

大規模固有値問題の master-worker 型並列解法

櫻井 鉄也^{†1,†2} 多田野 寛人^{†2,†3} 早川 賢太郎^{†4}
 佐藤 三久^{†1} 高橋 大介^{†1} 長嶋 雲平^{†2,†5}
 稲富 雄一^{†2,†5} 梅田 宏明^{†2,†5} 渡邊 寿雄^{†2,†5}

本論文では、大規模な一般化固有値問題の指定した領域内の固有値と固有ベクトルを求める並列解法について述べる。この方法では複数の大規模な連立一次方程式を解くことで、大規模問題を求めようとする固有値のみを持つような小規模な固有値問題に帰着させる。これらの連立一次方程式は独立に解くことができるため、これらを並列に解く。グリッド RPC システムである OmniRPC 上でこの方法のプログラムを作成した。広域ネットワークを介して PC クラスタを利用する環境において実験を行った。

A Master-worker Type Parallel Method for Large-scale Eigenvalue Problems

TETSUYA SAKURAI,^{†1,†2} HIROTO TADANO,^{†2,†3} KENTARO HAYAKAWA,^{†4}
 MITSUHISA SATO,^{†1} DAISUKE TAKAHASHI,^{†1} UMPEI NAGASHIMA,^{†2,†5}
 YUICHI INADOMI,^{†2,†5} HIROAKI UMEDA^{†2,†5} and TOSHIO WATANABE^{†2,†5}

In this paper we present a parallel method for finding several eigenvalues and eigenvectors of large-scale eigenvalue problems. In this method, a small matrix pencil that has only the desired eigenvalues is derived by solving large sparse linear equations constructed from matrices of the eigenvalue problem. Since these equations can be solved independently, we solve them on remote hosts in parallel. This approach is suitable for master-worker programming models. We have implemented and tested the proposed method in a grid environment using OmniRPC.

1. はじめに

一般化固有値問題 $Ax = \lambda Bx$ において、行列 A 、 B が大規模で疎な場合に、その一部の固有値とそれに対応する固有ベクトルを求める方法について考える。このような問題は、偏微分方程式の解法や構造解析、量子化学などの多くの分野で現れる。

分子軌道計算 (MO 計算) は大規模固有値問題が現れる 1 つの例である。個々の固有値はエネルギー状態に対応しており、必要とする固有値は限られている。Fragment MO 法 (FMO 法)⁴⁾ では MO の固有値の良い近似値を与える。したがって、FMO 法によってあらかじめ求めた固有値を利用することで、MO 計算において必要とする固有値の範囲や近似値を得ることができる。

行列 A 、 B が密行列のときの固有値計算では、行列の相似変換を用いる方法が有効で広く用いられている。しかしながら、これらの方法では計算量は $O(n^3)$ であり、大規模になると計算時間が大きく増加する。

行列が疎な場合には、Lanczos 法⁷⁾ や Arnoldi 法¹⁾ などの Krylov 部分空間に基づく方法^{9),10)} がある。これらの方法では行列とベクトルの積によって部分空間を生成する。多くの場合、この行列とベクトルの積の計算を分散して行うことで並列化をしている。固有値の計算は基本的には非線形問題であるため反復計算が

^{†1} 筑波大学大学院システム情報工学研究科
 Department of Computer Science, University of Tsukuba

^{†2} 科学技術振興機構戦略的創造研究推進事業
 Core Research for Evolutional Science and Technology, Japan Science and Technology Agency

^{†3} 筑波大学大学院博士課程システム情報工学研究科
 Doctoral Program of Systems and Information Engineering, University of Tsukuba

^{†4} 日本システック株式会社
 Nippon Systec Co., Ltd.

^{†5} 産業技術総合研究所グリッド研究センター
 Grid Technology Research Center, National Institute of Advanced Industrial Science and Technology

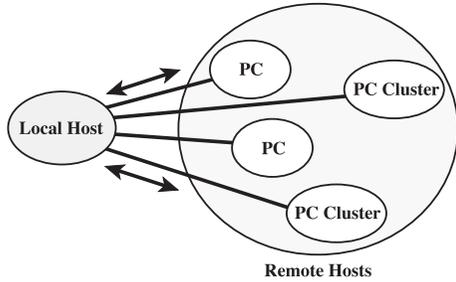


図 1 Master-worker モデルの計算環境

Fig. 1 Master-worker model.

必要となり、この反復の過程でプロセッサ間でのデータの交換が発生する。そのため、図 1 に示すようなローカルホストとリモートホスト間の通信を基本とする master-worker 型のモデルのもとでは並列化の効率が悪くなる。

指定した領域内にある固有値と対応する固有ベクトルを求める方法として、複素平面上の周回積分を用いた方法¹²⁾がある。この方法では、複数の連立一次方程式を解くことで、指定した領域内の固有値のみを持つ小規模な一般化固有値問題に帰着させる。行列が大規模なときには、これらの連立一次方程式の解法が計算の大部分を占め、この部分を並列化することで高速化を図ることができる。このような解法では粗粒度の並列性を持つため、グリッド環境における並列処理に適している。

本論文では、グリッド環境での並列プログラミングのための Grid RPC システム OmniRPC^{8),13)} 上でこの固有値解法を並列化した。OmniRPC は、グリッド環境において遠隔計算機への遠隔手続き呼び出し (RPC: Remote Procedure Call) を用いて、遠隔計算機資源で並列プログラムの実行を可能にするミドルウェアである。また、OmniRPC は master-worker 型の並列プログラミングをサポートし、特にグリッド環境において典型的なアプリケーションであるパラメータ検索などのアプリケーションを効率的にサポートする機能がある。初期化のための大量のデータ転送や計算が必要な場合に、これを再利用することで効率化する自動初期化実行モジュール機能を提供している。グリッド環境の認証には Globus のほか、ssh も利用可能で、ファイアウォールのある遠隔の計算機についても計算資源として利用できるなどの特徴を持つ。

本論文で示す方法は、はじめに行列のデータを各リモートホストに転送すると、その後の計算においてリモートホスト間の通信を必要としない。そのため、グリッド環境において通信による効率の低下を起こしに

くい。分散した計算機環境において数値実験を行い、その効率について検証した。

2. 複素モーメントを用いた固有値解法

行列 $A, B \in \mathbb{C}^{n \times n}$ とし、 $\lambda_1, \lambda_2, \dots, \lambda_n$ は行列束 $A - \lambda B$ の固有値とする。行列束 $A - \lambda B$ は $\lambda \in \mathbb{C}$ に対して $\det(A - \lambda B)$ が恒等的に 0 でないとき正規であるという。

零でない任意のベクトル $u, v \in \mathbb{C}^n$ に対して

$$f(z) := u^H(zB - A)^{-1}v \quad (1)$$

と定義する。 z が固有値のとき行列 $zB - A$ は特異となり、これは関数 $f(z)$ の特異点となる。文献 12) では、式 (1) のように定義した $f(z)$ が $\lambda_1, \lambda_2, \dots, \lambda_n$ を極に持つ有理式で表されることが示されている。この $f(z)$ に対して、文献 5) で示されている有理関数の指定した領域内の極を求める方法を適用する。固有値の分布と方法の性質の解析には文献 6), 11) に示されている方法が利用できる。

Γ を複素平面上の点 γ の周りを正の方向に 1 周する単一閉曲線とし、 Γ 内に m 個の相異なる固有値 $\lambda_1, \lambda_2, \dots, \lambda_m$ があるとする。ここで、複素モーメントを

$$\mu_j = \frac{1}{2\pi i} \int_{\Gamma} (z - \gamma)^j f(z) dz, \quad j = 0, 1, \dots \quad (2)$$

とする。ただし、 i は虚数単位を表す。 $k \times k$ の Hankel 行列 H_k 、および $H_k^<$ を

$$H_k := [\mu_{i+j-2}]_{i,j=1}^k = \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-1} \\ \mu_1 & \mu_2 & \cdots & \mu_k \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-2} \end{pmatrix},$$

および

$$H_k^< := [\mu_{i+j-1}]_{i,j=1}^k = \begin{pmatrix} \mu_1 & \mu_2 & \cdots & \mu_k \\ \mu_2 & \mu_3 & \cdots & \mu_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_k & \mu_{k+1} & \cdots & \mu_{2k-1} \end{pmatrix}$$

とおく。 $k = m$ のとき次の定理を得る¹²⁾。

定理 1 行列束 $A - \lambda B$ は正規であるとする。このとき、行列束 $H_m^< - \lambda H_m$ の固有値は $\lambda_1 - \gamma, \lambda_2 - \gamma, \dots, \lambda_m - \gamma$ である。

したがって、固有値問題 $H_m^< x' = \lambda H_m x'$ を解くことで、もとの問題の固有値 $\lambda_1, \lambda_2, \dots, \lambda_m$ を得ることができる。ここで $x' \in \mathbb{C}^m$ である。もし、 m がそ

れほど大きくないような Γ を与えることができれば、帰着された固有値問題の規模はもとの問題と比べてはるかに小さくなる。

このような Hankel 行列の固有値問題に帰着させる方法の利用例として、トモグラフィがある。文献 2) では、リモートセンシングにおける形状再構成問題を Hankel 行列の固有値問題に帰着させる方法を提案し、モーメントの計算を工夫することで安定性を改善する方法を提案している。

ベクトル q_1, q_2, \dots, q_m を固有値 $\lambda_1, \lambda_2, \dots, \lambda_m$ に対応する固有ベクトルとする。ベクトル s_k を

$$s_k := \frac{1}{2\pi i} \int_{\Gamma} (z - \gamma)^k (zB - A)^{-1} v dz, \quad k = 0, 1, \dots \quad (3)$$

とする。このとき、文献 6) および 12) より固有ベクトルについての次の関係を得る。

定理 2 $m \times m$ の行列 W_m は、行列束 $H_m^< - \lambda H_m$ の固有値 $\lambda_1 - \gamma, \lambda_2 - \gamma, \dots, \lambda_m - \gamma$ に対応した固有ベクトルを列ベクトルに持つとする。このとき

$$[q_1, q_2, \dots, q_m] = [s_0, s_1, \dots, s_{m-1}] W_m \quad (4)$$

である。

$\lambda_1, \lambda_2, \dots, \lambda_m$ の留数を $\nu_1, \nu_2, \dots, \nu_m$ とする。このとき

$$\mu_k = \sum_{j=1}^m \nu_j \lambda_j^k, \quad k = 0, 1, \dots$$

と表される。ここで μ_k は m 個の固有値のみと関係していることに注意する。留数 ν_j は以下の関係式により求められる。

$$(\nu_1, \nu_2, \dots, \nu_m) = (\mu_0, \mu_1, \dots, \mu_{m-1}) W_m. \quad (5)$$

次に、 Γ が円で与えられ、式 (2) の数値積分を台形則によって近似する場合を考える。ここで、 γ と ρ はそれぞれ円の中心と半径とする。 N は正の整数とし、円周上の等間隔点 ω_j を

$$\omega_j := \gamma + \rho e^{\frac{2\pi i}{N}(j + \frac{1}{2})}, \quad j = 0, 1, \dots, N-1$$

とする。

式 (2) の積分を以下のような N 点の台形則で近似することにより、 μ_k の近似値を得る。

$$\mu_k \approx \hat{\mu}_k := \frac{1}{N} \sum_{j=0}^{N-1} (\omega_j - \gamma)^{k+1} f(\omega_j), \quad k = 0, 1, \dots \quad (6)$$

このとき、 $m \times m$ の Hankel 行列 \hat{H}_m と $\hat{H}_m^<$ を $\hat{H}_m := [\hat{\mu}_{i+j-2}]_{i,j=1}^m$ 、および $\hat{H}_m^< := [\hat{\mu}_{i+j-1}]_{i,j=1}^m$ とする。また、 $\zeta_1, \zeta_2, \dots, \zeta_m$ を行列束 $\hat{H}_m^< - \lambda \hat{H}_m$

の固有値とし、 $\hat{\lambda}_j = \gamma + \zeta_j$, $1 \leq j \leq m$ を固有値 $\lambda_1, \lambda_2, \dots, \lambda_m$ の近似値と見なす。

行列 \hat{W}_m の各列ベクトルは $\hat{H}_m^< - \lambda \hat{H}_m$ の固有ベクトルであるとする。ここで、ベクトル y_j を

$$y_j = (\omega_j B - A)^{-1} v, \quad j = 0, 1, \dots, N-1,$$

とし、式 (3) の数値積分を台形則で近似して得られるベクトル \hat{s}_k を

$$\hat{s}_k := \frac{1}{N} \sum_{j=0}^{N-1} (\omega_j - \gamma)^{k+1} y_j, \quad k = 0, 1, \dots \quad (7)$$

とする。このとき、固有ベクトルの近似値 $\hat{q}_1, \hat{q}_2, \dots, \hat{q}_m$ は

$$[\hat{q}_1, \hat{q}_2, \dots, \hat{q}_m] = [\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{m-1}] \hat{W}_m \quad (8)$$

によって得られる。以上をまとめることにより、次のアルゴリズムを得る。

Algorithm:

Input: $u, v \in \mathbb{C}^n$, N, m, γ, ρ

Output: $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_m, \hat{q}_1, \hat{q}_2, \dots, \hat{q}_m$

1. Set $\omega_j \leftarrow \gamma + \rho \exp(2\pi i(j + 1/2)/N)$,
 $j = 0, 1, \dots, N-1$

2. Solve $(\omega_j B - A)y_j = v$ for y_j ,
 $j = 0, 1, \dots, N-1$

3. Set $f_j \leftarrow u^H y_j, j = 0, 1, \dots, N-1$

4. Compute $\hat{\mu}_k, k = 0, 1, \dots, 2m-1$ by (6)

5. Compute the eigenvalues $\zeta_1, \zeta_2, \dots, \zeta_m$ of the pencil $\hat{H}_m^< - \lambda \hat{H}_m$

6. Compute $\hat{q}_1, \hat{q}_2, \dots, \hat{q}_m$ by (8)

7. Set $\hat{\lambda}_j \leftarrow \gamma + \zeta_j, j = 1, 2, \dots, m$

3. 並列化

本章では、前章で示したアルゴリズムの並列化について述べる。 $f(z)$ の $z = \omega_j$ における値の計算は

$$f(\omega_j) = u^H (\omega_j B - A)^{-1} v$$

となる。ここで

$$G_j := \omega_j B - A$$

とおく。前章で示したアルゴリズムにおいて、Step 2、および Step 3 が $f(\omega_j)$ の計算に対応し、 N 個の連立一次方程式

$$G_j y_j = v, \quad j = 0, 1, \dots, N-1$$

を解き、 $u^H y_j$ により $f(\omega_0), f(\omega_1), \dots, f(\omega_{N-1})$ を求めている。

行列 A, B が大規模のときには、この連立一次方程式を解く時間が全体の計算の大部分を占めている。そのため、この複数の連立一次方程式を並列に解くこ

とで高速化を図る．

OmniRPC 上での固有値解法の並列化は、以下のように行った．

- i) あらかじめ A, B, u, v のデータをローカルホストから PC クラスタの各リモートホストに送る．
- ii) N 個の連立一次方程式を非同期遠隔手続き呼び出しを行うことにより、各リモートホストで並列に処理する．
- iii) 各リモートホストの計算結果をローカルホストに集めて固有値、および固有ベクトルを求める．

Step i) において一度行列のデータを各リモートホストに送った後は、Step ii) の実行中にリモートホスト間で通信する必要はない．OmniRPC が利用可能なリモートホストに順次計算を割り振るため、ユーザはプログラム中では単に Step ii) のモジュールをリモートコールするだけでよく、スケジュールの管理などは必要としない．

OmniRPC を用いるときのプログラムソースには以下に示すような記述を行う．

```

. . .
call OmniRPC_Init
call OmniRPC_Module_Init(A,B,u,v,...)
. . .
do j = 0, N-1
. . .
    call OmniRPC_Call_Async('SolveEQ',...)
. . .
end do
call OmniRPC_Wait_All
. . .

```

ここで OmniRPC_Module_Init において、リモートホストが最初に呼び出されたときにのみ行う処理を記述する．本方法では行列のデータをここで送るため、 A や B を引数として指定する．2 回目以降呼び出されたときは、初回に受け取ったデータを利用し、再度行列のデータを送らない．そのため、データ転送のための時間が節約される．

DO ループ中の OmniRPC_Call_Async では、リモートホスト上の連立一次方程式を解くサブルーチンを非同期で呼び出しており、アルゴリズム中の Step 2 に相当する計算を行う．このとき、引数にはリモートホストで実行するサブルーチン名や ω_j などを指定するが、どのリモートホストで実行するかを明示する必要はない．OmniRPC が自動的に空いているリモートホストに処理を割り当てる．

ループ後の OmniRPC_Wait_All において、すべて

のリモートホストでの処理が終わるのを待つ．その後、アルゴリズム中の Step 3 以降の処理を行う．大規模な問題では、Step 2 に要する時間が大半を占め、Step 3 以降の処理に要する時間はわずかである．

本方法は、複数の円が与えられた場合に簡単に拡張することが可能である．ここでは、 N_c 個の円が与えられているとし、 k 番目の円周上の等間隔点を $\omega_j^{(k)}$, $j = 0, 1, \dots, N-1$ とする．このとき $N \times N_c$ 個の連立一次方程式

$$(\omega_j^{(k)} B - A) \mathbf{y}_j^{(k)} = \mathbf{v},$$

$$j = 0, 1, \dots, N-1, k = 1, 2, \dots, N_c$$

を解き、各円ごとに小規模な固有値問題に帰着させればよい．

4. 数値実験

本方法の数値実験例を示す．実験環境として、10 台のリモートホストからなる PC クラスタ (Pentium 4 Xeon 2.4 GHz, dual CPU, Linux, 1000BASE-T 接続, メモリ 1 GB) を用いて、ローカルホスト (Celeron 2.4 GHz, Linux, 100BASE-T 接続, メモリ 2 GB) から実行した．なお、ローカルホストから PC クラスタは、広域ネットワークを経由して接続されている．実験に用いたリモートホストはデュアル CPU であるが、今回の実験ではシングル CPU のみを利用した．

A と B が実行列で、かつ u と v が実ベクトルである場合は、 $f(\bar{z}) = \overline{f(z)}$ の関係が成り立つ．そのため、 $N/2$ 個の関数値 $f(\omega_0), f(\omega_1), \dots, f(\omega_{N/2-1})$ を計算して、 $f(\omega_{N-1-j}) = \overline{f(\omega_j)}$ の関係から残りの関数値を求める．また、ベクトル u , および v の要素は $[0, 1]$ 区間に一様に分布する乱数とした．

4.1 数値例 1

行列 A, B を

$$A = I_n,$$

$$B = \begin{pmatrix} 5 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 6 & -4 \\ & & & & & & 1 & -4 & 5 \end{pmatrix}$$

とした．ここで、 I_n は $n \times n$ の単位行列を表している．また、行列のサイズは $n = 2000000$ とした．この例では固有値は理論的に求めることができ、

表 1 数値例 1 で得られた固有値とその誤差の一部

Table 1 Approximate eigenvalues and their errors in Example 1.

j	λ_j	$ \lambda_j - \lambda_j^* $
1	2.00001216473332	6.2×10^{-15}
2	2.00002572384809	6.2×10^{-15}
3	2.00003928308270	1.2×10^{-14}
4	2.00005284243710	4.8×10^{-14}
5	2.00006640191144	1.6×10^{-14}
6	2.00007996150560	5.3×10^{-15}
⋮	⋮	⋮
366	2.0049692129688	9.8×10^{-15}
367	2.0049828159108	8.9×10^{-15}
368	2.0049964189732	8.9×10^{-16}

表 2 数値例 1 の計算時間と Speedup

Table 2 Wall-clock time in second and speedup in Example 1.

Number of nodes	Time (sec)	Speedup
1	1149	1.00
2	583	1.97
4	304	3.78
6	214	5.37
8	174	6.60
10	150	7.66

$$\lambda_j^* = \frac{1}{16 \cos^4 \left(\frac{j\pi}{2(n+1)} \right)}, \quad j = 1, 2, \dots, n$$

である。

連立一次方程式は帯行列向きの直接解法によって解いた。区間 $[2, 2.005]$ に半径 $\rho = 2.5 \times 10^{-5}$ の円を 100 個配置し、円周上の点数は $N = 32$ とした。Hankel 行列のサイズは $m = 10$ とし、留数の値が $|\nu_j| \geq 10^{-8}$ の固有値を採用した。その結果、100 個の円内にある 368 個の固有値が得られた。得られた一部の固有値の値、および理論値と比較した誤差を表 1 に示す。また、表 2 にリモートホストの台数を 1 台から 10 台まで変えたときの計算時間と Speedup を示す。

この数値例では同じ半径の円を用いているため、一部固有値の誤差が大きくなるものがあり、誤差の最大値は 2.4×10^{-9} であった。また、得られた 368 個の固有値のうち、321 個の固有値の誤差は 1.0×10^{-12} 以下であった。

4.2 数値例 2

A, B は有限要素法を用いた水分子の電子状態計算で現れる固有値問題³⁾ から得られる行列とした。行列 A, B はともに実対称で、行列のサイズは $n = 12173$ 、非零要素数は 509901 である。

表 3 数値例 2 の計算時間と Speedup

Table 3 Wall-clock time in second and speedup in Example 2.

Number of nodes	Time (sec)	Speedup
1	159	1.00
2	82	1.94
4	42	3.79
6	30	5.30
8	24	6.63
10	22	7.57

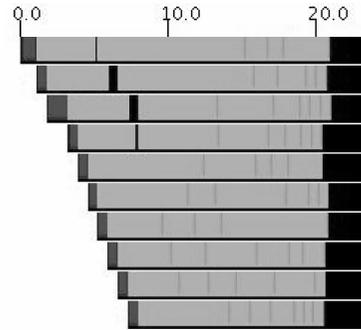


図 2 数値例 2 におけるリモートホストのプロセスの様子
Fig. 2 Timing results of remote hosts in Example 2.

与える円の中心を $\gamma = -6.3, -5.7, -3.5, -2.9, -2.7, -2.0$ とし、半径はすべて $\rho = 0.1$ とした。また、 $N = 32$ 、および $m = 10$ とし、領域内の 15 個の固有値と固有ベクトルを得た。

行列 A, B がともに実対称であるため、 $\omega_j B - A$ は複素対称行列となる。そのため、連立一次方程式は複素対称行列向きの反復解法である COCG 法¹⁴⁾ で解いた。連立一次方程式の前処理は、対角要素のみを計算する不完全 Cholesky 分解を用いた。

表 3 にリモートホストの台数を 1 台から 10 台まで変えたときの計算時間と Speedup を示す。また、図 2 にリモートホストのプロセスの様子を示す。各横棒はそれぞれリモートホスト上でのプロセスを表しており、濃い灰色の部分はデータを転送中であることを示している。薄い灰色の部分は計算中であることを示し、黒は待機状態であることを示している。

連立一次方程式は反復解法で解いているが、パラメータ ω_j の値によって反復回数が大きく異なる場合がある。そのため、リモートホストで計算する時間は均一ではない。ただし、処理の終わったりリモートホストに順に次の連立一次方程式を解かせるため、付加のバランスが調整される。図 2 から分かるように、リモートホストが計算を始めるためにはデータを送り終える必要があり、開始が順に遅くなる。これが Speedup の値が低くなる主要な原因となるため、より多くの台数

表 4 数値例 3 の計算時間と Speedup

Table 4 Wall-clock time in second and speedup in Example 3.

Number of nodes	Time (sec)	Speedup
1	278	1.00
2	143	1.94
4	76	3.66
6	57	4.88
8	50	5.56
10	42	6.62

で処理を行うときにはデータ配信の工夫が必要になると考えられる。

4.3 数値例 3

A, B は分子軌道計算で現れる行列とした。分子はアミノ酸 129 残基の小型のタンパクで、植物性細菌の細胞壁を壊し、殺菌作用のある酵素である。この酵素は、脊椎動物の細胞内や分泌物に広く含まれている。行列 A, B はともに実対称で、行列のサイズは $n = 6005$ 、非零要素数は 1922523 である。

最も高いエネルギー状態 (HOMO), およびそのすぐ上の軌道 (LUMO) を含む区間 $[-3.2, -2.4]$, および $[-1.3, -0.5]$ に半径 $\rho = 0.1$ の円をそれぞれ 4 個配置した。また、パラメータを $N = 32, m = 10$ とし、区間内の 18 個の固有値と対応する固有ベクトルが得られた。

連立一次方程式は前処理付きの COCG 法で解いた。この例では、対角要素のみを計算する不完全 Cholesky 分解では収束しない場合があり、非零要素すべてについて前処理行列の要素を計算する不完全 Cholesky 分解を用いた。このとき、反復に要する時間と比べて前処理行列の計算にかかる時間が大きいため、1 回のリモートホストの呼び出しで連続する 8 個の ω_j を与え、1 つ目の連立一次方程式の解を求めるときに得られた前処理行列をそのまま他の連立一次方程式でも用いるようにした。

表 4 にリモートホストの台数を 1 台から 10 台まで変えたときの計算時間と Speedup を示す。また、この例で用いたタンパク質のフロンティア電子軌道を図 3 に示す。

これらの実験では、円周上の点数 N はすべて 32 としており、共役な値を利用するため解いている連立一次方程式の数は、円 1 個あたり 16 である。求める円のすぐ外に固有値が存在する場合は積分誤差が大きくなるため、誤差を小さくするためには N を大きくとる必要がある。しかしながら、実際には m を大きく設定することによって円の外側の固有値も求めること

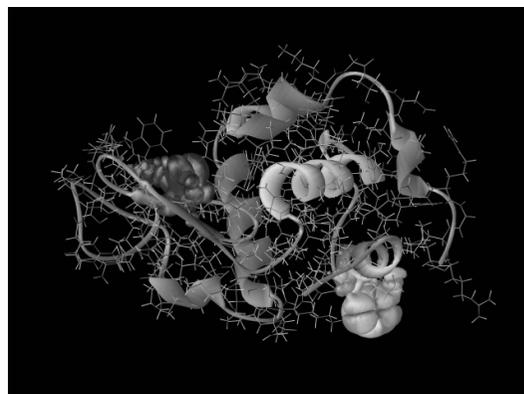


図 3 数値例 3 で用いたタンパク質のフロンティア電子軌道 (HOMO = -0.250067 , LUMO = -0.124338)

Fig. 3 Frontier orbital of the protein in Example 3.

ができ、この場合には積分誤差の影響は小さくなる。これにより、 $N = 32$ でも十分な精度が得られていた。

5. おわりに

本論文では、複素モーメントを用いることで大規模な一般化固有値問題向けの並列解法を示した。また、同解法をグリッド環境での並列プログラミングのための Grid RPC システム OmniRPC 上に実装した。OmniRPC の持つ自動初期化やプロセスの自動割当てを利用することで、容易に並列プログラムが得られた。

本方法はリモートホスト間でのデータ通信を必要としないため、グリッド環境での計算に適しており、広域ネットワークを介して PC クラスタを利用した実験でも高い効率を示した。

より大規模な並列環境でのテストや、実用的な問題での性能評価などが今後の課題である。

謝辞 本研究の一部は科学技術振興事業団の戦略的基礎研究推進事業 (CREST), および文部科学省科学研究費補助金 (基盤研究 (C), 研究課題番号 15560049) の補助を受けて行われた。

参考文献

- 1) Arnoldi, W.E.: The principle of minimized iteration in the solution of the matrix eigenproblem, *Quarterly of Appl. Math.*, Vol.9, pp.17-29 (1951).
- 2) Golub, G.H., Milanfar, P. and Varah, J.: A stable numerical method for inverting shape from moments, *SIAM J. Sci. Comp.*, Vol.21, No.4, pp.1222-1243 (1999).
- 3) Hyodo, S.: Meso-scale fusion: A method for molecular electronic state calculation in inho-

mogeneous materials, *J. Comput. Appl. Math.*, Vol.149, pp.101–118 (2002).

- 4) Inadomi, Y., Nakano, T., Kitaura, K. and Nagashima, U.: Definition of molecular orbitals in fragment molecular orbital method, *Chemical Physics Letters*, Vol.364, pp.139–143 (2002).
- 5) Kravanja, P., Sakurai, T. and Barel, M.V.: On locating clusters of zeros of analytic functions, *BIT*, Vol.39, pp.646–682 (1999).
- 6) Kravanja, P., Sakurai, T., Sugiura, H. and Barel, M.V.: A perturbation result for generalized eigenvalue problems and its application to error estimation in a quadrature method for computing zeros of analytic functions, *J. Comput. Appl. Math.*, Vol.161, pp.339–347 (2003).
- 7) Lanczos, C.: Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Stand.*, Vol.49, pp.33–53 (1952).
- 8) OmniRPC.
<http://www.omni.hpcc.jp/OmniRPC>
- 9) Ruhe, A.: Rational Krylov algorithms for nonsymmetric eigenvalue problems II: Matrix pairs, *Linear Algebr. Appl.*, Vol.197, pp.283–295 (1984).
- 10) Saad, Y.: *Iterative Methods for Large Eigenvalue Problems*, Manchester University Press (1992).
- 11) Sakurai, T., Kravanja, P., Sugiura, H. and Barel, M.V.: An error analysis of two related quadrature methods for computing zeros of analytic functions, *J. Comput. Appl. Math.*, Vol.152, pp.467–480 (2003).
- 12) Sakurai, T. and Sugiura, H.: A projection method for generalized eigenvalue problems, *J. Comput. Appl. Math.*, Vol.159, pp.119–128 (2003).
- 13) Sato, M., Boku, T. and Takahashi, D.: OmniRPC: A Grid RPC System for parallel programming in cluster and grid environment, *Proc. CCGrid 2003*, pp.206–213 (2003).
- 14) van der Vorst, H.A. and Melissen, J.B.M.: A Petrov-Galerkin type method for solving $Ax = b$, where A is a symmetric complex matrix, *IEEE Trans. Magnetics*, Vol.26, No.2, pp.706–708 (1990).

(平成 16 年 10 月 4 日受付)

(平成 17 年 2 月 9 日採録)



櫻井 鉄也 (正会員)

1986 年名古屋大学大学院工学研究科博士課程前期課程修了。同年、名古屋大学工学部助手。1993 年筑波大学電子・情報工学系講師。1996 年同大学電子・情報工学系助教授。現在、筑波大学大学院システム情報工学研究科助教授。独立行政法人産業技術総合研究所客員研究員。博士(工学)。方程式の反復解法と精度保証、大規模固有値問題の並列解法、数理ソフトウェアの利用支援環境の研究に従事。1996 年年日本応用数理学論文賞受賞。日本応用数理学学会会員。



多田野寛人

2001 年山形大学工学部電子情報工学研究科卒業。2003 年筑波大学大学院修士課程理工学研究科修了。現在、同大学大学院博士課程システム情報工学研究科在学中。連立一次方程式の反復解法に興味をもつ。日本応用数理学学会会員。



早川賢太郎

1979 年生。2002 年山形大学工学部電子情報工学科卒業。2004 年筑波大学大学院理工学研究科修了。2005 年同大学院システム情報工学研究科中退。同年、日本システック株式会社入社。並列数値計算に興味をもつ。



佐藤 三久 (正会員)

1959年生。1982年東京大学理学部情報科学科卒業。1986年同大学院理学系研究科博士課程中退。同年新技術事業団後藤磁束量子情報プロジェクトに参加。1991年通産省電子技術総合研究所入所。1996年新情報処理開発機構並列分散システムパフォーマンス研究室室長。2001年より、筑波大学システム情報工学研究科教授。同大学計算科学研究センター勤務。理学博士。並列処理アーキテクチャ、言語およびコンパイラ、計算機性能評価技術、グリッドコンピューティング等の研究に従事。IEEE, 日本応用数学会各会員。



高橋 大介 (正会員)

1970年生。1991年呉工業高等専門学校電気工学科卒業。1993年豊橋技術科学大学工学部情報工学課程卒業。1995年同大学院工学研究科情報工学専攻修士課程修了。1997年東京大学大学院理学系研究科情報科学専攻博士課程中退。同年同大学大型計算機センター助手。1999年同大学情報基盤センター助手。2000年埼玉大学大学院理工学研究科助手。2001年筑波大学電子・情報工学系講師。2004年筑波大学大学院システム情報工学研究科講師。博士(理学)。並列数値計算アルゴリズムに関する研究に従事。1998年度情報処理学会山下記念研究賞, 1998年度, 2003年度情報処理学会論文賞各受賞。日本応用数学会, ACM, IEEE, SIAM各会員。



長嶋 雲兵 (正会員)

1983年北海道大学大学院理学研究科博士後期課程化学第二専攻修了。理学博士。同年, 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手。1992年お茶の水女子大学理学部情報科学科助教授, 1996年同教授を経て, 1998年通産省工業技術院物質工学工業技術研究所基礎部理論化学研究室長。1999年同産業技術融合領域研究所計算科学研究グループ長, 2001年独立行政法人産業技術総合研究所先端情報計算センター情報基盤研究開発室長, 2002年より同グリッド研究センター総括研究員。筑波大学連携大学院大学教授。計算化学, 情報化学, 大規模数値計算, 広域分散並列処理の研究開発に従事。日本化学会, IEEE, 応用数学会, 計算工学会, 化学工学会, 日本コンピュータ化学会各会員。



稲富 雄一

1998年筑波大学大学院化学研究科博士課程修了。博士(理学)。同年同大学化学系技官, 2000年(株)富士総合研究所入社, 2002年独立行政法人産業技術総合研究所特別研究員, 2004年独立行政法人科学技術振興機構戦略的創造事業研究員, 現在に至る。専門分野は理論化学。



梅田 宏明

1970年生。1997年東北大学大学院理学研究科修了。理学博士。科学技術振興機構, CREST研究員。独立行政法人産業技術総合研究所グリッド研究センターにてグリッド向け大規模分子軌道計算プログラムの開発に従事。日本化学会会員。



渡邊 寿雄

2000年筑波大学大学院化学研究科博士課程修了。博士(理学)。同年同大学化学系助手, 2002年科学技術振興機構計算科学技術研究員。2004年同戦略的創造事業研究員, 現在に至る。日本化学会会員。