量や演算プロセッサ 1 つ当たりに必要となるメモリ量などを考慮して全体の計算時間が短くなるように並列計算方式を決定する必要がある。

そこで rs に関する単純な並列化のかわりに、様々な並列計算機の検討を行なったが、演算量及びメモリ量、通信量の点で、r,s,t,u の 4 つの index のうち、r に対する並列化、が効率が高いことが判る。次に 2 つの案 (X 案、Y 案) について計算方式を説明する。

4.1 X 案

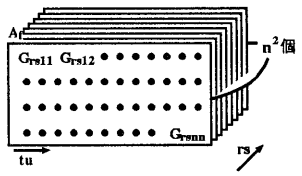
簡単のために演算プロセッサは n 個あるとする。F'_{rs} の添字 r に注目して、個々の演算プロセッサは特定の r に対する計算を行う。以下ある一つの演算プロセッサに対する計算方式の流れを示す。

```
for all r
 seeds of A,B 受信
 for u = 1 ～ n
   $P_{*u}$ 受信
   for t = 1 ～ n
     for s = 1 ～ n
        A_{rstu},B_{rstu} 計算
        for i
           A_{rstui},B_{rstui} の積和
        for end
        g_{rstu} := 積和結果
        F_{rs}' := F_{rs}' + P{tu}*g_{rstu}
```

● 2電子積分

■ A_fと G_{rstu}の関係

(r,s,t,u = 1,2,••• n)



あるrsに対する全てのGがA_fに存在

図4　単純な並列化による分子積分の生成のされ方

● A_f とP' の積和計算

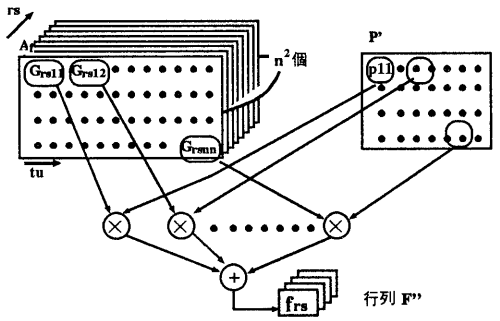


図5　単純な並列化による分子積分を仮定した F'_{rs} の生成

```
        F_{rt}' := F_{rt}' -1/2*P{su}*g_{rstu}
      for end
    for end
  for end
end
```

4.2 Y　　案

ある一つの演算プロセッサに対する計算の流れを示す。

```
for all r
 seed A,B 受信
 for u = 1 ～ n
   for t = 1 ～ n
     for s = 1 ～ n
        P_{tu},P_{su} 受信
        A_{rstu},B_{rstu} 計算
------ 以降 X 案と同様 ------
```

X 案は P_{tu} を u の値によって n 個のグループに分けてまとめて受信するが Y 案はある rstu に対して必要な P の要素を初期値 A_{rstu}, B_{rstu} の計算と一緒に一つずつ受信する。両案の違いはこの 1 点だけである。

4.3 メモリおよび転送量

上で述べた 2 つの案において密度行列 P、各積分タイプ毎の初期値 (A,B) および F'_{rs} の転送量およびメモリ量についての比較を行う。表 4 に P、表 5 に初期値 (A,B)、表 6 に F'_{rs} についてのそれぞれの案で必要な転送量およびメモリ量を示す。表中で特に' 全体の' と書いていない場合は 1 演算プロセッサ当たりの量を表している。

	X 案	Y 案
1 回当たりの転送量	n	2
転送の回数	n	n^3
転送量	n^2	$2n^3$
全体の転送量	n^2	$2n^3$
メモリ量	n	2
全体のメモリ量	n^2	2n

表4　P についてのメモリおよび転送量の比較

	X 案	Y 案
1 回当たりの転送量	n	n
転送の回数	1	1
転送量	n	n
全体の転送量	n	n
メモリ量	n	n
全体のメモリ量	n^2	n^2

表5　初期値 A、B についてのメモリおよび転送量の比較

	X 案	Y 案
1 回当たりの転送量	n	←
転送の回数	1	←
転送量	n	←
全体の転送量	n^2	←
メモリ量	n	←
全体のメモリ量	n^2	←

表6　F'_{rs} についてのメモリおよび転送量の比較

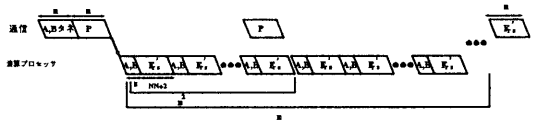


図6　X 案の処理の流れ

表 4 において転送量と全体の転送量が変わらないのは、P が添字 r に依存していない (演算プロセッサは r により特定されるので全ての演算プロセッサで P が共通であることを意味する) ので各演算プロセッサにブロードキャスト可能なためである。Y 案のほうが X 案にくらべ転送量が多く、その分メモリ量が少なくなっている。X 案はこの逆である。

表 5 において X 案および Y 案で転送量と全体の転送量が変わらないのは、A、B の初期データが各演算プロセッサにブロードキャスト可能なためである。

4.4 全体の処理時間

各案において全体の処理時間について大雑把な比較を行う。図 6、7 にそれぞれ X 案、Y 案の処理の流れを示す。

図中の横軸は時間を表す。また通信の場合はデータの個数を、計算の場合は積和の回数を表している。このときデータ 1 個の転送時間と積和 1 回の計算時間を等しいとする。また z は A、B を 1 組 (O(10) 個) 作るのに

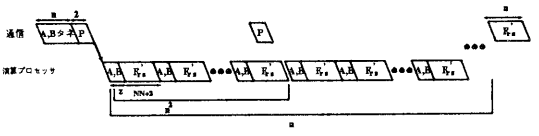


図7　Y案の処理の流れ

必要な演算数を積和演算に換算したものとする。処理時間はそれぞれ

X案: $n+n+n^3(z+NN+2)+n^2$

Y案:$n+2+n^3(z+NN+2)+n^2$

となる。

例えば上式において現在想定している問題サイズを考慮し、全て(pppp)タイプの積分を生成するとすると、$n=1000$(軌道数1000), $z=216$, $NN=44(pppp)$ であり

X案:O(10^{10})、Y案:O(10^{10}) となる。

X案とY案の違いは、Pについてのメモリ量と転送量が違うだけであり、両者のうちX案は、Y案に比べてメモリ量が多いかわりに転送量が少なく、Y案はその逆となっている。メモリ量が多いとコストがかかり、転送量が多いと時間がかかるので、より詳細なアルゴリズムの検討が必要である。

ラベルA, Bのメモリ量をW、また$z+NN=G$とすると本方式におけるおよその計算時間は$n^3(G+2)$で見積もられ、メモリ量は全体で$n(W+n+n^2)$となる。本方式と逐次で行なう場合での必要となる計算機資源の比較を表8に示した。計算コストは両者同じであるが、並列化のため必要メモリ量がほぼn倍に増大している。反復計算毎に分子積分計算を行なうことにし、その反復計算回数を$m(\ll n)$とすると、計算時間はm/nに減少し、補助記憶量はn^4からほぼ0に減少する。このときプロセッサにg_{rstu}が図8のようにn^3個存在し、それがF'_{rs}の生成に影響を与える。

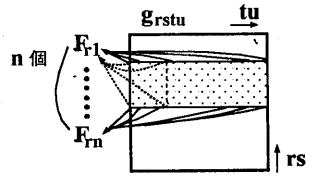


図8　並列化n(r)

方式	計算時間	メモリ量(1プロセッサ)	メモリ量(全体)
逐次	$n^4(G+2)$	$W+2n^2$	$W+2n^2$
n(r)	$n^3(G+2)$	$W+n+n^2$	$n(W+n+n^2)$

表7　逐次とrによる並列化で必要となる計算機資源

5. おわりに

MO(2電子積分計算)のハードウェア化における課題をまとめる。

まず小原法に従って、HF法で使用するフォック行列の一部F'_{rs}を求める時の2電子反発積分(ERI)の計算を行う部分に関するハードウェア上の課題について述べる。

まず誤差に影響する要因として、図2のStep1では、expの計算をテイラー展開した場合の近似誤差, 定数、変数などのデータを浮動小数点表現することによる入力誤差, 浮動小数点演算による丸め誤差(表3)の検討が不可欠である。またStep2では、$Fn(T)=\int_0^1 t^{2n}exp(-Tt^2)dt$ のテイラー展開における6次までの近似(打切り)誤差(10^{-15}), 平方根計算の近似誤差, 定数、変数などのデータを浮動小数点表現することによる入力誤差, 浮動小数点演算による丸め誤差(表3), sのみを含む2電子積分と、p,d軌道などまでを含む2電子積分との演算量と精度の差(表8), などが挙げられる。

また、ハードウェア規模に影響する要因としては、浮動小数点乗除算器、テイラー展開などの係数テーブルに使用するメモリの容量などが挙げられる。

軌道＼演算	加算	減算	乗算	除算	平方根(*2)
(ss,ss)	1回	0回	3回	1回	1回
(ps,ss)	4回	1回	5回	2回	1回
(pp,ss)	7回	2回	7回	3回	1回
(pp,ps)	10回	3回	9回	4回	1回
(pp,pp)	13回	4回	11回	5回	1回

(*1)Step1で求められる中間変数を使って$(ss,ss)^{(m)}$を含む積和計算によって2電子積分を計算するとき、式中で最も計算回数が多い項に関して演算回数を求めた。但し、Fn(T)の計算回数は数えていない。

(*2) 実際には近似式が用いられる見込み。

表8　積分計算に使用される演算回数(*1)

謝辞　本研究の一部は、科学技術振興調整費「第一原理計算手法に基づくシミュレーション技術に関する研究」、産業技術情報基盤研究開発「ヘテロジニアスネットワークコンピューティング環境における分子シミュレーションシステムの構築」、提案公募型・最先端分野研究開発「分子集合体の構造：ab initio 分子動力学を用いた高速プログラムの開発」、文部省科学研究費試験研究B「非経験的分子軌道計算の超高速化に関するハードウェア及びソフトウェアの開発」に基づくものである。

参考文献

1) R.McWeeny, "Methods of Molecular Quantum Mechanics", 2nd edition, Academic(1989). P.W.Atkins, "Molecular Quantum Mechanics", 2nd edition, Oxford(1983). A.Szabo and N.S.Ostlund, "Modern Quantum Chemistry", Macmillan(1982). 大野公男、阪井健男、望月祐志訳、"新しい量子化学（上・下）"、東大出版会(1987).

2) S.Obara and A.Saika, J.Chem.Phys. 84,3963-3974(1986). S.Obara and A.Saika, J.Chem.Phys. 89,1540-1559(1988).