

LOBPCG における段階的収束と window の効果について

今 村 俊 幸††

エルミート疎行列用の固有値アルゴリズムの一つである LOBPCG は、Davidson 法などに比べるとコストが高くまた収束性について未知の部分があるものの、量子多体問題での応用経験では方程式求解の必要が無く並列実装の容易さは特筆に値する。ブロック化による同時固有モード探索も強力であるが、同時計算するモード数の 2 乗に比例するコストと含有モード (固有値分布) による収束性の違いにより十分な性能が得られないことがある。本報告ではデフレーションを効果的に併用することでモード数を制限する window の導入と、SA (Simulated Annealing) のように収束判定の残差ノルムを徐々に減少させながら全モードを大域的に収束させる方法について考察する。

Effect of the multi-step relaxation and window on LOBPCG

TOSHIYUKI IMAMURA^{†,††}

Though LOBPCG as one of the eigenvalue algorithms for the hermitian sparse matrix has higher cost disadvantage and unknown issues with regard to theoretical convergence property compared to Davidson-based methods, we found that parallel implementation is quite easy because of no requirement of solving linear systems through our experiences on a quantum many-body simulation. In addition, blocking facility is also powerful property of the LOBPCG algorithm, however, it sometimes costs heavily due to $O(m^2)$ complexity (here m means the factor of blocking), and unbalanced distribution of the included eigenmodes. In this report, we investigate the idea and the effect of window combined with a deflation procedure, and the multi-step relaxation strategy like the Simulated Annealing method to converge all the eigenmode globally and successively.

1. はじめに

これまでの研究では、量子多体問題のシミュレーションでの Lanczos 法と LOBPCG¹⁾ 法について、地球シミュレータ等の高性能並列計算機での実装ならびにその性能を調査してきた^{2),3)}。対象問題の範囲は狭いものの、数千億次元の系の基底状態の計算を成功させており^{2),3)}、基本性能では LOBPCG が極めて高い性能を示してきた。特に、アルゴリズムの提案者である Knayzev が主張するように、適切な前処理を選択することでその性能が発揮されることは確認できるが¹⁰⁾、その収束性についての理論的な解決は完全なものではない。また、並列計算機上では対角化対象の行列の性質にも依存するが、通信量が極めて少ない対角スケールリングが総合的に計算時間を最小化するという結果を

得ている。

我々の実装とは別に、Jacobi-Davidson 法 (以下 JD 法と記す) をパッケージ化した PRIMME⁶⁾ に関する報告では^{8),9)}、LOBPCG は JD 法を一般化した Generalized Davidson 法 (GD 法) において部分空間の更新を進める過程で修正方程式を解くことなく直ちにリスタートモードに入ることで実装されている。そのため、LOBPCG に適した最適化がされているとは言い難いものの、多くの例で JD 法または GD 法が勝っていた (彼の論文では修正方程式の近似解法に QMR を用いた GDQMR 系統が最良との結論である)。また、PRIMME には LOBPCG の実装として window を用いたものが収録されている。window の効果は反復の過程で LOBPCG が要求する残差方向、共役勾配方向 (探索方向) の無駄を軽減する意味合いが強い。一方、計算コストの面から見た場合、計算モード数の 2 乗に比例する部分を縮小することで全体のコスト削減を期待できる。ただし、単純な window の導入は欲する固有モードから遥かに離れたモードを途中で取り込みながら計算を進行することになる。一般的に LOBPCG は収束の後半に上位のモード (絶対固有値の大きなモー

† 電気通信大学電気通信学部情報工学科

Department of Computer Science, The University of Electro-Communications

†† 戦略的創造研究推進事業 CREST (科学技術振興機構 JST) Core Research for Evolutional Science and Technology, CREST (Japan Science and Technology Agency, JST)

ド)の収束性が極端に悪くなるために、それが全体の収束性(計算時間)を悪化させている事実が存在する。つまり、途中で取り込むモードは離れたものではなく、十分に近くにあるべきである。言い換えれば、下位のモードも上位のモードもほぼ一様に収束させながら window の制御をしなければ、適切な効果は得られにくいと考えられる。本研究の基本的アイデアは次のものである。

- (1) window を導入し反復過程での計算すべきモード数を制御する。
- (2) 最下位モードと最上位モードがほぼ一様に収束するように、段階的に残差ノルムを緩和(減少)させていく。

の2点に集約される。また、window の導入には window を通過済みのモードによる deflation(減次)が必要であるが、この deflation のみでは未通過の固有モードの分布が相対的に0固有値から離れたところに存在し密集分布した状態に近くなってしまふことから、適切なシフトやスケージングにより相対的な(deflateされた行列での0固有値を除いた)固有値分布の改善が必要となる。研究10)で実施された、適切なシフトなどを併用し1),2)のアイデアを実用的なものにするようにした。

window の大きさの選択や、残差ノルムの緩和の状況などには幾つかの選択の余地がある(チューニングパラメタとなる)。これらは、収束安定性、高速化の両面に影響することは容易に予想ができる。さらに、これらの最適な選択は対角化すべき行列の性質に大きく依存することは従来の前処理選択などの問題に共通するものとも予想される。本研究では、基本的なアイデアの実現と共に、パラメタの選択について自動チューニングの立場からその効果について議論し実用的なLOBPCGの可能性について言及することが本研究の主題である。

2. LOBPCG

LOBPCG(Locally Optimal Block Preconditioned Conjugate Gradient)法はKnyazev¹⁾により提唱された部分空間反復法の一つであり、現ステップの方向ベクトルに加えて、前ステップの近似固有ベクトルと残差ベクトルの3本によって部分空間を構成しレイリー商局所最小化(もしくは最大化)によって、次の探索方向(次ステップの近似固有ベクトル)とする手法である。

探索方向にリッツベクトルを選択する点でGD(Generalized Davidson)法の枠組みに近いものがあるが、

```

 $x_0 \neq 0$  (an initial guess),  $p_0 = 0$ 
 $\mu_0 = x_0^T A x_0 / \|x_0\|^2$ 
do  $k=0, \dots$  until convergence
   $w_k = T(Ax_k - \mu_k x_k)$ 
   $V = [w_k, x_k, p_k]$ 
   $S_A = V^T A V$ ,  $S_B = V^T V$ 
  Solve  $S_A v = \mu S_B v$ ,  $v = (\alpha, \beta, \gamma)^T$ .
   $x_{k+1} = V v = \alpha w_k + \beta x_k + \gamma p_k$ 
   $p_{k+1} = V \bar{v} = \alpha w_k + \gamma p_k$ 
   $\mu_{k+1} = x_{k+1}^T A x_{k+1} / \|x_{k+1}\|^2$ 
  if  $\mu_{k+1}$  converge, exit this loop.
enddo

```

図1 LOBPCGの非ブロック版アルゴリズム、実際は部分空間を張る w_k, x_k, p_k は正規化される。

修正方程式を解くことなく(inverse free)リスタートをかける構成になっている点が大きく異なる。非線形最小化問題に対するCG法(NLCG)では、探索方向でのエネルギー最小化を適当な方法で解く必要があるが、リッツベクトルで代用(局所的に近似)することとすればLOBPCGと一致する。そういった意味では、幾つかの既存のアルゴリズムと接点をもったアルゴリズムともいえる。

LOBPCGを収録するライブラリ(ソルバー)として、提唱者Knyazevらが作るblopex⁴⁾、それを含むhydreプロジェクト⁵⁾、PETSc, PRIMME⁶⁾、AdventureCluster⁷⁾などがある。また、我々のように実装が容易であるためこれらライブラリを使用せずにアプリケーション内で独自実装する報告も多く見られる。

LOBPCGのコア部分は図1の様である。

図1は非ブロック版であるが、ブロック版では各ベクトルを w_k から $\{w_k^{(1)}, \dots, w_k^{(m)}\}$ の様に修正すればよい。計算を安定に計算するためには、部分空間を張るベクトルの直交化を施すことを要求されるが、一次独立性の破綻の兆しがあれば必ずしも必要ではないことを注意しておく¹¹⁾。

2.1 deflation

反復の過程で、収束した固有モードについてはその更新を停止させることができる。射影問題から収束した固有モードを取り除く操作をデフレーションと呼ぶが射影問題を縮小できるため $O(m^2)$ のコストから大きく計算量削減に寄与する。

* 文献8),9)での測定結果でLOBPCGが遅い理由の一因として直交化の負担が大きいの分析をする研究者の意見もあるが、直交化のタイミングを適切に選択すれば実際にはそれほどにもならないのではと個人的には考えている。

一般に対称行列の場合、ベクトル u のデフレーションは次式を用いて行われる。

$$A' = (I - uu^T)A(I - uu^T) \quad (1)$$

または、部分空間基底 V から u を除いた \tilde{V} に対して $(I - uu^T)$ を作用させ

$$V' = (I - uu^T)\tilde{V} = \tilde{V} - (u^T\tilde{V})u \quad (2)$$

とした V' を選択することも上式と等価な操作である。

文献 11) でも説明したように、この操作は u に対応する行と列の吐き出し操作に当たり射影行列上でも deflation を実施することができる。計算コストの上では後者が有利ではあるが、誤差の蓄積のリスクを含むので注意が必要である。

2.2 window

LOBPCG の収束性の議論については、文献 1) によりなされている。要約すると、小さい固有値のモードほど収束性がよく、また m が大きい、つまり、同時に計算する固有モードの数が多いほど収束が速い。しばしば、固有モードの収束性の極端な違いから、固有値探索が非効率になっている懸念が残る。

また、LOBPCG は探索 1 モードあたり最低 3 本の部分空間を使って固有空間を探索する。これは、探索すべきモード数が大きくなれば他のアルゴリズムよりも消費する作業領域 (メモリ) が大きくなることを意味している。これに対して、LOBPCG を収録する PRIMME ライブラリ⁶⁾ では window の導入により探索空間の増大 (メモリの浪費) を一時的に制限する。図 2 に示すように、最小モードの収束により window をシフトし未収束のモードを window に取り込みながら計算を進行させていく。図中では収束済みの固有モードは \tilde{V} に取り入れられ、未収束モードに対して deflation の対象ベクトルとなる。

window の導入の意図はモード数を大きく取れば収束性が改善するという Knyazev らの主張とは相容れないものであるが、1 反復あたりのコストを大きく改善できる点では性能改善の可能性がある。ちなみに、 m 本の探索モードがある場合は計算コストの概算は $O(4Nm^2)$ 、ここで N はベクトルの次元である。deflation を常に行う場合は、deflation の対象ベクトルの本数 q に対して $m - q$ 本のベクトルにそれぞれ $O(2Nq)$ のコストが必要となる (つまり、ある反復では $O(2Nq(m - q))$ が必要)。常に、 m を固定または $q = 0$ に固定する場合よりも、若干計算コストは削減できる可能性があることになる。実際には、 N が非常に大きいためキャッシュミス等の影響により演算量に計算時間が因らない部分があるが、大よその性能としては改善の見込みを期待できることになる。

Set $X_0 = \text{Normal}\{x_0^{(1)}, \dots, x_0^{(m)}\}$,

$P_0 = 0$, and $\tilde{V} = \emptyset$

do $i=0, m$

$\tilde{X}_k = \{x_i, \dots, x_{\min(i+w, m)}\}$,

$\Lambda_0 = \text{diag}(\tilde{X}_0^T A \tilde{X}_0)$, and

if $(i + w \leq m)$ then

$\tilde{P}_k = \{p_i, \dots, 0\}$

else

$\tilde{P}_k = \{p_i, \dots, \}$

endif

do $k=0, \dots$ until convergence

$\tilde{W}_k = T(A\tilde{X}_k - \Lambda_k\tilde{X}_k)$

$V = (I - \tilde{V}\tilde{V}^T)[\tilde{W}_k, \tilde{X}_k, \tilde{P}_k]$

$S_A = V^T A V$, $S_B = V^T V$

Solve $S_A v = \mu S_B v$, $\{v, \dots\} = (A, B, C)^T$.

$\tilde{X}_{k+1} = \text{Normal}(\tilde{W}_k A + \tilde{X}_k B + \tilde{P}_k C)$

$\tilde{P}_{k+1} = \tilde{W}_k A + \tilde{P}_k C$

$\Lambda_{k+1} = \text{diag}(\tilde{X}_{k+1}^T A \tilde{X}_{k+1})$

if $\Lambda_{k+1}^{(1)}$ converge, exit this loop.

enddo

$\{x_i, \dots, x_{\min(i+w, m)}\} = \tilde{X}_k$,

$\{p_i, \dots, p_{\min(i+w, m)}\} = \tilde{P}_k$,

$\tilde{V} = [\tilde{V}, x_i]$

enddo

図 2 window (幅 w) を導入した LOBPCG アルゴリズム

特に断りが無い場合は添え字付大文字記号はブロック化された (複数) ベクトルを指す。

2.3 収束の段階的な緩和 (Relaxation)

window の導入は各反復の計算コストの軽減の可能性を見せたが、実のところ大域的な収束における計算事件の改善にはあまり寄与をしないことが予想される。何故なら、序節でも示したように反復の後半になればなる程、上位モードの収束が停滞するために window シフトによって後半に window に取り込まれる固有モードは初めから停滞の原因となる可能性がある。

この停滞をできるだけ改善するために、下位から上位の全てのモードに対してできるだけ収束の状態 (残差の減少の程度) を均一化しながら、できるだけ上位モードの計算時に数多くの部分空間候補を供給することが必要となる。

図 2 で示したアルゴリズムの最外ループを LOBPCG の基本コアとして打ち切り判定誤差を ϵ を用いて LOBPCG(X, Γ, m, ϵ) で表現する。このとき、 ϵ を問題の最終打ち切り誤差に設定するのではなく、

$$\epsilon_k = a^k \epsilon_0, \quad a < 1 \quad (3)$$

の様に段階的に下位モードから上位モードまでを含む全モードの残差を減少させていくことで、下位モードと上位モードの極端な収束性の違いが現れないように制御が可能となる。

この様な打ち切り残差の段階的な緩和はエネルギー最小化問題の一手法の Simulated Annealing に似ている。厳密には、本手法には焼鈍しの手続きは含まれてはいないが window に新たに入った探索モードをシステムへの擾乱と捕らえてもよい。また、別の観点から見れば緩和を行わずに直接大域的な残差まで収束させるのは深さ優先的な探索で、緩和を行うのは幅優先的な探索に近い。

2.4 適応シフト

研究 10) では、LOBPCG の収束の過程で得られた近似固有値でのシフトを行った上で作られる行列の前処理により収束特性 (性能) が向上することが報告されている。また、近似固有値でのシフトの効果は deflation と組み合わせることで、原点近傍の固有モード探索へもとの問題を変更する効果がある。この際に原点近傍に求めるべき固有モードが集まることで、相対的な固有モード間での差異が拡大される。具体的には、原点近くに固有値が集約することで固有値表現のオフセット部分による情報落ちを避けることができ、モードを切り分け易くなる (近い固有モードによって構成される固有空間に停滞することが少なくなる)。

2.5 提案スキーム

本報告で検証するスキームについてまとめた。ここまで述べた、deflation+window+relaxation+shift をうまく組み合わせたものである (本稿では以下 DWRS-LOBPCG と呼ぶ)。これまでに示した図 1 や 2 に加えて、Relaxation スキームと適応シフトを導入したアルゴリズムを図 3 に示す。4 番目の適応的なシフトに基づく前処理が図中で不明確だが、 $T(\Lambda_k)$ が $A - \Gamma_k^{(1)}$ に対する前処理を意味している。場合によっては、文献 10) の様に各近似固有値でシフトし逆反復による固有モードフィルタとして作用させてもよいであろう。

3. 数値実験

3.1 計算機環境

実験に使用した計算機は Intel Xeon 5345 (2.33GHz, Quad-core, dual CPU's, FBDIMM-533MHz 4GB) を搭載するサーバマシンであるが、実際に使用したのは 1 コアのみである*。また、使用したコンパイラは In-

```
// Algorithm DWRS-LOBPCG
// Input is a matrix A
// Output is a set of eigenvectors {x(1), ...}
Set X0 = Normal{x0(1), ..., x0(m)},
P0 = 0, and V̄ = ∅, ε = max ||X0||
do
  ε = ε * a (a < 1 is given somehow)
do i=0,m
  X̄k = {xi, ..., xmin(i+w,m)},
  Λ0 = diag(X̄0T A X̄0), and
  if(i + w ≤ m) then
    P̄k = {pi, ..., 0}
  else
    P̄k = {pi, ..., }
  endif
do k=0, ... until convergence
  A means symbolically A - Λk(0) I,
  W̄k = T(Λk)(A X̄k - Λk X̄k})
  V = (I - V̄ V̄T)[W̄k, X̄k, P̄k]
  SA = VT A V, SB = VT V
  Solve SA v = μ SB v,
  and set the v as {v, ...} = (A, B, C)T.
  X̄k+1 = Normal(W̄k A + X̄k B + P̄k C)
  P̄k+1 = W̄k A + P̄k C
  Λk+1 = diag(X̄k+1T A X̄k+1)
  if Λk+1(1) < ε, exit this loop.
enddo
  {xi, ..., xmin(i+w,m)} = X̄k,
  {pi, ..., pmin(i+w,m)} = P̄k,
  V̄ = [V̄, xi]
enddo
enddo
```

図 3 proposed-LOBPCG アルゴリズム (DWRS-LOBPCG) window-LOBPCG と同様に、断りが無い場合は添え字付大文字記号はブロック化された (複数) ベクトルを指す。

tel Fortran Compiler 10.1(20080112 版) である。使用最適化オプションは -O3 -axP -fp-model fast=2 である。

射影空間問題でのリッツベクトルを計算する内部ルーチンとして、LAPACK+ATLAS の dsyevd 関数を使用した。

3.2 対象問題

提案手法の検証には疎行列ベンチマークにしばしば利用される Florida Matrix Collection¹²⁾ や Matrix

* 今回の計算は計算機の差異によって生じる計算時間を比較する

ことが目的ではなく、提案手法の効果を示すものである故に。

Market¹³⁾ から、次の 8 種類を選択した。

- H2O (67024, 1141880)
- Ga41As41H72 (268096, 9378286)
- Ga19As19H42 (133123, 4508981)
- SiO2 (155331, 5719417)
- Lin (256000, 1011200)
- cfd2 (123440, 1605669)
- mhd4800b (4800, 16160)
- bcsstk28 (4410, 111717)

表に示したカッコ内の数字はそれぞれ、次元と行列の非ゼロ要素数を示している。初めの 4 行列は DFT(Density Functional Theory) に基づく電子状態計算コード PARSEC¹⁴⁾ から生成されたものである。作者の分析では密集固有値によるクラスタがあり、固有値計算のベンチマークには適したものである。続く 2 つは Matrix Collection から、残りの 2 つは MatrixMarket から選択した。個々の行列の特性分析についてはそれぞれの配布もとの HP にあるので、ここでは詳細は省略する。

3.3 計算結果

選択した行列に対して、最小固有値から 24 本の固有モードを同時に計算するのに要した計算時間ならびに、行列ベクトル積の回数を測定した。一般に疎行列固有値計算の計算コストは行列ベクトル積の回数で評価されることが多いため本報告でも同様のデータを測定している。表 1 は window の幅を 1 から 24 まで変化させたものに加え、表中 “Normal” で記している「通常の LOBPCG」の測定対象行列での計算時間と行列ベクトル積の回数を配した。但し、前処理の違いによる収束性の違いを排除するために、両者には適応シフトを同時に行う対角スケールリングでの前処理を実施している。また、緩和係数 (図 3 における変数 α) は $0.0625=1/16$ を選択した。

いずれの対象行列でも、Normal な LOBPCG アルゴリズムよりも DWRS-LOBPCG の方が計算時間は速くなっている。しかしながら、計算時間に依存すると考えられている、行列ベクトル積の発行回数は必ずしも最速のアルゴリズム (最適な window の設定) で最小になっているわけではない。最適な window サイズ w は、H2O, Ga41As41H72, Ga19As19H42, SiO2, Lin については $w = 4$ で計算時間を最小化している。また、cfd2 では $w = 12$, mhd4800b や bcsstk28 ではそれぞれ 1, 2 がよいなど行列生成のもととなったアプリケーションの種別によってその傾向が分かれる結果となった。

殆どの例で、30%から 40%、最も高速化がなされた

mhd4800b では 50% 以上の性能改善がなされている。適切な前処理による劇的な性能改善とまではいかないものの、今回の例の様に重複固有モードを保有する問題では活かされるアルゴリズムではないかとの期待ができる。これらを立証するには理論面からの更なる検証、他実装との比較などを実施しなければならない。また、今回は個定したままの緩和係数、さらには緩和過程の制御も含めて性能に影響を与えるのかの追加実験も必要である。

いずれにせよ現実実験結果から得られる良好な示唆について今後より解析を深めていくことが今後の課題である。

4. 最後 に

疎行列向け固有値アルゴリズム LOBPCG の実装において、昨年 of SWoPP2007 の発表に引き続き量子多体問題における多数子固有モードの同時探索から端を発した各種チューニング問題を經由して、LOBPCG の最適化について deflation+window+relaxzation+shift の 4 要素を組み合わせた DWRS-LOBPCG について考察をした。実験対象が 8 つと対象アプリケーションも限られるものの全体を通じて良好な性能を示した。今後は未着手のパラメータ制御や理論的な解析なども視野に入れつつ実装系の充実にも着手していきたい。

謝 辞

本研究の一部は文部科学省科学研究費、JST CREST の支援を受けています。

参 考 文 献

- 1) A.V. Knyazev : Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method, SIAM, J. Sci. Comput., Vol.23, No.2, pp.571-541, 2001.
- 2) S.Yamada, T.Imamura, M.Machida: 16.447 TFlops and 159-Billion-dimensional Exact-diagonalization for Trapped Fermion-Hubbard Model on the Earth Simulator, IEEE&ACM, Proceedings of SC|05, 2005.
- 3) S.Yamada, T.Imamura, T.Kano, and M.Machida: High-Performance Computing for Exact Numerical Approaches to Quantum Many-Body Problems on the Earth Simulator, IEEE&ACM, Proceedings of SC|06, 2006.
- 4) A.V. Knayzev, et. al : Block Locally Optimal Preconditioned Eigenvalue Solvers (BLOPEX) in hypre and PETSc.
- 5) Scalable Linear Solver project, High Perfor-

表 1 提案アルゴリズム (DWRS-LOBPCG) の Frolida Matrix Collection を用いた検証結果:
 最小固有値から 24 本の固有モードを同時に計算する. 各コラムの上段は反復終了までに発行された行列ベクトル積演算の総数, 下段は計算時間.

	H2O	Ga41As41H72	Ga19As19H42	SiO2	Lin	cf2	mhd4800b	bcstk28
Normal	13983	46333	24700	14484	12205	156387	187738	41249
	431[s]	8631[s]	1979[s]	1417[s]	1649[s]	8324[s]	379[s]	87.5[s]
w=1	11425	36724	23861	16722	12743	193431	196062	54358
	335[s]	8199[s]	1988[s]	1825[s]	1809[s]	11483[s]	166[s]	67.4[s]
w=2	13440	32671	24235	14388	13524	176424	256831	43794
	327[s]	5921[s]	1677[s]	1275[s]	1415[s]	8478[s]	191[s]	45.9[s]
w=4	13841	30697	20541	14303	13987	149915	253860	41690
	317[s]	5357[s]	1394[s]	1185[s]	1390[s]	6919[s]	209[s]	46.5[s]
w=8	17574	31385	21782	15260	15239	128369	286884	36532
	455[s]	5662[s]	1588[s]	1296[s]	1603[s]	5855[s]	286[s]	48.9[s]
w=12	15507	33255	28819	16370	15124	120380	271744	32699
	397[s]	6187[s]	2218[s]	1477[s]	1762[s]	5794[s]	309[s]	48.0[s]
w=24	16441	60488	27715	16772	15799	114452	278448	43657
	510[s]	12077[s]	2396[s]	1705[s]	2336[s]	6123[s]	473[s]	89.2[s]

mance Preconditioner,

http://www.llnl.gov/CASC/linear_solvers/.

- 6) PRIMME, PReconditioned Iterative Multi-Method Eigensolver,
<http://www.cs.wm.edu/andreas/software/>.
- 7) 大山知信, 鈴木正文, 他: ADEVENTURECluster の振動解析, 日本機会学会 2002 年度年次大会講演資料集, S59-3, 2002.9.
- 8) A.Stathopoulos: Nearly Optimal Preconditioned Methods for Hermitian Eigenproblems under Limited Memory. PART I: Seeking One Eigenvalue, SIAM, J. Sci. Comput., Vol.29, No.2, pp.481-514, 2007.
- 9) A.Stathopoulos, and J.R.McCombs: Nearly Optimal Preconditioned Methods for Hermitian Eigenproblems under Limited Memory. PART I: Seeking Many Eigenvalues, SIAM, J. Sci. Comput., Vol.29, No.5, pp.2162-2188, 2007.
- 10) 山田進, 今村俊幸, 町田昌彦: 量子大規模固有値問題における共役勾配法の収束性: 適応的シフト前処理の収束性の評価, 計算工学会論文誌, No.20060027, 2006.
- 11) 今村俊幸, 山田進, 町田昌彦: 疎行列固有値ソルバーの自動チューニング, LOBPCG の量子多体問題への応用を中心に, 情処研報, Vol.2007, No.80, pp.167-172, 2007.
- 12) University of Florida Sparse Matrix Collection by T.Davis,
<http://www.cise.ufl.edu/research/sparse/matrices/>
- 13) Matrix Market,
<http://math.nist.gov/MatrixMarket/>
- 14) Y.Saad, Y.Zhou, et al.: Diagonalization methods in PARSEC, Phys. Stat. Sol.(b) 243, No.9, pp.2188-2197, 2006.