

ハバードモデルの超大規模固有値問題に対する
地球シミュレータでの並列計算法

日本原子力研究所 山田 進
日本原子力研究所 町田昌彦
電気通信大学 今村俊幸

Parallel solver for Large-Scale Eigenvalue Problems of
Hubbard Model on the Earth Simulator

Susumu Yamada (Japan Atomic Energy Research Institute)
Masahiko Machida (Japan Atomic Energy Research Institute)
Toshiyuki Imamura (The University of Electro-Communications)

1 はじめに

1986年, ベドノルツとミュラーにより, 従来の物性物理の常識を超えた銅酸化物高温超伝導体が発見されて以来 [1], 電子間クーロン反発力が主要な役割を果たす強相関電子系に関する研究が一躍脚光を浴びて来た. しかしながら, 未だ強相関電子系の物性を確実に予測できる強力な理論的手段は確立されておらず, 既知の物理現象さえ, 多くの説明が乱立するといった混乱した状況がしばしば見受けられる. このような現状を打開するため, これまでに数多くの数値計算による研究手法が提案されてきたが, それらにより得られた結果も出版される論文ごとに異なるなど, 多くの問題を抱えており, シミュレーションサイズを大きくし, かつ高精度で計算することが望まれている.

本発表では, 強相関電子系問題において最も高精度な手法である厳密対角化法に焦点を当て, 超大規模なシミュレーションサイズの問題を地球シミュレータで効率良く計算するためのベクトル化・並列化手法について述べ, 地球シミュレータ上で実行した性能を評価する. また, Lanczos法の精度についての考察も行なう. なお, 今回はハバードモデルのエッセンスを含みより現実に近い1次元 d-p モデル問題を例に説明を行なう.

2 強相関電子系問題

2.1 ハバードモデルと d-p モデル

固体中, プラスイオンとなる原子が放出した電子は, 負の電荷を持ち, お互いにクーロン斥力を及ぼしあいながら固体中を運動する. この際, 電子を放出する原子が遷移金属である場合, キャリアとなる電子は d 軌道電子であるため, 局在しやすく同じ軌道に二つ以上の電子が入るために生じるクーロン反発力は s 軌道電子の場合と比べて極めて強くなる. こうして, キャリアとなる電子は非局在化し運動エネルギーの利得を得ようとする一方, 同じ軌道に二つ以上の電子が入ることを嫌うクーロン反発力がせめぎあうため, その電子状態は極めて相関の強い状態となる. こうした系では, 多くの半導体や金属などで成功を収めてきたバンド計算は破綻し, まともに上記のクーロン斥力による相関の強い状態を記述する理論的手法が必要となる.

上記のような電子間反発の強い固体が示す最も典型的な現象は, モット転移と呼ばれる金属 - 絶縁体転移であり, この描像を最も端的に表現するモデルがハバードモデルである. ハバードモデルは

$$H = t \sum_{\langle ij \rangle \sigma} c_{i\sigma}^\dagger c_{j\sigma} + h.c. + U \sum_i n_{i\uparrow} n_{i\downarrow} \tag{1}$$

と表せ, 現在, 強相関電子系のエッセンスを持つ最も単純なモデルとして多くの研究が行われている. ここで, t は, サイト間 $\langle ij \rangle$ のホッピングマトリックス要素であり, 定数で与えられる. また, U は同じ

サイトにスピンの違う電子が二つある場合に発生するクーロン反発エネルギーを表している。銅酸化物超伝導体に代表される遷移金属酸化物ではこのハバードモデルが強い電子相関の基本となっているとする見方がある一方で、酸素の p 軌道と銅などの遷移金属の d 軌道とが混成するという電荷移動の自由度も有しており、これが格子変位の自由度とも絡まりあい、より複雑な電子状態を示すことも議論されている。本研究では、この遷移金属酸化物の物性、ひいては、銅酸化物超伝導体の高温超伝導の解明という最終目標を重視し、より現実的モデルである銅及び酸素の混成を考えた d-p モデルを採用する。尚、本論文で示す並列化手法等は、d-p モデルだけでなく、基本となるハバードモデルのアルゴリズム自体に着目しているため、強相関電子系に共通の手法でもあることを強調する。

d-p モデルは酸化物高温超伝導体の発見により、その超伝導体の電子状態を記述する基本モデルとして提案された。以下、この d-p モデルを手短かに説明する。このモデルのハミルトニアン行列は、

$$H_{d-p} = \sum_{i\sigma} \epsilon_d d_{i\sigma}^\dagger d_{i\sigma} + \sum_{i\sigma} \epsilon_p p_{i\sigma}^\dagger p_{i\sigma} + \sum_i U_d n_{d\uparrow} n_{d\downarrow} + \sum_i U_p n_{p\uparrow} n_{p\downarrow} + \sum_i V [n_{d\uparrow} n_{p\uparrow} + n_{d\downarrow} n_{p\downarrow}] + t \sum_i [d_{i\sigma}^\dagger p_{i\sigma} + d_{i\sigma}^\dagger p_{i-1\sigma} + h.c.] \quad (2)$$

と表せる。ここで、 ϵ_d, ϵ_p は、銅の d 軌道、酸素の p 軌道の軌道エネルギーであり、 U_d 及び U_p は、d 軌道及び p 軌道上でのクーロン反発エネルギーである。この U_d と U_p により、同じサイトに \uparrow の電子と \downarrow の電子が来る (2 重占有) とエネルギーが、各々、 U_d 及び U_p だけ増加するため、電子 (ホール) は 2 重占有をできるだけ避けながらサイト間を渡り歩くことになる。これが、強相関電子系の特徴である。尚、酸素サイトを省略し、 U_d だけをクーロン反発として残したモデルは、ハバードモデルに帰着する [2]。また、 V は d-p 軌道間の電荷移動に纏わるクーロンエネルギーであり、 t は同じく d-p 軌道間のホッピングエネルギーである。 t は、銅サイトと酸素サイトの間の距離が格子系の歪みにより変化するため、動的 (電子系が格子系に比べて素早く応答する場合には静的に扱っても良い) に変化する量であり、銅酸化物超伝導体を含めて遷移金属酸化物では、こうした格子変位も電子系に重要な役割を果たしている可能性が強い [3]。ここでは、将来こうした効果を含めるため、 t がサイトごとに変動しうる可能性を残し、系の並進対称性によるハミルトニアン行列次数の低減は行わない。こうして得られるハミルトニアン行列の次元は 1 次元 d-p モデルの場合、表 1 のような大きさとなる。尚、電子数はアップスピン、ダウンスピンともにサイト数の 4 分の 1 としている。

ところで、表 1 が示すように d-p モデルのサイト数を 4 個増加させると行列の次元は約 2 桁増加する。そのため、必要なメモリサイズも同程度増大する。この必要メモリサイズのサイト数依存性は、並列化しても高々 2, 3 サイトの拡大しか望めないことを意味しており、強相関電子系研究者の並列化に対する意欲を失わせるに十分であった。しかしながら、地球シミュレータ [4] のような 1000 を越えるプロセッサを持ち、テラバイト級のメモリを有する超並列計算機の出現により、物理的に意味あるサイト数の拡大が可能になった。そのため、本論文は並列化を行ない地球シミュレータ規模の計算機を用いることで、これまで以上のサイト数の問題が計算可能になったことを強相関電子系研究者に報告するという性格も併せ持つことを付記しておく。

表 1: サイト数と行列の次元 (1 次元 d-p モデル)

サイト数	行列の次元
12	48,400
16	3,312,400
20	240,374,016
24	18,116,083,216
28	1,401,950,721,600

2.2 ハミルトニアン行列の特徴

本研究で扱う d-p モデルだけでなく、強相関電子系のハミルトニアン行列は、一般に、クーロン反発力項由来の対角項とサイト間トランスファー由来の非対角項とから構成される。これは、クーロン反発力項がアップ及びダウンスピン電子の粒子数演算子の積からなり、状態を変化させないため、ハミルトニアン行列の対角要素となる一方、サイト間トランスファー演算子は電子が違ったサイトに分布する状態間の行列要素、すなわち、非対角行列要素を与えるからである。ここで、電子のサイト間トランスファーに際し、電子のスピン反転が起こらないということに着目すると、非対角要素としては、例えばアップスピンを持つ電子配置に関係するものだけを用意し、それと単位行列との直積を取ることによって全電子配置の非対角成分行列を表すことができる。以上の理由から、強相関電子系のハミルトニアン行列は形式的に

$$H = (I \otimes A) + (A \otimes I) + D \quad (3)$$

と表せる。このとき、 I は A と同じ次元の単位行列、 D は対角行列であり、 A はアップスピン(ダウンスピン)電子のトランスファー演算子に対して非ゼロ要素を与える行列に対応している。そのため行列 A は疎行列であり、またハミルトニアン行列 H も疎行列である。

強相関電子系の電子状態を求めるためには、ハミルトニアン行列 H の全ての固有ベクトル(固有状態)を求めることになるが、物理的にはエネルギーの低い固有状態が低温では支配的な役割を果たすため、実際は最小または最小付近の数個の固有値および固有ベクトルを求めれば良い。そのため、この分野では伝統的に Lanczos 法が用いられている [3][5][6]。Lanczos 法の計算は、ベクトルの内積(ノルム)、ベクトルの線形和、ベクトルの正規化および行列とベクトルの積から成っている。この行列ベクトル積 Hv は式(3)の関係から $(I \otimes A)v + (A \otimes I)v + Dv$ と分解できる。そのため、行列 A および対角行列 D の情報のみを格納しておけば十分であり、行列 H の全ての非ゼロ要素を格納する必要はない。また、このとき対角行列 D およびベクトル v は行列 H の構造から行列 A の次元を n とすると要素数 n の n 個のブロックで構成されていることに注意が必要である。上記の Lanczos 法を構成している計算のうちベクトルの内積、線形和、正規化および行列ベクトル積のうち Dv の計算はそのまま効率的なベクトル計算および並列計算が可能である。そのため、次章以降で $(I \otimes A)v$ および $(A \otimes I)v$ のベクトル化・並列化手法について説明する。

3 ベクトル化

前章で述べたように行列 A は疎行列である。通常、疎行列とベクトルの積を行なう場合、行列の格納形式にベクトル長を長くできる Jagged diagonal storage (JDS) 形式を採用するのが一般的である [7]。JDS 形式を用いた $(I \otimes A)v$ および $(A \otimes I)v$ の計算プログラムは図 1 と表せる。ここで `nzrow`, `jr`, `jc`, `a` および `ipt` はそれぞれ行列 A についての 1 行あたりの非ゼロ要素の個数の最大、JDS 形式の列方向の始まりを表すポインタ、JDS 形式の非ゼロ成分の列位置、JDS 形式の係数、および置換前の行番号である。このとき、どちらの計算も v および w は間接指標アクセスになっており、メモリアクセスは最高速では行なえない [8]。

そこで、最高速のアクセスが可能になる奇数等間隔および連続のアクセスになるようにループの順序を交換し、またブロックサイズ n が偶数の場合は各ブロックの最後に要素を 1 つ追加する [9]。このとき、最内のループの長さはブロックサイズ n になるため、行列 A の格納方法に複雑な JDS を採用する必要はなくなるため、疎行列の一般的な格納法である Compressed row storage (CRS) 形式を採用する。その際の $(I \otimes A)v$ および $(A \otimes I)v$ の計算方法は図 2 となる。ただし、`n`, `ir`, `ic` および `a` はそれぞれ行列 A の次元、CRS 形式の各行の始まりを表すポインタ、CRS 形式の非ゼロ要素の列位置および CRS 形式における

```

do l=1,n
  nn=(l-1)*n
  do i=1,nnzrow
    do j=jr(i),jr(i+1)-1
      k=j-jr(i)+1
      w(nn+ipt(k))=w(nn+ipt(k))+a(j)*v(nn+jc(j))
    enddo
  enddo
enddo

```

a) $(I \otimes A)v$ のベクトル化

```

do l=1,n
  do i=1,nnzrow
    do j=jr(i),jr(i+1)-1
      k=j-jr(i)+1
      nn0=(ipt(k)-1)*n
      nn1=(jc(j)-1)*n
      w(nn0+1)=w(nn0+1)+a(j)*v(nn1+1)
    enddo
  enddo
enddo

```

b) $(A \otimes I)v$ のベクトル化

図 1: JDS形式を用いたベクトル化

```

do i=1,n
  do j=ir(i),ir(i+1)-1
    av=a(j)
    icc=ic(j)
    do l=1,n
      nn=(l-1)*n0
      w(nn+i)=w(nn+i)+av*v(nn+icc)
    enddo
  enddo
enddo

```

a) $(I \otimes A)v$ のベクトル化

```

do i=1,n
  nn0=(i-1)*n0
  do j=ir(i),ir(i+1)-1
    av=a(j)
    nn1=(ic(j)-1)*n0
    do l=1,n
      w(nn0+1)=w(nn0+1)+av*v(nn1+1)
    enddo
  enddo
enddo

```

b) $(A \otimes I)v$ のベクトル化

図 2: ループを交換しCRS形式を用いたベクトル化

係数とする。また $n_0 = n + \text{mod}(n+1, 2)$ とする。このとき、図 2a), b) の最内ループがそれぞれ奇数の等間隔および連続のアドレスの参照になっていることが確認できる。

4 並列化

地球シミュレータは8個のプロセッサで1つのノードを構成しているため、効率の良い並列計算を行なうためには、ノード内では通信を行なわないノード内並列を行なう必要がある。このノード内並列には、地球シミュレータ用の並列化指示オプションを利用した自動並列化機能を利用する。また複数ノードを利用した並列計算には、データをブロック単位に分割し（分割方法の詳細は [9] を参照）、ノード間の通信に MPI (Message Passing Interface) を利用した並列化を行なう。このとき、行列・ベクトル積の計算のうち、 Dv および $(I \otimes A)v$ は D および v の分割方法および行列 A の情報の保存方法から通信を行わず並列計算できる。そのため、 $(A \otimes I)v$ の並列化方法について説明する。

4.1 方法1

行列ベクトル積の並列計算では他のノードが持っている計算に必要なベクトル v を通信し、ベクトル v を再構成し、掛け算を行なうのが一般的である。このとき、行列が疎行列である場合は、図 3 のように計算に必要なデータのみを通信することにより、通信時間を削減を行なう。本研究で対象にしている 1 次元 24 サイトの場合の通信量は表 2 に示す。ノード数が増えるに従い、合計の通信量および通信回数とともに増加していることが確認できる。また $(A \otimes I)v$ の計算量（行列の成分とベクトル成分の積の回数）について表 3 に示す。この表からノードごとの計算量のばらつきが大きいことが確認できる。

また、ベクトルを再構成する際に追加される要素数を表 4 に示す。この表からわかるように、今回の問題では、受信するデータ量が大きく、ベクトルの再構成を行なうためには大量のメモリを必要とする。そのため本研究では、ベクトルの再構成を行わず、他のノードからのデータを受信する都度その値を用い

表 2: 方法 1 の通信量

ノード数	通信回数	通信量 (要素数)	
		合計 ($\times 10^9$)	平均 ($\times 10^6$)
32	226	48.837	216.093
48	368	54.346	147.679
64	524	57.660	110.038
96	834	62.695	75.174
128	1170	66.122	56.514

表 3: 方法 1 の計算量

ノード数	計算量 (積の回数) ($\times 10^9$)		
	最大	最小	平均
32	5.949	4.540	5.317
48	4.002	2.893	3.544
64	3.036	2.070	2.658
96	2.091	1.299	1.772
128	1.563	0.937	1.329

表 4: 追加要素数

ノード数	追加要素数 ($\times 10^9$)			ノード自身の要素数 ($\times 10^9$)
	最大	最小	平均	
32	2.305	0.462	1.526	0.566
48	1.661	0.322	1.132	0.378
64	1.282	0.236	0.901	0.283
96	0.971	0.168	0.653	0.189
128	0.767	0.130	0.517	0.142

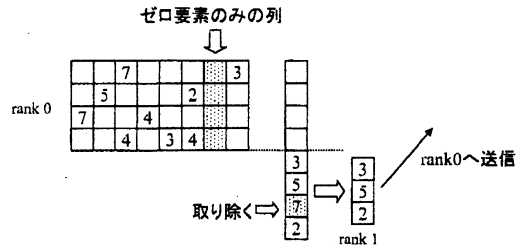


図 3: ベクトルデータの送信方法 (ノード数 2 の場合)

て計算を行ない, 計算後, その値を破棄する方法を採用する. この方法では, 複数のノードからの受信データを同時に格納する必要がないため, メモリの使用量を抑えることができる.

4.2 方法 2

ベクトル v は要素数が n の n 個のブロックで構成されているため, $v = (v_{1,1}, v_{1,2}, \dots, v_{1,n}, v_{2,1}, \dots, v_{n,n})^T$ と表せる. また同様に w を $w = (w_{1,1}, w_{1,2}, \dots, w_{1,n}, w_{2,1}, \dots, w_{n,n})^T$ と表すと, 行列とベクトルの積 $w = (A \otimes I)v$ は

$$W = VA^T \tag{4}$$

と表せる. ただし $V = (v_1^T, v_2^T, \dots, v_n^T)$, $W = (w_1^T, w_2^T, \dots, w_n^T)$ であり, v_i および w_i はそれぞれ $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,n})$, $w_i = (w_{i,1}, w_{i,2}, \dots, w_{i,n})$ である. このとき, 行列 A の情報は全てのノードで持っているため, V を行方向に分割することにより効率的な並列計算が行なえる. しかし, V の値は列方向で分割しているため, データの通信を行なう必要がある. そこで, 次の計算方法を提案する. ここで np および nb をそれぞれ並列計算時のノード数およびそのノードが担当しているブロック数とする.

1. 分散メモリ構造のため rank k ($k = 0, 1, 2, \dots, np - 1$) のノードでベクトル v の値として

$$v^{(k)} = (v_{n_k+1,1}, v_{n_k+1,2}, \dots, v_{n_k+1,n}, v_{n_k+2,1}, \dots, v_{n_k+nb,n})^T$$

を格納している. このとき $n_k = k \times nb$ である. この $v^{(k)}$ の成分 $v_{*,j}$ を rank $\{(j-1)/nb\}$ のノードに送信し,

$$vtmp_k = (v_{1,n_k+1}, v_{1,n_k+2}, \dots, v_{1,n_k+nb}, v_{2,n_k+1}, \dots, v_{n,n_k+nb})^T$$

を作成する.

2. v を行方向に分割した $vtmp_k$ を利用し, 式 (4) を並列計算し

$$wtmp_k = (w_{1,n_k+1}, w_{1,n_k+2}, \dots, w_{1,n_k+nb}, w_{2,n_k+1}, \dots, w_{n,n_k+nb})^T$$

を求める.

3. 得られた wmp_k の値を rank k のノードの w の値が

$$w^{(k)} = (w_{n_k+1,1}, w_{n_k+1,2}, \dots, w_{n_k+1,n}, w_{n_k+2,1}, \dots, w_{n_k+nb,n})^T$$

となるように通信する.

この計算方法において行列とベクトルの積の前後に、 nb^2 個の成分を自身以外の $np-1$ 個のノードに通信する必要がある。今回扱う 1 次元 24 サイトの場合の通信量を表 5 に示す。この表からノード数が増加しても通信の合計はほとんど増加しないことおよび計算量がほぼ均一であることがわかる。しかし、通信回数は増加し、方法 1 と比較し、128 ノードで約 28 倍多い。ところが、一回あたりの平均の通信量が倍精度のデータで 10^6 個以上と多いため、通信時間に関しては通信開始のためのオーバーヘッドより実際に通信を行っている時間の方が支配的であると思われる。また、ノードごとの計算量を表 6 に示す。この表から全てのノードの計算量はほぼ同じであり、計算量の不均衡のない効率的な並列計算が行なえると思われる。そのため、方法 1 よりも通信量の少ない方法 2 の方が優れた並列計算性能になることが予想される。

5 地球シミュレータでの並列計算

5.1 性能評価

以上の強相関電子系に現れる大規模な行列に対する Lanczos 法のベクトル化・並列化の性能を評価するため、地球シミュレータを用いて 1 次元 24 サイト d-p モデル問題に現れる行列の最小固有値およびその固有ベクトルを地球シミュレータを用いて並列計算する。この行列は、行列 A の次元および要素数はそれぞれ 134,596 および 1,264,032 であり、行列 H の次元は 18,116,083,216 である。この問題を 96 ノード (768 プロセッサ) および 128 ノード (1024 プロセッサ) の 2 通りで計算する。このとき、通信するデータは高速に通信ができるグローバルメモリ [11] に割り付ける。また、Lanczos 法により作成される 3 重対角行列の次元は 200 とする。この 3 重対角行列に対する固有値および固有ベクトルの計算は 1 回だけであり、Lanczos 法の計算時間と比較し少ないため、今回は地球シミュレータ用のプログラムは作成せず、日本原子力研究所計算科学技術推進センターで公開している並列数値計算ライブラリ PARCEL[10] の 2 分法および逆反復法のルーチンを利用する。一般的な方法および提案した方法により、最小固有値および固有ベクトルを求めた際の経過時間および相対誤差を表 7 に示す。ここでの誤差の値は $\|\lambda x - Hx\|_2 / \|x\|_2$ を採用する。ただし、 λ および x はそれぞれ求めた固有値および固有ベクトルである。また、行列成分の作成等を含めた全ての計算におけるベクトル演算率、FLOPS 値およびメモリ使用量 (グローバルメモリを利用した場合には正確なメモリ使用量が計測できないためローカルメモリを利用して同様の計算した時のメモリの使用量) を表 8 に示す。さらに、地球シミュレータの簡易性能解析機能 [8] を利用し得られたノード単位の固有値、固有ベクトル計算を行なう際の通信時間 (通信手続きの実行に要した時間、同期のための時間などを含んだ全ての通信時間)、および通信の待ち時間 (通信を行なうまでの待ち時間および同期待ちに要した経過時間の合計) の最大、最小、平均をそれぞれ表 9 および表 10 に示す。

表 5: 方法 2 の通信量

ノード数	通信回数	通信量 (要素数)	
		合計 ($\times 10^9$)	平均 ($\times 10^6$)
32	1984	35.100	17.691
48	4512	35.477	7.863
64	8064	35.666	4.423
96	18240	35.855	1.966
128	32512	35.949	1.106

表 6: 方法 2 の計算量

ノード数	計算量 (積の回数) ($\times 10^9$)		
	最大	最小	平均
32	5.318	5.282	5.317
48	3.546	3.490	3.544
64	2.660	2.584	2.658
96	1.773	1.657	1.772
128	1.330	1.254	1.329

表 7: 経過時間および誤差

ノード数	a) 方法 1 経過時間 (sec)			相対誤差 ($\times 10^{-6}$)	ノード数	b) 方法 2 経過時間 (sec)			相対誤差 ($\times 10^{-6}$)
	固有値	ベクトル	合計			固有値	ベクトル	合計	
96	717.97	716.20	1434.17	1.018	96	155.61	153.92	309.53	1.018
128	610.54	607.13	1217.67	2.141	128	105.49	102.71	208.20	2.141

表 8: プログラム実行解析情報

ノード数	a) 方法 1			ノード数	b) 方法 2		
	V. OP RATIO	FLOPS ($\times 10^9$)	メモリ使用量 (GB)		V. OP RATIO	FLOPS ($\times 10^9$)	メモリ使用量 (GB)
96	96.807	233.086	924.578	96	98.647	1058.770	1109.281
128	96.288	276.946	951.234	128	98.518	1561.437	1117.547

V. OP RATIO...ベクトル演算率 (%)

表 9: 通信時間

ノード数	a) 方法 1 通信時間 (sec)			ノード数	b) 方法 2 通信時間 (sec)		
	最大	最小	平均		最大	最小	平均
96	1298.023	1200.533	1242.756	96	188.795	180.433	181.175
128	1118.167	1042.866	1071.916	128	114.952	108.268	108.832

表 10: 通信の待ち時間

ノード数	a) 方法 1 待ち時間 (sec)			ノード数	b) 方法 2 待ち時間 (sec)		
	最大	最小	平均		最大	最小	平均
96	1249.832	952.889	1072.837	96	32.891	19.212	23.143
128	1078.173	851.624	935.733	128	24.623	14.690	15.837

表 11: 演算時間 (経過時間-通信時間)

ノード数	a) 方法 1 演算時間 (sec)			ノード数	b) 方法 2 演算時間 (sec)		
	最大	最小	平均		最大	最小	平均
96	233.247	135.829	191.053	96	128.302	119.903	127.559
128	174.250	99.004	145.214	128	98.829	92.152	98.272

一般的な方法は表 4 に示したように受信するデータ量が多く、すべてのデータを同時に持つことができないため、受信する毎にそのデータを用いて計算を行なう。その後、そのデータを破棄し次の通信を行なうようにしている。そのため、全てのデータを受信して計算する提案した方法よりメモリ使用量は少なくなる(表 8 参照)。しかし、一般の方法では 1 組でも通信が起ると全てのノードが通信を待たなくてはならないため、表 10 に示されているように、大量の通信の待ち時間が生じてしまう。一方、提案した方法は上記の理由でメモリ使用量は多くなるが、通信の同期は全てのデータを受信した後にのみ行なえば良いため、表 10 のように通信の待ち時間は少ない。

また、ノード単位の経過時間と通信時間の差を演算を行なった時間として最大、最小、平均を表 11 に示す。この結果から、提案した方法の方が演算時間が少なく、また演算時間が均一であることが確認できる。

以上のことから、通信の待ち時間が少ないことおよび計算が均等に分割されていることが提案した方法が一般的な方法より早く計算できることの理由であると結論付けられる。

5.2 精度

また、本解法の精度を確認するため、前節と同じ問題を 128 ノードで計算した際の Lanczos 法の反復回数とその際に計算される小さい方から 10 個の固有値を表 12 に示す。この結果から、今回の問題の場合、

表 12: 反復回数と固有値

反復回数	100	200	300	500	1000
最小固有値	-52.3800350528377	-52.3805243694673	-52.3805243694687	-52.3805243694687	-52.3805243694687
	-52.2707289428148	-52.3038776012698	-52.3038776029458	-52.3805243693801	-52.3805243694687
	-52.0600448112372	-52.1236516050813	-52.1242853508689	-52.3038776029458	-52.3805243694687
	-51.8586743170588	-52.0991384468377	-52.1033175727898	-52.1242853675176	-52.3038776029458
	-51.6301553511271	-52.0309975434936	-52.0530759173400	-52.1033180282741	-52.3038776029458
	-51.3603890356300	-51.9589256238527	-52.0227508091416	-52.0534154010958	-52.3038776029456
	-51.0522781373765	-51.8703884751054	-51.9914468374499	-52.0387875636340	-52.1242853675176
	-50.7365632192349	-51.7745032960706	-51.9562486828735	-52.0260883344726	-52.1242853673373
	-50.3674498547609	-51.6742239956138	-51.9026551389419	-52.0186838469426	-52.1033180282741
	-49.8409969347216	-51.5488504141446	-51.8544054528985	-52.0081997948339	-52.1033180258666

最小およびその付近の数個の固有値を求めるには、300~500回位の反復で収束していることが確認できる。

また、反復回数が多くなると同じ値に近づく複数の固有値が現れる。これは、Lanczos法が丸め誤差に弱いためであると思われる。そのため、複数の固有値を求める際には deflation 等の工夫を行なう必要がある。

6 まとめ

本論文では、地球シミュレータにおいて強相関電子系の計算に現れる大規模な行列の固有値計算を効率的に行なえるベクトル並列計算方法を Lanczos 法を基に提案した。この方法はベクトル計算時のメモリアクセスが連続または奇数等間隔になるようにループの交換を行ない、また並列計算時の通信量が少なく、各ノードの計算量がほぼ均一になる通信および計算方法を採用している。実際に地球シミュレータを利用した数値計算から本計算手法がベクトル演算率 98%以上を達成し、また計算を全てのノードにほぼ均一分散できていることが確認できた。以上のことから、本解法が地球シミュレータのようなベクトル並列計算機に有効であると結論付けられる。

謝辞

本研究を行なうにあたり、24 サイト d-p モデル厳密対角化の物理的意義を御教授して頂いたペンシルベニア大 (現米国オークリッジ研究所) の江上教授、P.Piekarczyk 博士および厳密対角化手法についての議論をして頂いた堀田主任研究員 (原研先端研) に感謝致します。また、地球シミュレータ利用に関して御協力頂いた地球シミュレータセンターの方々および叶野氏 (原研 CCSE) に深く感謝致します。

参考文献

- [1] Bednorz, J. G. and Müller, K. A.: Possible High T_c Superconductivity in the Ba-La-Cu-O System, *Z. Phys.* **B64**, 189(1986).
- [2] Dagotto, E.: Correlated Electrons in High-Temperature Superconductors, *Rev. Mod. Phys.* **66**, 763(1994).
- [3] Tachiki, M., Machida, M. and Egami, T.: Vibronic Mechanism of High-T_c Superconductivity, *Phys. Rev.* **B67**, 174506(2003).
- [4] 地球シミュレータセンターホームページ: <http://www.es.jamstec.go.jp>

- [5] Cullum, J. K. et al.: *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol.1: Theory*, SIAM (2002).
- [6] 森正武, 杉原正顕, 室田一雄: 岩波講座 応用数学 [方法 2] 線形計算, 岩波書店 (1994).
- [7] Barrett, R. et al.: *Templates for the solution of linear systems: Building block for iterative methods*, SIAM(1994).
- [8] 日本電気株式会社: FORTRAN90/ES プログラミングの手引, 日本電気株式会社 (2002).
- [9] 山田進, 町田昌彦, 今村俊幸: 強相関電子系における超大規模固有値問題 -地球シミュレータ上でのベクトル並列計算-, HPCS2004 論文集, pp.103-110(2004).
- [10] 並列数値計算ライブラリ PARCEL ホームページ: <http://parcel.koma.jaeri.go.jp>
- [11] 地球シミュレータセンターホームページ Message Passing Interface:
<http://www.es.jamstec.go.jp/esc/jp/Programming/message.html>