

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

今 村 俊 幸^{†,††} 山 田 進^{†,††} 町 田 昌 彦^{†,††}

我々は、エルミート疎行列用の固有値アルゴリズムである LOBPCG を量子多体問題に応用し、従来法のランチョス以上の性能を発揮することを示してきた。LOBPCG のもう一つの特徴である、ブロック化による同時固有モード探索においては、実装時に注意しなくてはならない点が幾つか分かってきた。これらの項目は実行速度や計算精度にも影響を及ぼすパラメタであり利用者のポリシー指示により様々に変更できるべきものである。自動チューニングの観点からこれらパラメタについて考えていくが、本論文では準備代価意図してパラメタの抽出、それらの数学的な性質・振る舞いなどを中心に議論する。

Automatic tuning technology for a sparse eigensolver, application of LOBPCG to quantum many-body problems

TOSHIYUKI IMAMURA,^{†,††} SUSUMU YAMADA^{†,††}
and MASAHIKO MACHIDA^{†,††}

We have shown that LOBPCG that is one of the eigenvalue algorithms for the hermitian sparse matrix is applied to quantum multi-body problems, and it demonstrates higher performance than the Lanczos method. In an eigen-mode search, LOBPCG has another feature of simultaneous or blocking searching, and several points must be carefully treated in the implementation. They are the parameters that affects the performance and the accuracy, and should be the variously specified by user's policy. In this report, the extraction of the parameter, those mathematical characters, behavior, and so on are discussed intending the preparation stage from the viewpoint of an automatic tuning.

1. はじめに

ハバードモデルの様な量子多体問題のシミュレーションにおいて、我々はこれまでに Lanczos 法と LOBPCG¹⁾ 法を用いることで、地球シミュレータ上で数千億次元の系の基底状態の計算を成功させてきた^{2),3)}。これまでの基底状態計算のみの場合は、最小固有値モードの追跡のみであり LOBPCG では適切な前処理を選択することで非常によい収束性を得ることができた。Lanczos 法も精度的に LOBPCG と同等の結果を残したが、解法の性質上、より高い精度に解(この

場合固有値と固有ベクトル)を改善することは通常の Lanczos 法単体では難しいといえる。また、Lanczos 法は省メモリで固有値計算ができるがその場合は固有ベクトルは別途計算(およそ2倍の計算時間)しなくてはならず、計算時間と使用メモリの観点から多くの場合 LOBPCG が有効であるとの結論付けられた。

また、一方、LOBPCG は多重モードを同時に計算することができる解法であり、ゴーストモードの問題が懸念される Lanczos と比べ非常に有用なアルゴリズムといえる。先行研究では単に基底状態による定常的な振る舞いのみを解析してきたが、時間発展型も必要性が高まっており、本研究では LOBPCG を用いて基底状態から数えて数十の固有モードを同時に計算するが、その実装・実行において発見された計算不安定性やその改善のための様々なチューニングパラメタが必要となる。各種パラメタについては本報告中で説明するが、これらパラメタの適切な選択による LOBPCG ソルバーの安定化・高速化について議論することが本研究の主題である。

[†] 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

^{††} 日本原子力研究開発機構システム計算科学センター
Center for Computational Science& e-Systems, Japan
Atomic Energy Agency

^{†††} 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)

```

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

2. 従来実装の問題点

2.1 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, 修正方程式を解かなくても良い点(inverse free)から非線形最小化問題に対するCG法(NLCG)からの改良と捉える方が自然かもしれない。前ステップの近似固有ベクトルを用いるアイデアはEIGIFP⁶⁾などにも採用されているが, 僅か3本の部分空間により高速に最小固有値に到達するアルゴリズムは他に見られない。このLOBPCGを使用した固有値ソルバーは, 提唱者らが作るblopex⁴⁾やそれを含むhyprプロジェクト⁵⁾やPETSc, PRIMME⁷⁾やAdventureCluster⁸⁾にも採用されている。

LOBPCGのコア部分を示すと図1の様になる。我々は地球シミュレータ用に, blopex等移植するのではなく, 地球シミュレータ専用コードを作成しSC05,06にて報告をしてきた。その際, 求解モードは基底状態のみであったためLOBPCGのBlock解法としての特性を活かすことはなかった。本年度我々のテーマが時間発展問題になり, 若干システムの規模は小さくなるものの多数のモードを同時に計算することになり, これまでには直面しなかった幾つかの問題

```

do i=1,n
  do j=1,n
    do k=1,n
      y(k,j,i)=d(k,j,i)*x(k,j,i)\
      +a*(x(k-1,j,i)+x(k+1,j,i)\
      +x(k,j-1,i)+x(k,j+1,i)\
      +x(k,j,i-1)+x(k,j,i+1))\
      +b*(x(k-1,j-1,i)+x(k+1,j+1,i)\
      -x(k-1,j+1,i)-x(k+1,j-1,i))\
      +x(k,j-1,i-1)+x(k,j+1,i+1)\
      -x(k,j-1,i+1)-x(k,j+1,i-1))
    enddo
  enddo
enddo

```

図2 3層構造システムの行列計算コアループ(周期境界条件)

に遭遇した。以降ではそれについて簡単に触れる。

2.2 計算困難な問題

上記プログラム(図2)は, 3層構造をなすジョセフソン素子の計算を実施する際に現れるハミルトン行列で行列ベクトル積($y = Ax$)を計算するための中核ループを表したものである。変数dによって, 外場・拘束ポテンシャルなどを設定することができる。このdを定数とし, $d+6*a=0$ を満足するように決めれば, 本システムは0固有値を持ち(固有ベクトルは同一要素値, つまり一様分布状態), なおかつ対称性から多くの重複固有モードを持つ。表1はターゲットモデルとなる4層システムにおいて64⁴⁾の解像度で下から24モードを計算したものである。表からも分かるように, 固有値が重複しており部分空間法で過不足無く直交基底を探索することが要求される。

blopexを実際に用いて計算したところ, 求解精度を残差ベクトルの相対ノルムの 10^{-6} 程度にした場合は破綻することなく全ての固有値を計算することができた。一方, 打ち切り相対残差を 10^{-10} にしたところ, 内部エラーを起こして計算を終了してしまった。実行ログ図3にあるように, 内部ルーチンとして利用して

表1 4層ジョセフソン構造における固有モード(次章以下に述べる各種チューニングを施しLOBPCGを用いて下から24モードを小数点以下11桁の精度で計算したもの)

No.	近似固有値	No.	近似固有値
1	3.89073345186838	13	4.58929855884546
2	4.33498012571068	14	4.58929855884552
3	4.33498012571074	15	4.58929855884558
4	4.33498012571074	16	4.77922679955573
5	4.33498012571076	17	4.77922679955574
6	4.33498012571077	18	4.77922679955575
7	4.33498012571077	19	4.77922679955577
8	4.33498012571078	20	4.77922679955578
9	4.33498012571082	21	4.77922679955579
10	4.58929855884542	22	4.77922679955581
11	4.58929855884543	23	4.77922679955581
12	4.58929855884545	24	4.77922679955582

いる GEVP が何らかの障害を起こしたものであるが詳細は今のところ不明である。

Solving standard eigenvalue problem without preconditioning

block size 24

No constraints

```
Initial Max. Residual 8.3974751983050986e+00
Iteration 1  bsize 24      maxres 7.5919344189564653e+00
Iteration 2  bsize 24      maxres 6.4853821663563426e+00
Iteration 3  bsize 24      maxres 5.3305583613163598e+00
Iteration 4  bsize 24      maxres 4.2125983231377848e+00
Iteration 5  bsize 24      maxres 3.37045375800575269e+00
```

.....

```
Iteration 228 bsize 23      maxres 1.7319375763532152e-05
Iteration 229 bsize 23      maxres 1.4448864669909518e-05
Error in LOBPCG:
GEVP solver failure
```

図 3 blopex の実行ログ (前処理なし)

我々が開発したルーチンも破綻を起こす恐れがあったため、念のために非ブロック版を block 版に改良したもののテストを実施した (計算は Intel Xeon server にて)。結果が図 4 であり、何のケアもなしに反復を進行させると一時的に収束した固有モードであっても発散に転じるなどの不具合が生じ、全体として計算が破綻することが確認された。我々の開発版は、地球シミュレータの性能を最大限に発揮するために、通信が頻発する行列ベクトル積を線型計算で代用していく方式をとっていたために、線型計算 ($Ax^{(k+1)}$ を $X^{(k)} = Ax^{(k+1)} := A(\alpha x^{(k)} + \beta w^{(k)} + \gamma p^{(k)}) = \alpha X^{(k)} + \beta W^{(k)} + \gamma P^{(k)}$ でもって計算する方法。 $X^{(k)} = Ax^{(k)}, W^{(k)} = Aw^{(k)}, P^{(k)} = Ap^{(k)}$ は予め何らかの方法で計算) によって $Ax^{(k)}$ 等の計算を削減している。我々の問題の場合、最小固有モードを計算しているため、この値が反復が進むにつれ実値と大きく異なってくるおそれがある。実際、全てのステップにおいて行列ベクトル積を実施したものが図 5 である。残差が全体的にうねっている傾向にあるが、予想通り計算が破綻することなく収束することができた。

この様に、性能安定化について何らかのパラメタ調整が必要な場合もあり、その調整を利用者の計算ポリシー指定のもとで最良にするための知見収集を次章以下述べていくことにする。

3. LOBPCG の実行前・実行時パラメータ

3.1 AX の再計算

前章でも述べたように、行列ベクトル積は反復の安

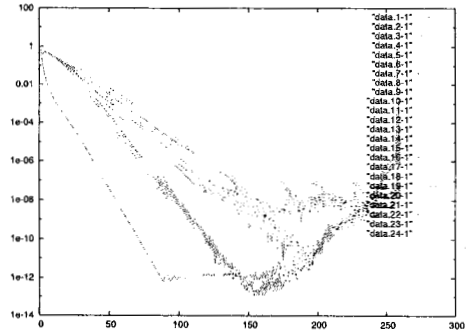


図 4 初期開発版 LOBPCG の収束特性 (前処理なし)

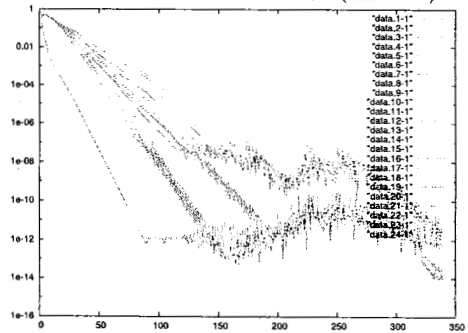


図 5 常時行列積を計算する版の収束特性 (前処理なし)

定性の他計算精度にも大きく関与してくる要素である。多くの論文での評価に (残差/行列乗算) が用いられていることが多い⁹⁾。これは、行列乗算が非常に高価であり、1 反復あたりの乗算回数が実質の計算時間に関連しているためである。LOBPCG で常時行列乗算を行うと 1 反復あたり $3m(m+1)/2$ の乗算が必要となり、多くの解法と比べても非常に不利である。

我々の実装による線型計算による乗算削減では、計算誤差などの影響が必ず含まれてしまう。そこで、次の方針で行列乗算を実施し誤差の蓄積を解消する。主な方針は

- (1) 周期的に実施。
- (2) 残差が急激に増加したものがあつたときに実施
ただし、収束済みの固有モード数に応じて変動幅を設定。

であり、問題に応じて実行時に変更できるパラメタを設定した。

```
#M=N.of.Converged_Mode;
if(itr%MAX(3*m,50)<=2){
    OK=TRUE; // 周期的に再乗算
}
if(#M<=m*.50){
```

```

    κ = 1.0E+03;
} else if (#M<m*.75){
    κ = 1.0E+02;
} else if (#M<m*.90){
    κ = 1.0E+01;
} else {
    κ = 1.0E+00;
}
for(i=0;i<M;i++){
    δ=res[i]/res_min[i];
    if(δ > κ){ // 残差が急増した場合
        resm[i]=res_min[i];
        OK=TRUE;
        break
    }
    if(res_last[i]<TOL && δ >1.0E+01){
        OK=TRUE;
        break
    }
}
}

```

3.2 部分空間基底 V の直交化

部分空間の選択は効率的な解探索の上で重要であり、基本的には部分空間を張る基底は直交することが要請される。一方、直交しない場合には、 A ではなく $A - \lambda I$ の pencil となり内部で一般化固有値問題を扱うことになる。したがって、射影問題の固有値計算が標準固有値問題になるか一般化固有値問題になるかの違いであり、実装上の問題は大きくはない。

ただし、部分空間反復法の考え方である、部分空間の制約条件の中で最良近似を得る点で、基底の一次独立性が破れることは許されないため、何らかの直交化を行い部分空間の縮退を避けなければならない。blopex などでは、ベクトル p に対して直交化を施し一次従属となるベクトルを排除している。直交化を行わない場合でも、一般化固有値問題の解法で

$$Ax = \lambda R^T R x$$

$$\rightarrow R^{-T} A R^T (R x) = \lambda (R x) \quad (1)$$

としている部分がある。この右辺のコレスキー分解が、射影をしない部分空間基底に対するグラムシュミットの直交化法に対応するため、実際には直交化を行っていることになる。しかしながら、一般の固有値計算ルーチンではコレスキー分解が破綻する場合、部分空間基底が一次従属になり正值性が失われるときにエラーとしてしまうため、これ以上計算を進行させることができなくなる。軸選択などの工夫を行い、一次従属基底

(この場合正確には基底とはいえないが) を排除して、部分空間を縮小または基底を新たに追加するなどして計算を進める必要がある。

なお、射影問題でのコレスキー分解では基底間の内積でのみ評価し計算を行うため、実際の部分空間基底において誤差が一様に拡大される恐れがある。一方、LOBPCG のブロック版では対象となる部分空間基底が 3 倍となるためそのコストは行列乗算並みに膨れ上がる可能性がある。そのため、

- (1) 殆どのステップで射影問題のコレスキー分解で済ませ、周期的に部分空間基底で直交化する
- (2) 一次独立が保証できない状態、負固有値や残差の爆発が観測された際に部分空間基底での直交化を実施する。

3.3 デフレーション 1(収束済み固有モードの扱い)

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計算量削減に寄与すると共に、収束モード更新によるノイズのために反復回数増加させる問題を解消することができる。

デフレーションも部分空間基底での扱いと、射影問題での扱いの 2 種類が存在する。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う。

$$A' = (I - uu^T)A(I - uu^T) \quad (2)$$

または、部分空間基底 V から u を除いた V' に対して $(I - uu^T)$ を作用させ $(I - uu^T)V' = V' - (u^T V)u$ とすることも、上式と等価な操作である。この操作を射影問題の上で解釈すると、 u に対応する行と列の吐き出し操作に当たるため、吐き出し操作の後に行と列の入れ替えにより u に対応する部分を取り除き減次することができる。

一方、デフレーションにより収束したと判断される基底を固定した場合に、そのその基底が内包する誤差をこれ以上改善できないリスクを含んでいる。したがって、部分空間全体が解空間に近づいた際には(全ての近似固有ベクトルの残差が基準値を下回った場合)、解の精度の改善のための手当てが必要であろう。なお、blopex などのルーチンではこのようなアンチデフレーションの処置は考えられていない。

3.4 デフレーション 2(未収束モードへの部分空間補充)

Knayzev によれば、漸近収束係数は

$$q_j = \left(\frac{1 - \sqrt{\xi_j}}{1 + \sqrt{\xi_j}} \right), \xi_j = \kappa_A^{-1} \frac{\mu_j - \mu_{m+1}}{\mu_j - \mu_{\max}} \quad (3)$$

を用いて、 q_j^2 に比例するとされている¹⁾。これは、小さい固有値ほど収束性がよく、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ことを示している。より具体的には、最大固有値と次固有値との相対差比が大きいほど収束性がよくなる。したがって、固有モード (固有値) によって収束性が違うため、大きな固有値の探索が全体の固有値計算を支配する。一方、小さな固有値はすぐに収束し計算には寄与しない (実際には直交化に登場するが)。

したがって、収束後に作業ベクトルの中で更新されない残差ベクトル $w^{(k)}$ や前探索方向 $p^{(k)}$ について、 m 以降の固有モード計算に利用し収束係数を緩和させ収束性能を上げることを行う。

3.5 初期ベクトルの選択

初期ベクトルの選択も、反復特性に影響を与える。

- (1) 単位ベクトル: e_i
- (2) 乱数ベクトル
- (3) 近似固有ベクトル (LOBPCG 以外の方法で計算)

などが考えられる。当然、(3) がよいのであるが、我々の開発版のデフォルトでは (2) を利用する。

3.6 前処理の選択

前処理は LOBPCG 系統の反復特性を改善する大きな項目であることは間違いない。各種前処理についての報告は存在するため本報告ではあえて取り上げない。

4. 予備実験結果

先の例と同様に、4 層ジョセフソン素子問題について取り扱う。a) 外部ポテンシャルなし (図 4,5 と同様)、b) 外部ポテンシャルありの 2 種類の問題設定に対して、デフレーション 1,2 の有り無しの 4 種類でのソルバパラメタ設定下での測定を実施した。なお、行列乗算は 3.1 で示した方針で実施し、前処理は行っていない。初期固有ベクトルは乱数を使用した。収束判定を相対残差 ($\|A\hat{x} - \hat{\mu}\hat{x}\|/\|\hat{\mu}\hat{x}\|$) について 10^{-10} で判定を行った。図 6 から 9 が問題設定 (a)、図 10 から 13 が (b) である。

いずれの場合もデフレーション (1 あり,2 あり) が比較的安定にかつ、最小の反復回数で解に到達している。デフレーション 1 あり 2 ありの併用では残差が突然大きく振れることがあり、別の行列データを用いた場合には残差全体が同時に振れることもある。したがって、デフレーションの選択を逐次切り替えるなどのことは可能である。また、本研究では取り上げなかった前処理や初期ベクトルの選択もチューニングパラメタとして有用である。

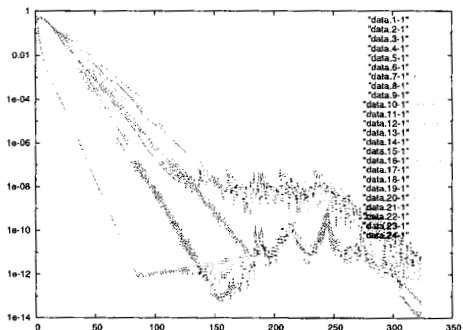


図 6 問題 a, デフレーション (1 なし,2 なし)

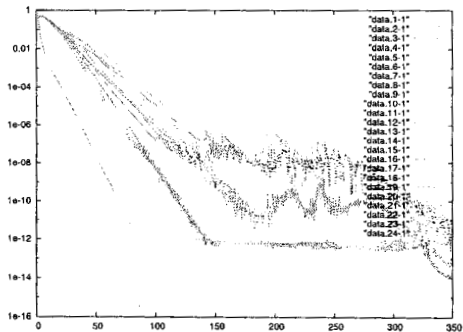


図 7 問題 a, デフレーション (1 あり,2 なし)

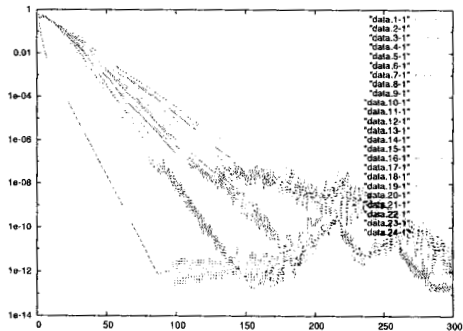


図 8 問題 a, デフレーション (1 なし,2 あり)

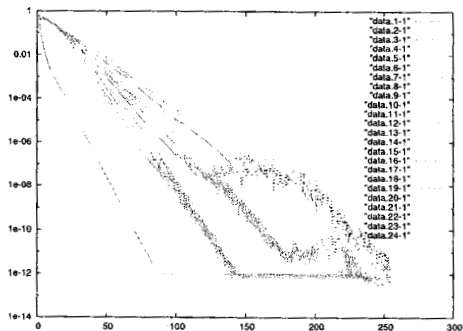


図 9 問題 a, デフレーション (1 あり,2 あり)

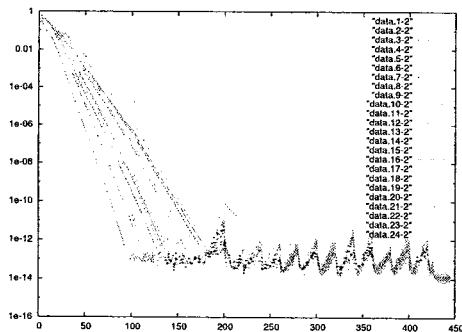


図 10 問題 b, デフレーション (1 なし, 2 なし)

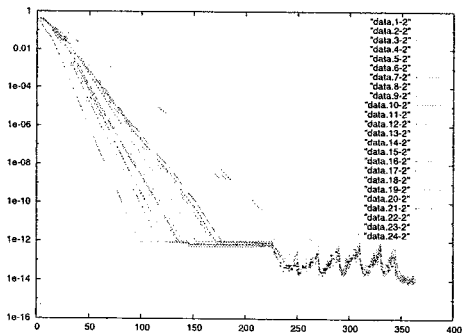


図 11 問題 b, デフレーション (1 あり, 2 なし)

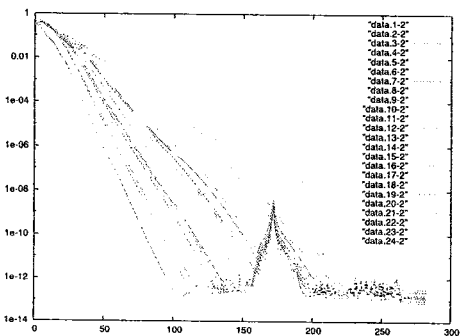


図 12 問題 b, デフレーション (1 なし, 2 あり)

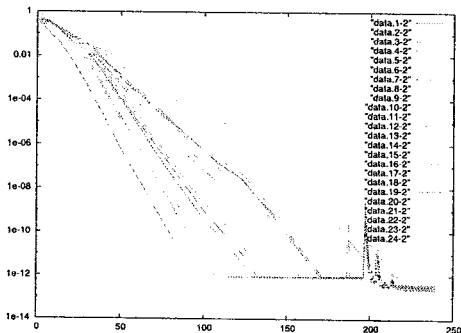


図 13 問題 b, デフレーション (1 あり, 2 あり)

5. 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、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行列乗算の実行タイミングや直交性等の各種パラメタの重要性が確認できた。これらは、収束性ならびに固有モードの過不足について重要なファクターであり適切なパラメタ選択が必要である。実行時間から判断するとパラメタ選択は必ずしも有効というわけではなく、問題に応じてパラメタ選択する仕組み=自動チューニングの応用がこの先の課題である。

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 Performance Preconditioner, http://www.llnl.gov/CASC/linear_solvers/.
- 6) G.Golub, Q.Ye: An Inverse Free Preconditioned Krylov Subspace Method for Symmetric Generalized Eigenvalue Problems, SIAM, J. Sci. Comput., Vol.24, No.1, pp.312-334, 2002.
- 7) PRIMME, PReconditioned Iterative Multi-Method Eigensolver, <http://www.cs.wm.edu/andreas/software/>.
- 8) 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.
- 9) 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.