

Ab initio ペア近似法による並列分子軌道計算プログラム ABINIT-MP の作成と性能評価

佐藤 智之^{t1} 秋山 泰^{t2,*} 中野 達也^{t3}
上林 正巳^{t4} 北浦 和夫^{t5}

巨大タンパク質を対象とした精密な電子状態の計算を実現するため、*ab initio* 分子軌道法 (*ab initio* MO 法) の枠組みの中でフラグメントペア近似を行う。Kitaura (1999) の「*ab initio* ペア近似」の理論を採用した並列分子軌道計算プログラム ABINIT-MP を作成し、並列計算機上での性能評価を行った。*Ab initio* MO 法の本来の計算量が系の大きさの 4 乗に比例するのに対して、*ab initio* ペア近似法では系の大きさの 2 乗に比例するため、大幅に計算時間を短縮できる。本手法は分子をフラグメントに分割し、フラグメント単位、あるいはフラグメントペア単位で分子軌道計算の大部分を行うため、並列化に適している。ペア近似による計算精度の犠牲は問題にならないことが分かっている。HF/STO-3G 基底関数レベルのテスト計算を Hitachi SR2201 (150 MHz PA-RISC, 256 PU × 256 MB) 上で行ったところ、193 プロセッサを使用することにより、アクチン結合タンパク質 (PDB entry 1VII, 36 アミノ酸残基) に対して 6120 秒で全エネルギーと分子の電荷密度を計算することができた。また、溶媒の効果を考慮するときに重要になる水クラスター 995 分子の計算では、249 プロセッサ使用時に並列化効率 78% を達成することができた。ABINIT-MP の研究成果は WWW 上 (<http://hse.nihs.go.jp/abinitmp/>) で公開している。

Parallelization and Performance Evaluation of Fragment Molecular Orbital Method Program ABINIT-MP

TOSHIYUKI SATO,^{t1} YUTAKA AKIYAMA,^{t2,*} TATSUYA NAKANO,^{t3}
MASAMI UEBAYASI^{t4} and KAZUO KITAURA^{t5}

We have developed a new parallelized molecular orbital program called ABINIT-MP. In order to achieve *ab initio* MO calculation of biological macromolecules like proteins, ABINIT-MP combines a new fast calculation method and a large-scale parallelization technique. The program employs Kitaura's fragment molecular orbital method (1999). With the pair approximation, calculation is reduced from $O(N^4)$ to $O(N^2)$ where N is bases sizes at the problem. Efficient parallelization of the method has been done on ABINIT-MP. With ABINIT-MP we can calculate the total energy and charge density of an actin binding protein (PDB 1VII, 36 amino acids) within only 6120 seconds (on Hitachi SR2201 150 MHz) using 193 processors in parallel. A water cluster with 995 H_2O molecules is calculated using 249 processors, showing very high parallel efficiency of 78%. The research results of ABINIT-MP are available on the WWW server (<http://hse.nihs.go.jp/abinitmp/>).

^{t1} 株式会社富士総合研究所
Fuji Research Institute Corporation
^{t2} 技術研究組合新情報処理開発機構
Real World Computing Partnership
^{*} 現在、電子技術総合研究所
Presently with Electrotechnical Laboratory
^{t3} 国立医薬品食品衛生研究所
National Institute of Health Sciences
^{t4} 生命工学工業技術研究所
National Institute of Bioscience and Human-Technology
^{t5} 大阪府立大学
Osaka Prefecture University

1. はじめに

近年のコンピュータの高速化やアルゴリズムの改良によって最大で 50 残基程度のタンパク質の *ab initio* MO 法計算が行われるようになったが^{1),2)}, *ab initio* MO 法の計算量が系の大きさの 4 乗に比例するため、たとえば 500 残基程度のタンパク質の計算を行うには 50 残基の場合の計算量の $(500/50)^4 = 10000$ 倍をも要することになり、現時点での実現は難しい。このように従来の方法では計算が困難な生体高分子などの巨大分子への適用を目的として、Kitaura らが提唱し

た *ab initio* ペア近似法^{3),4)}に基づいた巨大分子計算プログラムの開発を Nakano らが行った⁵⁾。本研究では、さらに Nakano らのプログラムを MPI ライブラリを用いて並列化し、数百プロセッサ規模の並列計算機を用いることにより、従来は計算速度とメモリ容量の関係から実行が難しかった大規模問題への分子軌道計算の適用を可能としたので、以下にその詳細を報告する。

Ab initio ペア近似法は計算量が系の大きさの 2 乗に比例するため、500 残基のタンパク質の計算量が 50 残基の場合の $(500/50)^2 = 100$ 倍程度で済む。本手法は原理的に並列分散処理に適するので高い並列化効率が期待でき、フラグメント数のプロセッサを用意できれば系の大きさに比例した計算時間で *ab initio* 計算が実行できる。また、ポテンシャルを完全な量子論で計算できること、任意性のあるパラメータをいっさい用いる必要がないこと、分子内と分子間ポテンシャルを同列に計算できること、などの特徴がある。

2. *Ab initio* ペア近似法の定式化

Ab initio ペア近似法は Kitaura により提案された手法である。文献 3), 4) に従い、以下に説明する。分子をフラグメントに分割し、フラグメントとフラグメントペアとの相互作用まで考慮することにより、多体相互作用を含んだ近似になっている。*Ab initio* ペア近似法では、分子を N 個のフラグメントに分割したときのフラグメント I のハミルトニアン H_I を次のように定義する。

$$H_I = \sum_{i \in I} \left\{ -\nabla_i^2 / 2 - \sum_A Z_A / |\mathbf{r}_i - \mathbf{r}_A| \right. \\ \left. + \sum_{J \neq I}^N \int \rho_J(\mathbf{r}') / |\mathbf{r}_i - \mathbf{r}'| d\mathbf{r}' \right\} \\ + \sum_{i \in I} \sum_{i > j \in I} 1 / |\mathbf{r}_i - \mathbf{r}_j| \quad (1)$$

ここで Z_A は核 A の原子番号、 \mathbf{r}_i , \mathbf{r}_A はそれぞれ電子 i , 核 A の位置ベクトル、 $\rho_J(\mathbf{r}')$ はフラグメント J に含まれる電子の位置 \mathbf{r}' での電子密度である。フラグメントペア IJ のハミルトニアン H_{IJ} についても H_I と同様に定義する。

$$H_{IJ} = \sum_{i \in I, j \in J} \left\{ -\nabla_i^2 / 2 - \sum_A Z_A / |\mathbf{r}_i - \mathbf{r}_A| \right. \\ \left. + \sum_{K \neq I, J}^N \int \rho_K(\mathbf{r}') / |\mathbf{r}_i - \mathbf{r}'| d\mathbf{r}' \right\} \\ + \sum_{i \in I, j \in J} \sum_{i > j \in I, J} 1 / |\mathbf{r}_i - \mathbf{r}_j| \quad (2)$$

分子中のフラグメントとフラグメントペアのハミルトニアン H_I , H_{IJ} は周りのフラグメントからのクーロンポテンシャル項を含んでいる。そこですべてのフラグメントについて電子密度分布が収束するまで式 (1) の計算を繰り返す。こうして得られた $\rho_I(\mathbf{r})$ を用いて

フラグメントペアのハミルトニアン H_{IJ} を計算する。 H_I および H_{IJ} を用いて得られるフラグメントおよびフラグメントペアの全エネルギーを各々 E_I および E_{IJ} とすると、分子の全エネルギー E_{total} は近似的に次のように計算できる。

$$E_{total} = \sum_{I > J} E_{IJ} - (N - 2) \sum_I E_I \\ + \sum_{A > B} Z_A Z_B / |\mathbf{r}_A - \mathbf{r}_B| \quad (3)$$

同様に分子の電子密度 $\rho(\mathbf{r})$ は次のように計算できる。

$$\rho(\mathbf{r}) = \sum_{I > J} \rho_{IJ}(\mathbf{r}) - (N - 2) \sum_I \rho_I(\mathbf{r}) \quad (4)$$

フラグメントペアのエネルギー E_{IJ} は、フラグメント間の最短原子間距離が 10 bohr 以上の場合は各々のフラグメントのエネルギー E_I , E_J から、

$$E_{IJ} = E_I + E_J \\ - \int \int \rho_I(\mathbf{r}_i) \rho_J(\mathbf{r}_j) / |\mathbf{r}_i - \mathbf{r}_j| d\mathbf{r}_i d\mathbf{r}_j \quad (5)$$

を用いて求めることでプログラムの高速化を行う⁵⁾。

3. *Ab initio* ペア近似法の処理の流れ

Ab initio ペア近似法の処理内容を図 1 に示す。以下の手順で計算を行う。

- (1) タンパク質を残基単位に分割してフラグメントと定義する⁵⁾。
- (2) フラグメントの一電子積分計算を行い、フラグメントの初期電子密度を計算する。
- (3) フラグメントの二電子積分計算を行う。必要なメモリはフラグメントの基底サイズの 4 乗とフラグメント数に比例する。
- (4) 各フラグメントの SCF (Self Consistent Field) 計算を行う。計算量はフラグメントの基底サイズの 4 乗とフラグメント数に比例する。
- (5) フラグメントの電子密度分布が収束したら (6) へ、収束していなければ (3) へ。
- (6) 得られたフラグメントの電子密度分布から、フラグメントペア-フラグメント間のクーロンポテンシャルを含むフラグメントペアの一電子積分計算を行う。
- (7) フラグメントペアの二電子積分計算を行う。必要なメモリはフラグメントの基底サイズの 4 乗とフラグメントペア数に比例する。
- (8) フラグメントペアの SCF 計算を行う。全計算時間の半分以上を占める。計算量はフラグメントの基底サイズの 4 乗とフラグメントペア数に比例する。
- (9) 全エネルギー、電子密度を計算する。

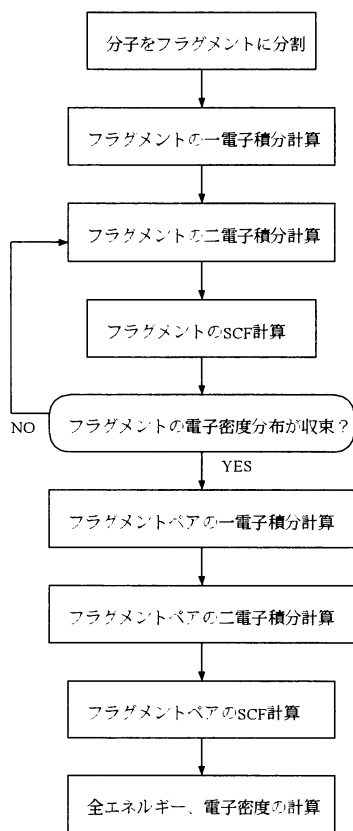


図1 *Ab initio* ペア近似法の処理内容。各フラグメントの電子密度分布を求めた後、フラグメントペアの処理を行う。

Fig. 1 Flowchart of fragment molecular orbital method. We obtain fragment charge density and use it to calculate fragment pair process.

4. 並列化と性能評価

4.1 フラグメント分割による単純な並列化

Ab initio ペア近似法における計算は、フラグメントループ、あるいはフラグメントペアループ単位で行うことによって簡単に並列化することができる。このうち、フラグメントループの並列化の概念を図2に示す。ただしこのような単純な並列化だけでは、4.2節で述べるように台数効果が上がらないという問題点がある。

8残基の Poly Gly (原子数 59) を 1残基 1フラグメントに割り当て、基底関数 STO-3G を用いて計算したときの計算時間と速度向上比の台数効果を表1に示す。C末端をカルボキシル基でキャップし 1フラグメントとカウントしているため、フラグメント数は 9、フラグメントペア数は $9 \times (9 - 1) / 2 = 36$ である。本論文の ABINIT-MP の性能評価はすべて Hitachi SR2201 (150 MHz PA-RISC, 256 PU \times 256 MB) 上で行っ

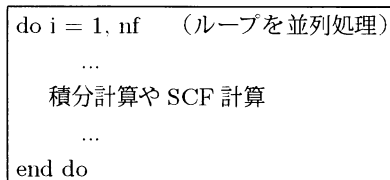


図2 フラグメントループの分割による単純な並列化。フラグメント内の積分計算や SCF 計算はフラグメント単位で行われるため、フラグメント数 nf のループ (do $i=1, nf$) を並列化することができる。

Fig. 2 Simple parallelization of fragment loops. It is possible to parallelize fragment loops (do $i=1, nf$) where nf is number of fragments, for integration and SCF calculations.

表1 Poly Gly ($N = 8$) の計算時間と速度向上比の台数効果。フラグメント数 9、フラグメントペア数 36 の計算モデルのため、フラグメントループ計算は 9 プロセッサ、フラグメントペアループ計算は 36 プロセッサまで性能向上が見られた。

Table 1 Elapsed time and speedup ratio for calculation of Poly Gly ($N = 8$) with each number of processors. It is effective to use until 9 processors to calculate for fragment loops and 36 processors to calculate for fragment pair loops, respectively, because of number of this target model's fragments and fragment pairs.

PU 数	1	3	9	36
Elapsed time (秒)	2239.1	924.7	427.7	212.6
Speedup ratio	(1.0)	2.4	5.2	10.5

た。SR2201 は疑似ベクトル処理機構を持った分散メモリ型スカラ並列計算機であり、各々のプロセッサはプロセッサ間通信速度 300 MB/s の 3次元ハイパクロスパネットワークにより接続されている。次に掲げる 4つの処理に対して並列化を行った。

- フラグメントの二電子積分計算
- フラグメントの SCF 計算
- フラグメントペアの二電子積分計算
- フラグメントペアの SCF 計算

フラグメントループ計算の速度向上は 9 プロセッサまでで、それ以上は台数効果がない。一方、フラグメントペアループ計算の速度向上は 36 プロセッサまでである。

最もメモリ使用量の多い二電子クーロン積分の値を保存するデータ領域は、フラグメントの基底数を nao 、フラグメント数を nf とすると、

$$\text{必要メモリ (Byte)} = 4nao^2(nao + 1)^2 \times nf(nf - 1)/2 \quad (6)$$

となる。前半部分の $4nao^2(nao + 1)^2$ が 1つのフラグメントペアの計算に必要なメモリ量であり、後半部分の $nf(nf - 1)/2$ がフラグメントペア数である。フラグメントの基底数を 30、フラグメント数を 9 とす

ると、式 (6) から二電子クーロン積分の値を保存するデータ領域は 120 MB 必要になる。他の積分データや OS もメモリを消費することから、この方法では前記の構成の SR2201 を使用しても 8 残基の Poly Gly の計算が限界であった。

4.2 並列化の問題点

フラグメントループ計算とフラグメントペアループ計算を単純に並列化しただけの場合には、以下に列挙するような問題点が存在する。

- (1) 配列データのコピーを全プロセッサが持つことはメモリ使用量が逐次処理と変わらずに多いこと（データを分散的に保持させれば、制御は複雑になるが、大規模問題に対応可能）。
- (2) フラグメントの分割を増やせない例題において、プロセッサが余る傾向があり、余剰プロセッサの有効利用がなされないこと。
- (3) 大きなフラグメントを計算するときに、プロセッサあたりの必要メモリが大きくなること。
- (4) 各フラグメントの計算規模の違いにより、負荷分散効率が悪くなること。

以下に、これらを解決するために ABINIT-MP において採用した方法を述べるとともに、テスト計算を行いプログラムの性能を評価する。

4.3 工夫 1: 配列データの各プロセッサへの分割

フラグメントの二電子積分に必要な配列データのメモリ使用量は、フラグメントの基底数の 4 乗とフラグメント数に比例する。また、フラグメントペアの二電子積分に必要な配列データのメモリ使用量は、フラグメントの基底数の 4 乗とフラグメントペア数に比例する。これらのメモリ使用量の大きな配列データを各プロセッサに分割し、「(1) 配列データのコピーを全プロセッサが持つことはメモリ使用量が逐次処理と変わらずに多いこと」の問題に対応した。以下の 2 つの方法についてテストを行った。

- 方法 1 (データ分割法) :
メモリ使用量の大きな二電子積分データを各プロセッサに分割する。通信量は若干増加するが、ほとんど計算速度を落すことなくさらに大きな計算モデルに対応できる。
- 方法 2 (Semi-direct SCF 法) :
二電子積分データの一部を必要なつど毎回作り直す semi-direct SCF 法を行うことによって、使用メモリ量を大幅に削減する。計算量は増加するが、配列データを各プロセッサに分割することと等しい処理が自動的に行え、プロセッサ間の通信量も少ない。

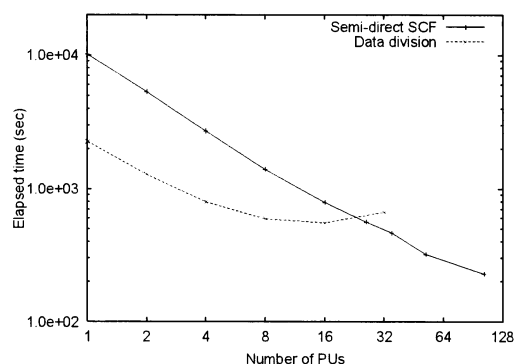


図 3 水クラスター ($N = 103$) の計算時間の台数効果。横軸はプロセッサ数、縦軸は計算時間を表す。実線は semi-direct SCF 法による計算時間、点線はデータ分割を行ったときの計算時間を示す。データ分割を行ったときは少ないプロセッサ数では高速だが、16 台程度で飽和する。Semi-direct SCF 法はプロセッサ数が多くなっても並列化効率が高い。

Fig. 3 Elapsed times for calculation of water cluster ($N = 103$) with each number of processors. It is ineffective to calculate using data division method with more than 16 processors, although it is effective to calculate using semi-direct SCF method if many processors were used.

フラグメントの一電子積分計算とフラグメントペアの一電子積分計算の並列処理を各々の方法に対して追加し、103 分子の水クラスター（原子数 309）の計算時間の台数効果を比較した（図 3）。水分子は球状にランダムに発生させた。水 1 分子を 1 フラグメントとして並列化し、基底関数は STO-3G を用いた。

Semi-direct SCF 法はデータ分割による処理と比較して計算量は多くなるが、プロセッサ数を増やしても並列化効率が非常に高い。式 (5) で近似計算されるフラグメントペアの計算量は少ないので、それ以外のフラグメントペアの計算と分離させて個別に並列化することによって、負荷分散処理の効率が向上する。Semi-direct SCF 法にこの負荷分散処理を追加し、水クラスター ($N = 48, 103, 471, 995$) を計算したときの計算時間と速度向上の台数効果を図 4、図 5 に各々示す。995 分子（原子数 2985）の水クラスターの計算では 249 プロセッサ使用時に 78% という高い並列化効率を得ることができた。

4.4 工夫 2: フラグメント計算の内部処理の並列化と、内包原子数の大きなフラグメントへの対応

前節の方法 1 (データ分割法) や方法 2 (Semi-direct SCF 法) によって大規模な計算が可能になったが、ここではさらに「(2) フラグメントの分割を増やせない例題におけるプロセッサの有効利用」、「(3) 大き

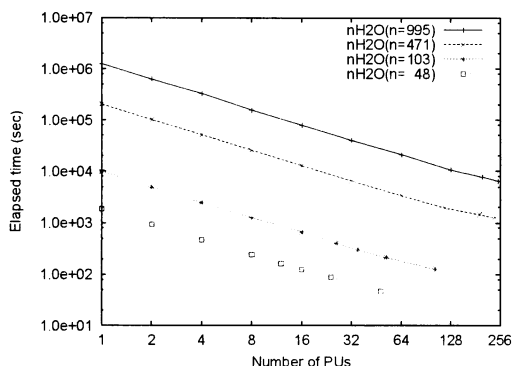


図4 Semi-direct SCF 法による水クラスターの計算時間の台数効果。横軸はプロセッサ数、縦軸は計算時間を表す。上から水分子数 995, 471, 103, 48 のクラスターの計算時間を示している。

Fig. 4 Elapsed times for calculation of water clusters ($N = 995, 471, 103, 48$) using semi-direct SCF method with each number of processors.

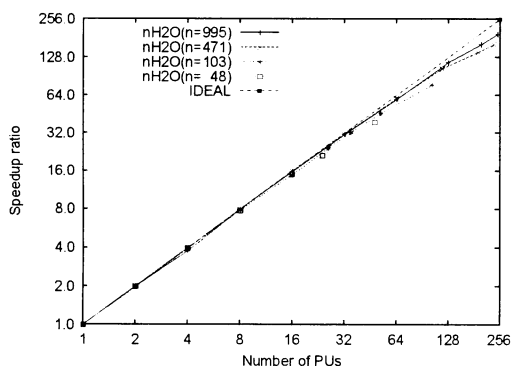


図5 Semi-direct SCF 法による水クラスターの計算速度の向上比。横軸はプロセッサ数、縦軸は計算速度の向上比を表す。水分子数 48~995 の各々の計算速度の向上比は、理想的な直線に近い値を示している。

Fig. 5 Speedup ratios for calculation of water clusters using semi-direct SCF method with each number of processors. Every lines are distributed near the ideal straight line.

なフラグメントを計算するとき、プロセッサあたりの必要メモリが大きくなること」の問題点の解決を図るために、フラグメント内部の計算に並列処理を導入した。

フラグメント内計算を並列化し、必要メモリの大きな配列データを各プロセッサに分割することで上記問題が解決できる。この処理によってさらに大きなモデルの計算が可能となるため、ここでは計算量の多い semi-direct SCF 法を用いず、計算速度の面で有利なデータ分割法に対して実装した。式 (5) による計算を分離する負荷分散処理を追加したときの、20 残基の

表2 Poly Gly ($N = 20$) の計算時間 (秒)。np はフラグメントの内部分割数、PUs は計算に使用しているプロセッサ数を表す。したがって PUs/np はフラグメント単位の分割数を示している。太文字の値はプロセッサのメモリを増設したときの計算時間 (本文参照)、× はメモリ不足のため計算不可能であったことを示す。

Table 2 Elapsed times (sec) for calculation of Poly Gly ($N = 20$). np shows the number of parallelization of inner fragment calculation and PUs shows the number of total used processors. Then PUs/np means number of parallelization of fragment and fragment pair loops. The bold numbers indicate the calculation times that used large memory (see text for details). × indicates that calculation cannot perform because of memory shortage.

PUs/np	1	3	7	21
np = 1	×	10417	4590	1751
np = 2	15470	5266	2340	912
np = 4	8130	2810	1262	527
np = 8	4628	1625	784	393
np = 16	2892	1130	648	-

表3 Poly Ala ($N = 20$) の計算時間 (秒)。np と PUs は表 2 と同様である。計算規模が Poly Gly のケースより大きいため、np = 2, PUs/np = 1 (2 プロセッサ使用) のケースなどはメモリ不足のため計算できなかった。

Table 3 Elapsed times (sec) for calculation of Poly Ala ($N = 20$). np and PUs are identical with Table 2. It cannot perform the case that np = 2 and PUs/np = 1, etc. because this calculation model is larger than the case of Poly Gly.

PUs/np	1	3	7	21
np = 1	×	24563	10905	4115
np = 2	×	13316	5828	2226
np = 4	21037	×	3208	1290
np = 8	11615	4066	1898	878
np = 16	7244	2656	1415	-

Poly Gly (原子数 143) の計算時間を表 2 に、Poly Ala (原子数 203) の計算時間を表 3 に示す。8 残基の Poly Gly の場合と同様に、C 末端をカルボキシル基でキャップし 1 フラグメントとカウントしているため、フラグメント数は 21 である。表中の np はフラグメント内部の分割数であり、PUs は計算に使用しているプロセッサ数を表す。したがって PUs/np はフラグメント単位の分割数を示している。PUs/np が 21 で np が 8 のケースでは 168 プロセッサを計算に使用していることになる。表中の太文字の値は、SR2201 の 256 プロセッサ中 8 プロセッサのメモリを 256 MB から 1 GB に増設して測定したときのものである。内部分割数を増やすことによって大きなフラグメントの計算に対応できるとともに、計算機のメモリやプロセッサ数に応じた並列処理を行うことができる。各々のモデルの計算時間と速度向上の台数効果を図 6、図 7 に

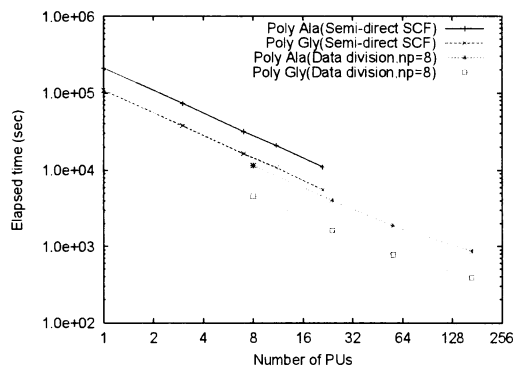


図 6 20 残基のポリペプチドの計算時間の台数効果. 横軸はプロセス数, 縦軸は計算時間を表す. 比較のため semi-direct SCF 法の計算時間も示している. フラグメント内を内部並列化したデータ分割法の計算が高い並列性能を示している.

Fig. 6 Elapsed times for calculation of poly peptides ($N = 20$) with each number of processors. Results of semi-direct SCF method are shown for reference data. Data division method with parallelization of inner procedures shows better performances.

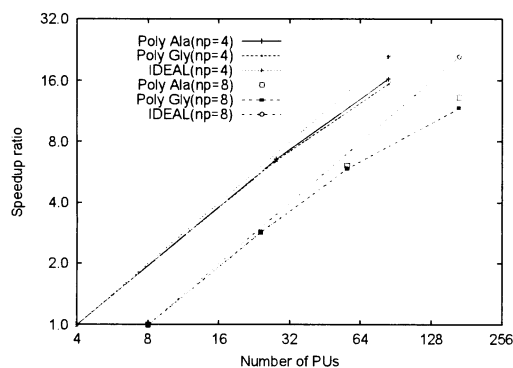


図 7 20 残基のポリペプチドの計算速度の向上比. 横軸はプロセス数, 縦軸は計算速度の向上比を表す. np が 4 のときと 8 のときについて各々 PUs/np = 1 の計算時間を基準に速度向上の台数効果を算出している.

Fig. 7 Speedup ratios for calculation of poly peptides ($N = 20$) with each number of processors. They show speedup ratios relative to PUs/np = 1 in the case of np = 4 and np = 8, respectively.

示す. 比較のため図 6 には semi-direct SCF 法の計算時間も示した. フラグメント内を内部並列化したときのデータ分割法の計算が, 高い並列性能を示している. フラグメント単位の分割数とフラグメント内部処理の分割数をバランス良く指定することによって, 大規模並列処理を高い並列化効率で実行することが可能である. 図 7 では, np が 4 のときと 8 のときについて各々 PUs/np = 1 の計算時間を基準に速度向上比を算出している.

```
do i = 2, nf
  do j = 1, i-1
    (2重ループを動的負荷分散処理)
    ...
    フラグメントペアの SCF 計算
    ...
  end do
end do
```

図 8 動的負荷分散処理を行うフラグメントペアループ. 各フラグメントの計算規模の違いにより負荷分散が悪くなることを避けるため, 全計算時間の半分以上を占めるフラグメントペアの SCF 計算処理を, マスターワーカー方式の動的負荷分散処理により並列化する.

Fig. 8 Load balancing for fragment pair loops. To avoid load unbalancing dependent on fragment size, we use master worker model for fragment pair loops including SCF calculation that occupy more than half of all calculation time.

4.5 工夫 3: フラグメントの大きさが異なるモデルの負荷分散処理

実際のタンパク質では, 同一残基の繰り返しからなるポリペプチドや, 水クラスターなどの例題とは異なり, フラグメントの大きさにばらつきが生じる. 計算粒度の違いによる並列化効率の低下を防ぎ, 「(4) 各フラグメントの計算規模の違いにより, 負荷分散効率が悪くなること」の問題に対応するために, 全計算時間の半分以上を占めるフラグメントペアの SCF 計算処理 (図 8) に対して, マスターワーカー方式の動的負荷分散処理による並列化を行った. テスト計算に利用したタンパク質立体構造データを表 4 に示す.

PDB データ 3AL1 の A 鎖を用いて, 内部分割数 np = 16, 使用プロセス数 $16 \times 7 = 112$ の場合の動的負荷分散処理の効果を確認した. 1 フラグメントに 2 アミノ酸残基を割り当てている. 動的負荷分散処理を行わずに並列化した場合の計算時間が 2145 秒であったのに対して, 動的負荷分散処理を行った場合は 1854 秒に短縮した. 20 残基のポリペプチドの場合, 2 残基を 1 フラグメントに割り当てることによって, 全エネルギーの計算誤差を 2 kcal/mol 程度に抑えることができる⁵⁾. フラグメントの分割方法によって, 得られる計算精度を調節することが可能である. 動的負荷分散処理を実装したプログラムで PDB データ 3AL1 の A 鎖と 1VII の計算時間の台数効果を測定した. 測定の実験結果を図 9 に示す. 内部並列化は 3AL1 を分割数 np = 16, 1VII を np = 32 とした. 1VII の計算では 33 プロセッサ (32 + マスタ) 使用時はメモリ不足であった. 3AL1 はフラグメント数が 7 なので, 内部

表 4 利用したタンパク質立体構造データ。3AL1のA鎖の残基数はアセチル基を1残基と数えている。1フラグメントに2残基を割り当ててフラグメント分割を行った。

Table 4 Protein structure data used.

PDB ID	記述	残基数	フラグメント分割数
3AL1 ⁶⁾ のA鎖	designed polypeptide ALPHA-1	13	7
1VII ⁷⁾	actin binding protein	36	18

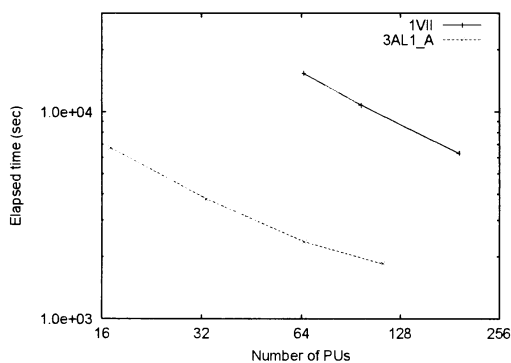


図 9 3AL1のA鎖と1VIIの計算時間の台数効果。横軸はプロセッサ数、縦軸は計算時間を表す。実線は1VII、破線は3AL1のA鎖の計算時間を示している。

Fig. 9 A solid line indicates elapsed time for calculation of 1VII. A broken line indicates elapsed time for calculation of A chain of 3AL1.

分割数 $np = 16$ のときは113プロセッサ ($7 \times 16 +$ マスタ)までが効率的に使用可能である。

5. 考 察

並列化効率を高くするためには、各フラグメントループとフラグメントペアループ単位で並列化を行うことが、プロセッサ間の通信量の面では有利である。しかしプロセッサのメモリ不足から、1フラグメントに数プロセッサを割り当てる必要が生ずる。この場合、フラグメント内部で並列化されていない処理があると大幅に並列化効率を下げてしまう。SCF計算では多数の固有値計算を行っており、固有値計算の計算時間は全体の1~2割程度を占めている。これは内部並列化していない処理の中で最大となっている。フラグメント単位では完全に並列化されているが、フラグメント内部の処理でも並列化する必要がある。行列の3重対角化をHouseholder法で行い、3重対角行列からQL法によって固有値と固有ベクトルを求めている。Householder法は比較的並列化が容易だが、ここでは扱う行列サイズが30~100程度のため、並列化効率を上げるのは難しい。固有値計算の内部並列化を行う前は1VIIの計算時間は193プロセッサ使用時に6345秒であったが、固有値計算の内部並列化によって6120秒に短縮することができた。並列化が容易なアルゴリ

ズムを採用することによって、さらに高速化が期待できる。同様に行列の対角化ルーチンの内部並列化を検討中である。

フラグメントペアのSCF計算の動的負荷分散処理が有効であったことから、フラグメントについてもSCF計算の動的負荷分散処理を行うことによって高速化が期待できる。

36残基のタンパク質が6120秒で計算可能であったため、*ab initio*ペア近似法の計算量が系の大きさの2乗に比例することから、500残基のタンパク質は約2週間で計算できることになる。ただし現システムでの計算はメモリ必要量がフラグメントペア数に比例して増すので難しいが、二電子積分データをまったく保存せずに必要なつど毎回作り直す(full-)direct SCF法を適用し、内部並列処理を実装することによって、メモリ使用量の問題は解決可能である。Direct SCF法を適用した場合、1つのフラグメントペアの二電子積分データが計算可能なメモリさえ用意できれば、問題のサイズを大きくすることによる使用メモリの増加が小さいため、非常に大きなサイズの問題に適用可能になる。1フラグメントに2残基を割り当て、アミノ酸残基がすべてTyrであると仮定した場合、必要メモリは約2.3GBである。今回利用したSR2201の場合、内部分割数 $np = 15$ 程度を指定することによって、メモリ面での問題は解決する。最近のコンピュータの高速化と大容量化を考えると、本論文で述べた手法などの適用により、500残基の*ab initio*分子軌道計算が行われる日も遠くはないと思われる。

6. おわりに

*Ab initio*ペア近似法による並列*ab initio*分子軌道計算プログラムABINIT-MPの作成と性能評価を行った。その際に次のような工夫を導入した。

- 工夫1：配列データの各プロセッサへの分割。
 - 方法1：データ分割法。
 - 方法2：Semi-direct SCF法。
- 工夫2：フラグメント計算の内部処理の並列化と、内包原子数の大きなフラグメントへの対応。
- 工夫3：フラグメントの大きさが異なるモデルの負荷分散処理。

ABINIT-MP の性能を評価するため、さまざまな実用規模の問題を用いて HF/STO-3G 基底関数レベルのテスト計算を Hitachi SR2201 上で行い、高い並列化効率を確認した。フラグメント単位の並列化とフラグメント内の処理の並列化によって、193 プロセッサを使用することにより、36 残基のタンパク質 1VII に対して 6120 秒で全エネルギーと分子の電荷密度を計算することができた。また、溶媒の効果を考慮するときに重要になる水クラスター 995 分子の計算で、semi-direct SCF 法によって 249 プロセッサ使用時に並列化効率 78% を達成することができた。

参考文献

- 1) Alsenoy, C.V., Yu, C.-H., Peeters, A., Martin, J.M.L. and Schäfer, L.: Ab Initio Geometry Determinations of Proteins. 1. Crambin, *Journal of Physical Chemistry*, Vol.A, No.102, pp.2246-2251 (1998).
- 2) Challacombe, M. and Schwegler, E.: Linear scaling computation of the Fock matrix, *Journal of Chemical Physics*, Vol.106, No.13, pp.5526-5536 (1997).
- 3) Kitaura, K., Sawai, T., Asada, T., Nakano, T. and Uebayasi, M.: Pair interaction molecular orbital method: an approximate computational method for molecular interactions, *Chemical Physics Letters*, Vol.312, pp.319-324 (1999).
- 4) Kitaura, K., Ikeo, E., Asada, T., Nakano, T. and Uebayasi, M.: Fragment molecular orbital method: an approximate computational method for large molecules, *Chemical Physics Letters*, Vol.313, pp.701-706 (1999).
- 5) Nakano, T., Kaminuma, T., Sato, T., Akiyama, Y., Uebayasi, M. and Kitaura, K.: Fragment molecular orbital: application to polypeptides, *Chemical Physics Letters*, Vol.318, pp.614-618 (2000).
- 6) Patterson, W.R., Anderson, D.H., DeGrado, W.F., Cascio, D. and Eisenberg, D.: Centrosymmetric bilayers in the 0.75 Å resolution structure of a designed alpha-helical peptide, d,l-Alpha-1, *Protein Science*, Vol.8, No.7, pp.1410-1422 (1999).
- 7) McKnight, C.J., Doering, D.S., Matsudaira, P.T. and Kim, P.S.: Thermostable 35-Residue Subdomain within Villin Headpiece, *Journal of Molecular Biology*, Vol.260, No.2, pp.126-134 (1996).

(平成 12 年 2 月 14 日受付)

(平成 12 年 4 月 20 日採録)

佐藤 智之 (正会員)



昭和 41 年生。平成 3 年東京都立大学大学院理学研究科物理学専攻修士課程修了。同年(株)富士総合研究所入社。伝熱流動解析、構造解析に関連した数値計算プログラムの開発、並列処理業務に従事。現在に至る。平成 10 年より技術研究組合新情報処理開発機構並列応用つくば研究室にて勤務。分子軌道計算プログラムの並列化やペプチドの配座解析に関する研究に従事。平成 12 年より通産省電子技術総合研究所生命情報科学ラボにて勤務。日本生物物理学会会員。

秋山 泰 (正会員)



昭和 36 年生。平成 2 年慶應義塾大学大学院理工学研究科電気工学専攻博士課程修了。工学博士。同年通産省電子技術総合研究所研究官。平成 4 年京都大学化学研究所助教授。平成 8 年技術研究組合新情報処理開発機構並列応用つくば研究室室長。平成 12 年通産省電子技術総合研究所主任研究官(生命情報科学ラボ)。現在に至る。並列計算機を用いたタンパク質立体構造および遺伝子配列情報解析等の研究に従事。電子情報通信学会、日本生物物理学会、分子生物学会、神経回路学会、IEEE 各会員、日本バイオインフォマティクス学会幹事。

中野 達也



昭和 39 年生。平成 4 年筑波大学大学院化学研究科博士課程修了。理学博士。同年厚生省国立衛生試験所(現国立医薬品食品衛生研究所)入所。現在に至る。計算機を用いた生体系の理論的研究に従事。日本化学会、日本薬学会各会員。

**上林 正巳**

昭和48年大阪市立大学大学院修士課程修了。同年通産省工業技術院微生物工業技術研究所（現生命工学工業技術研究所）入所。生命工学工業技術研究所生体物質部主任研究官。

現在に至る。Hyphomicrobium 属細菌の脱窒作用、C1化合物代謝、水素利用生育、ヒドロゲナーゼ、光合成細菌の水素代謝等の研究。馬肝アルコール脱水素酵素光学活性反応モデル、RNA切断酵素反応モデル、trp-represser モデル等の生体反応の計算機シミュレーションの研究に従事。

**北浦 和夫**

昭和51年大阪市立大学大学院理学研究科化学専攻博士課程修了。理学博士。同年日本学術振興会奨励研究員。昭和52年日本学術振興会特定領域奨励研究員。昭和54年分子

科学研究所助手。昭和56年大阪市立大学理学部助手。平成元年岡崎国立共同研究機構分子科学研究所電子計算機センター助教授。平成5年大阪府立大学総合科学部教授。現在に至る。共役系分子の励起状態の理論的研究。遷移金属錯体の構造と反応性の理論的研究。分子間相互作用と分子間ポテンシャルの理論的研究。分子シミュレーションによる水素結合性液体の構造・物性の研究等に従事。