

非対称固有値問題への並列 AMG 前処理付共役残差法の適用と評価

西 田 晃^{†,††}

大規模疎行列の固有値を数値的に求める場合、従来は Lanczos 法やその非対称問題への拡張である Arnoldi 法、あるいは量子化学分野由来の Davidson 法やその一種である Jacobi-Davidson 法などが用いられてきた。しかしながら、近年の研究により、共役勾配法を代数的マルチグリッド法 (AMG) などの適切な前処理手法と組み合わせることによって、他の手法と比較して高速に固有値を計算できることが明らかになってきた。本研究では、共役残差法と AMG 前処理を利用することにより、この手法が非対称固有値解法に拡張できることを示す。

Parallel AMG Preconditioned Conjugate Residual Methods for Nonsymmetric Eigenproblems and its Evaluation

AKIRA NISHIDA^{†,††}

When we need to compute the eigenvalues of a large sparse matrix numerically, the projection method such as the Lanczos/Arnoldi methods or the Davidson type methods from quantum chemistry is the most orthodox choice. However, recent studies revealed that the conjugate gradient type method combined with appropriate preconditioners can compute a few eigenpairs of such problems quite efficiently. In this paper, the combination of the conjugate residual type method and the AMG preconditioner is evaluated as the expansion of the conjugate gradient type method and shown to be the appropriate approach for large scale sparse nonsymmetric eigenproblems.

1. 背景

大規模疎行列の固有値を数値的に求める場合、いくつかの解法を考えることができる。主なものとしては、Lanczos 法やその非対称問題への拡張である Arnoldi 法、あるいは量子化学計算で利用されることの多い Davidson 法や、その一種である Jacobi-Davidson 法などの解法を挙げることができる。一般に、大規模疎行列の固有値問題では、最大または最小のものから数個までの固有値及び固有ベクトルを求めればよい場合が多い。

2. 共役勾配法の導入

ここでは、実対称行列 A , B に関する一般化固有値問題

$$Ax = \lambda Bx \quad (1)$$

の最小固有値、またはこれと同値な問題

$$Bx = \mu Ax, \quad \mu = 1/\lambda \quad (2)$$

の最大固有値 1 個を求めることを考える。これは Rayleigh 商

$$\mu(x) = \frac{x^T Bx}{x^T Ax} \quad (3)$$

の極値問題に帰着でき、最急勾配方向が

$$\nabla \mu(x) \equiv g(x) = \frac{2(Bx - \mu Ax)}{x^T Ax} \quad (4)$$

であることから、Rayleigh 商を局所的に最小化する係数 α_i を用いて、共役勾配法

$$x_{i+1} = x_i + \alpha_i p_i, \quad (5)$$

$$p_i = -g_i + \beta_{i-1} p_{i-1}, \quad \beta_{i-1} = \frac{g_i^T g_i}{g_{i-1}^T g_{i-1}} \quad (6)$$

により固有値を計算することができる⁴⁾。これは 1951 年に Hestenes らにより提案され、Fletcher らが発展させた手法であるが^{2),6)}、1980 年代以降の Knyazev らの研究^{7),8)}により、適切な前処理と組み合わせることによって高速に固有値を計算できることが分かってきた¹⁾。

一般に、前処理付固有値解法のアルゴリズムは、前処理行列 $T \approx A^{-1}$, TA , TB に関する m_k 次多項式 $P_{m_k}(TA, TB)$ を用いて以下のように書くことができる。

- (1) 初期ベクトル $x^{(0)}$ を選択する
- (2) m_k 回の反復により $x^{(k)} = P_{m_k}(TA, TB)x^{(0)}$ を計算する

[†] 東京大学大学院情報理工学系研究科コンピュータ科学専攻
Department of Computer Science, the University of Tokyo

^{††} 科学技術振興機構 CREST
CREST, Japan Science and Technology Agency

(3) Rayleigh 商 $\mu^{(k)} = (x^{(k)}, Bx^{(k)}) / (x^{(k)}, Ax^{(k)})$ を計算する

Rayleigh 商の計算には様々な方法を用いることができるが、ここでは共役勾配法の場合について説明する。前処理付共役勾配法の反復は、適当な初期ベクトル $x^{(0)}$ と対応する修正ベクトル $p^{(0)} = 0$ を用いて、

$$\mu^{(i)} = (x^{(i)}, Bx^{(i)}) / (x^{(i)}, Ax^{(i)}) \quad (7)$$

$$r = Bx^{(i)} - \mu^{(i)} Ax^{(i)} \quad (8)$$

$$w^{(i)} = Tr \quad (9)$$

$$x^{(i+1)} = w^{(i)} + \tau^{(i)} x^{(i)} + \gamma^{(i)} p^{(i)} \quad (10)$$

$$p^{(i+1)} = w^{(i)} + \gamma^{(i)} p^{(i)} \quad (11)$$

と書くことができる。Knyazev の手法では、ここで行列束 $Bx^{(i)} - \mu^{(i)} Ax^{(i)}$ に関する $\text{span}\{w, x^{(i)}, p^{(i)}\}$ 上の Ritz 値, Ritz ベクトルを Rayleigh-Ritz 法を用いて計算し、最大 Ritz 値に対応する Ritz ベクトルを $x^{(i+1)}$ とする。すなわち、係数 $\tau^{(i)}, \gamma^{(i)}$ の値は、 $\text{span}\{w, x^{(i)}, p^{(i)}\}$ 上での局所的な最適解をもとに決められる。これによって、ベクトル間の直交性をもとに各係数を明に計算する必要のある従来の方法と比較して、容易に更新値を求めることができる。

この解法は容易にブロック化することができ、初期ベクトルとして $x_1^{(0)}, \dots, x_m^{(0)}$ と対応する修正ベクトル $p_1^{(0)} = \dots = p_m^{(0)} = 0$ を取ることにより、 $j = 1, \dots, m$ について

$$\mu_j^{(i)} = (x_j^{(i)}, Bx_j^{(i)}) / (x_j^{(i)}, Ax_j^{(i)}) \quad (12)$$

$$r_j = Bx_j^{(i)} - \mu_j^{(i)} Ax_j^{(i)} \quad (13)$$

$$w_j^{(i)} = Tr_j \quad (14)$$

より、適当な係数 $\alpha^{(i)}$ を用いて

$$\text{span}\{w_1^{(i)}, \dots, w_m^{(i)}, x_1^{(i)}, \dots, x_m^{(i)}, p_1^{(i)}, \dots, p_m^{(i)}\} \quad (15)$$

上で j 番目に大きな Ritz 値と対応する j 番目の Ritz ベクトルとその修正ベクトル

$$x_j^{(i+1)} = \sum_{k=1, \dots, m} \alpha_k^{(i)} w_k^{(i)} + \tau_k^{(i)} x_k^{(i)} + \gamma_k^{(i)} p_k^{(i)} \quad (16)$$

$$p_j^{(i+1)} = \sum_{k=1, \dots, m} \alpha_k^{(i)} w_k^{(i)} + \gamma_k^{(i)} p_k^{(i)} \quad (17)$$

を計算する。出力として、最大固有値 μ_j と対応する固有ベクトルに関する近似 $\mu_j^{(k)}$ と固有ベクトル $x_j^{(k)}$ を $j = 1, \dots, m$ について得る。

3. 非対称問題への拡張

以上の手法を非対称問題に拡張することを考える。非対称固有値問題については共役残差法 (Orthomin(1)) が適用できることが知られている¹²⁾。ここではこれにつ

いて簡単に述べる。

行列 A, B が非対称な場合も、共役勾配法の代わりに Rayleigh 商以外の汎関数に対して共役残差法を適用することにより、勾配法系の反復解法で解くことができる。ここでは、固有値問題 (1) の残差を

$$r = \lambda Bx - Ax, \quad (18)$$

$$\lambda = (Ax, Bx) / (Bx, Bx) \quad (19)$$

で定義する。 x が固有値ベクトルに等しければ、 λ は対応する固有値に一致する。ここでは最小化すべき関数として残差 r の内積

$$F(r) = (r, r) \quad (20)$$

を選び、共役残差法 (Orthomin(1)) を適用する。すなわち、初期値 x_0 から

$$\lambda^{(0)} = (Ax^{(0)}, Bx^{(0)}) / (Bx^{(0)}, Bx^{(0)}), \quad (21)$$

$$r^{(0)} = \lambda^{(0)} Bx^{(0)} - Ax^{(0)}, \quad (22)$$

$$p^{(0)} = r^{(0)} \quad (23)$$

を求め、以下の反復

$$\alpha^{(i)} = [(r^{(i)}, Ap^{(i)}) - \lambda^{(i)}(r^{(i)}, Bp^{(i)})] / [(Ap^{(i)}, Ap^{(i)}) - 2\lambda^{(i)}(Ap^{(i)}, Bp^{(i)}) + (\lambda^{(i)})^2(Bp^{(i)}, Bp^{(i)})], \quad (24)$$

$$x^{(i+1)} = x^{(i)} + \alpha^{(i)} p^{(i)}, \quad (25)$$

$$\lambda^{(i+1)} = (Ax^{(i+1)}, Bx^{(i+1)}) / (Bx^{(i+1)}, Bx^{(i+1)}), \quad (26)$$

$$r^{(i+1)} = \lambda^{(i+1)} Bx^{(i+1)} - Ax^{(i+1)}, \quad (27)$$

$$\beta^{(i)} = -[(Ar^{(i+1)}, Ap^{(i)}) - \lambda^{(i+1)}\{(Ar^{(i+1)}, Bp^{(i)}) + (Ap^{(i)}, Br^{(i+1)})\} + (\lambda^{(i+1)})^2(Br^{(i+1)}, Bp^{(i)})] / [(Ap^{(i)}, Ap^{(i)}) - 2\lambda^{(i+1)}(Ap^{(i)}, Bp^{(i)}) + (\lambda^{(i+1)})^2(Bp^{(i)}, Bp^{(i)})], \quad (28)$$

$$p^{(i+1)} = r^{(i+1)} + \beta^{(i)} p^{(i)} \quad (29)$$

を相対残差

$$\epsilon^{(i)} = \frac{\| \lambda^{(i)} Bx^{(i)} - Ax^{(i)} \|_2}{\| \lambda^{(i)} Bx^{(i)} \|_2} \quad (30)$$

が十分小さくなるまで繰り返す。

4. 前処理

一般化固有値問題

$$Ax = \lambda Bx \quad (31)$$

において、固有値 λ が既知であると仮定すると、これに対応する固有ベクトルは

$$(A - \lambda B)x = 0, \quad x \neq 0 \quad (32)$$

を解くことにより求めることができる。すなわち、固有値解法における理想的な前処理行列 T は $A - \lambda B$ の逆行列であり、実際には未知の λ を Ritz 値で置き換えた前処理行列を考えることもできる。しかしながら、一般

ここでは正確な結果を要しないので、直交化は行わない。

にこのように前処理行列を取ると不定値となることが多く、

$$T \approx A^{-1} \quad (33)$$

と取るのが自然であり、特に連立一次方程式 $Ax = f$ の解法が与えられている場合には、前処理の計算も容易である。このように定めた T に関して、

$$w^{(i)} = Tr \quad (34)$$

すなわち

$$T^{-1}w^{(i)} = r \quad (35)$$

を解く。

解くべき固有値問題における行列の性質が分かっている場合には、これを利用して適切な前処理手法を選択することができる。問題が非対称である場合、以下のような前処理が考えられる。

4.1 不完全 LU 分解

上記の共役残差法のアルゴリズムから、行列 A を

$$A = LDU + E \quad (36)$$

の形に不完全 LU 分解した結果から、左からの前処理

$$(LDU)^{-1}Ax = \lambda(LDU)^{-1}Bx \quad (37)$$

と右からの前処理

$$\begin{aligned} & A(LDU)^{-1}(LDUx) \\ & = \lambda B(LDU)^{-1}(LDUx) \end{aligned} \quad (38)$$

を考えることができる¹²⁾。前者は上記の共役残差法のアルゴリズムにおいて係数行列 A , B を $(LDU)^{-1}A$, $(LDU)^{-1}B$ で、また後者は直交係数 β と修正方向 p をそれぞれ

$$\begin{aligned} \beta^{(i)} = & -[(A(LDU)^{-1}r^{(i+1)}, Ap^{(i)}) \\ & -\lambda^{(i+1)}\{(A(LDU)^{-1}r^{(i+1)}, Bp^{(i)}) \\ & + (Ap^{(i)}, B(LDU)^{-1}r^{(i+1)})\} \\ & + (\lambda^{(i+1)})^2(B(LDU)^{-1}r^{(i+1)}, Bp^{(i)})] \\ & / [(Ap^{(i)}, Ap^{(i)}) - 2\lambda^{(i+1)}(Ap^{(i)}, Bp^{(i)}) \\ & + (\lambda^{(i+1)})^2(Bp^{(i)}, Bp^{(i)})], \end{aligned} \quad (39)$$

$$p^{(i+1)} = (LDU)^{-1}r^{(i+1)} + \beta^{(i)}p^{(i)}. \quad (40)$$

で置き換えたものに相当する。数値実験¹²⁾の結果から、前処理の結果非対称性が増大する場合に性能に影響が生じることがあるものの、いずれの場合にも収束することが分かっているが、これらの方法は LU 分解を伴うため、並列化が困難である。

4.2 代数的マルチグリッド法

代数的マルチグリッド法は、問題行列の代数的な性質のみをもとにアルゴリズムを構成するマルチグリッド法の一つである。問題の物理的性質に依存しない利点があり、固有値解法の前処理アルゴリズムとしても適しているといえる。共役勾配法系の固有値解法において、代数的マルチグリッド前処理が有効であることは既に知られており¹⁰⁾、実装も行われているが、非対称固有値問題に

関連した事例は報告されていない。以下ではまず、共役残差法に関して代数的マルチグリッド前処理が可能であるかどうかを検討する。

問題 $Ax = f$ に対する代数的マルチグリッド法の記法を以下に定義する。 A の次数を $n \times n$ 、要素を a_{ij} 、点 i での u の値を u_i 、また格子は $\Omega = \{1, 2, \dots, n\}$ で表すものとする。細かい格子上で残った誤差 e は、粗い格子上で残差方程式 $Ae = r$ を解くことにより取り除かれ、結果は補間されて細かい格子に $u \leftarrow u + e$ として戻される。以下では添字によりレベル番号を示す。すなわち格子の包含関係は $\Omega^1 \supset \Omega^2 \supset \dots \supset \Omega^M$ であり、各格子上での演算子は A^1, A^2, \dots, A^M である。格子間の移動は

$$\text{補間: } I_{k+1}^k, k = 1, 2, \dots, M-1, \quad (41)$$

$$\text{縮約: } I_k^{k+1}, k = 1, 2, \dots, M-1 \quad (42)$$

によって実現され、各段で適当な緩和を行う。各段は以下のように再帰的に定義される。

Algorithm $MV^k(u^k, f^k)$. The (μ_1, μ_2) V-cycle.
If $k = M$, set $u^M = (A^M)^{-1}f^M$.

Otherwise:

Relax μ_1 times on $A^k u^k = f^k$.

Perform coarse-grid correction:

Set $u^{k+1} = 0$, $f^{k+1} = I_k^{k+1}(f^k - A^k u^k)$.

Solve on level $k+1$ with $MV^{k+1}(u^{k+1}, f^{k+1})$.

Correct the solution by $u^k \leftarrow u^k + I_{k+1}^k u^{k+1}$.

Relax μ_2 times on $A^k u^k = f^k$.

一方、各段の要素の選択は事前に行われる。

AMG Setup Phase.

- (1) Set $k = 1$.
- (2) Partition Ω^k into disjoint sets C^k and F^k .
 - (a) Set $\Omega^{k+1} = C^k$.
 - (b) Define interpolation I_{k+1}^k .
- (3) Set $I_k^{k+1} = (I_{k+1}^k)^T$ and $A^{k+1} = I_k^{k+1} A^k I_{k+1}^k$.
- (4) If Ω^{k+1} is small enough, set $M = k+1$ and stop. Otherwise, set $k = k+1$ and go to step 2.

古典的なアルゴリズムでは、粗い格子の選択は 2 段階に分けることができる。すなわち、はじめにある規則にしたがって粗な点 C とそれ以外の点 F を分けた後、補間演算子が条件を満たすよう F の一部を C に戻して調整する。ここでの目標は、粗い格子点の集合 C 、及び各 $i \in F \equiv \Omega - C$ について補間点の集合 $C_i \in C$ を選ぶことである。

C を選ぶ際、ある点 i において、 i の値が影響を受ける隣接点の集合を S_i とする。古典的な AMG アルゴリズムでは

$$S_i = \{j \neq i : -a_{ij} \geq \alpha \max_{k \neq j}(-a_{ik})\}, \quad (43)$$

で定義される。すなわち、以下の基準に従って C と F とを分けることになる。

(C1) 各点 $i \in F$ について、点 $j \in S_i$ は C に含まれるか、または $S_j \cap C_i \neq \emptyset$ である。

(C2) C は、 C に含まれる点のそれぞれが C の他の点の値に依存しない最大部分集合でなくてはならない。

(C1) は、 i が j に強く依存している場合に、(j が C に含まれていなくても) u_j の値が u_i に反映されることを保証し、(C2) は粗い格子点の量を調整する。古典的 AMG アルゴリズムでは (C1) は必須であるが、(C2) は必ずしも満たされない。

縮約の手順は以下の通りである。まず各点 i に、 i の影響を強く受ける他の点の数 l_i を割り当てる。次に最大の l を持つ点を最初の C として選ぶ。点 i に強く影響を受けるものは F となり、これらの F に影響を与えるその他の点は潜在的に C となる。これはすべての点が C か F に割り当てられるまで続く。図 1 に Dirichlet 境界条件、規則格子上の 9 点 Laplacian の場合を示す。

ただし、図 2 に示した例のように、1 回の選択で C に接続しない F が生じる場合には、それらに係る F - F の依存関係を調べ、どちらかを C とする。

代数的マルチグリッド法にはいくつかのアルゴリズムがあり、並列化に関しても複数の手法が知られている¹¹⁾。2002 年に Lawrence Livermore 米国立研究所の Van Emden Henson らによって提案、開発された BoomerAMG⁵⁾ は、代数的マルチグリッド法 (Algebraic Multigrid Method) の並列実装の一種であり、非対称行列にも適用が可能である。以下ではまずアルゴリズムについて簡単に述べる。

4.3 BoomerAMG

BoomerAMG には複数の縮約アルゴリズムが含まれており、このうち最も簡単なものが、以下に述べる Cleary-Luby-Jones-Plassman (CLJP) 並列縮約アルゴリズム³⁾ である。このアルゴリズムは Luby⁹⁾ によって提案された並列グラフ分割アルゴリズムをもとにしている。まず、以下で定義される影響行列 S を定義する。

$$S_{ij} = \begin{cases} 1 & \text{if } j \in S_i, \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

すなわち、 i が j に依存している場合のみ $S_{ij} = 1$ となり、 S の第 i 行は i の依存関係を示す集合 S_i を与え、第 i 列は i の影響を示す集合 S_i^T を与える。(有向グラフ S において、節点 i から j への枝は、 $S_{ij} \neq 0$ の場合にのみ存在する。)

各点 i について、 i に影響を受ける点の数と $(0, 1)$ 内の乱数の和 $w(i) = |S_i^T| + \sigma(i)$ なる量を定義し、すべ

ての $k \in S_i \cap S_i^T$ について $w(i) > w(k)$ を満たす点 i の集合を D と定義する。独立集合 D から、以下の発見的基準をもとにグラフを修正する。

(H1) C の点の値は補間されない。したがって、 C に影響を与える隣接点は C よりも重要度が低い。

(H2) k, j の双方が C の点 i に依存し、また j が k に影響を及ぼしているならば、 k は i から補間されるので j は C の点としては k ほど重要ではない。

これをまとめると、

```
for each  $i \in D$ ,
  for each  $j$  that influences  $i$ ,
    decrement  $w(j)$ 
    set  $S_{ij} \leftarrow 0$  (remove edge  $ij$  from the graph)
```

```
for each  $j$  that influences  $i$ ,
  set  $S_{ji} \leftarrow 0$  (remove edge  $ji$  from the graph)
  for each  $k$  that depends on  $j$ ,
    if  $k$  depends on  $i$ ,
      decrement  $w(j)$ 
      set  $S_{kj} \leftarrow 0$  (remove edge  $kj$  from the graph)
```

(H1), (H2) を D に適用し、修正されたグラフ S をもとに、すべての点が C または F として分類されるまでこれを繰り返す。図 3 に、CLJP アルゴリズムによる規則格子上の 9 点 Laplacian の色分けを示す。

CLJP アルゴリズムの利点はその並列性にあり、(同じ乱数の集合を与える限り) 並列度に関わらず同一の粗な格子点集合を与える。

前述の逐次のアルゴリズムは Ruge-Stüben アルゴリズムと呼ばれ、こちらの場合でも領域を分割して適用することができる。また、各領域内部において RS アルゴリズムを、また境界上で CLJP アルゴリズムを適用することにより、CLJP の縮約の遅さを解決したものを Falgout アルゴリズムと呼ぶ。

これらを非対称問題

$$-\nabla^2 u + c(u_x + u_y + u_z) = f \quad (45)$$

に適用した結果を表 1 に示す⁵⁾。問題サイズはプロセス当たり 40^3 、 $\alpha = 0.75$ として計算している。使用した計算機は ASCI Blue Pacific である。GMRES や BiCGSTAB 等と比較しても、良好に収束していることが分かる。

5. む す び

本稿では、非対称固有値解法への前処理付共役残差法の適用の可能性について、予備的な評価をもとに検討し、非対称問題においても十分な性能が得られると考えられることを示した。代数的マルチグリッド前処理は比較的新しい解法であり、収束条件などに関して未解明な部分も残っている。今後とも特性を明らかにしていくとも

α は通常 0.25 と取ることが多い。

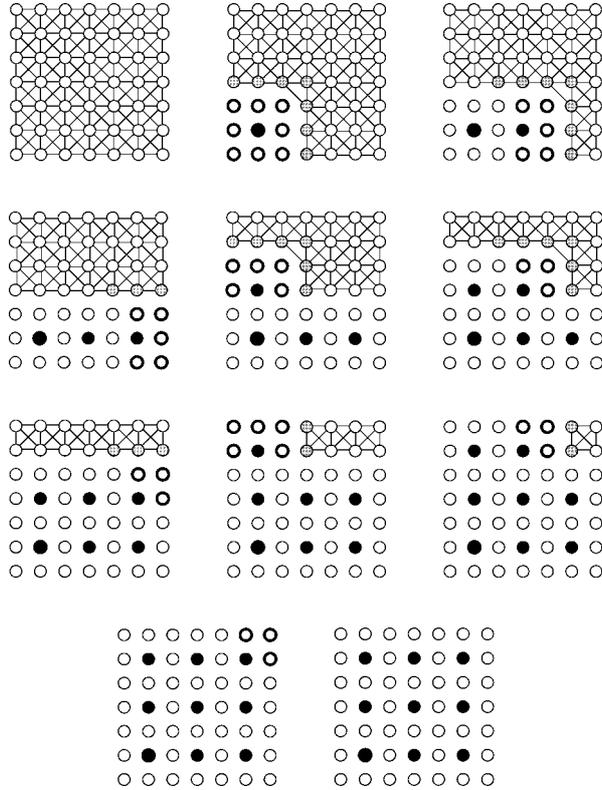


図1 規則格子上の9点 Laplacian の場合の色分け

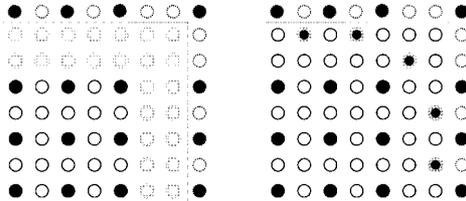


図2 周期境界条件, 規則格子上の9点 Laplacian の場合の色分け

表1 非対称問題での反復回数

p	Falgout	RS	GMRES(10)	BiCGSTAB
1	5	5	143	77
8	7	7	273	144
27	7	8	398	226
64	9	10	534	323
125	8	10	642	367
216	9	12	793	493
343	8	11	842	500
512	9	14	1046	644

に, 大規模固有値解法に対する有力な解法の一つとして, 効果的な実装手法について検討を進めていきたい。

謝辞 本研究の一部は, 科学技術振興機構戦略的創造

研究推進事業 (CREST) 「大規模シミュレーション向け基盤ソフトウェアの開発」及び科学研究費基盤研究 (C) 15607005 「非構造多重格子を用いた離散化手法とその効率的な並列実装技術に関する研究」によるものである。

参考文献

- 1) P. ARBENZ AND R. LEHOUCQ, *A comparison of algorithms for modal analysis in the absence of a sparse direct method*, Tech. Rep. SAND2003-1028J, Sandia National Laboratories, 2003.
- 2) W. W. BRADBURY AND R. FLETCHER, *New Iterative Method for Solution of the Eigenproblem*, Numer. Math., 9 (1966), pp. 259–267.

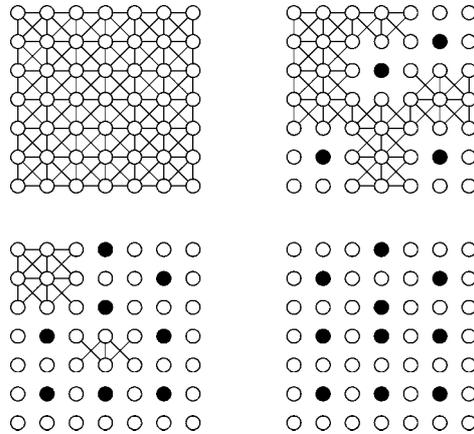


図3 左上のグラフをもとに、独立点集合の計算とその修正を2回(右上, 左下)繰り返した結果(右下).

- 3) A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, AND J. E. JONES, *Coarse-grid selection for parallel algebraic multigrid*, in Workshop on Parallel Algorithms for Irregularly Structured Problems, 1998, pp. 104–115.
- 4) R. FLETCHER AND C. M. REEVES, *Function minimization by conjugate gradients*, *Comp. J.*, 7 (1964), pp. 149–154.
- 5) V. E. HENSON AND U. M. YANG, *Boomer-AMG: A parallel algebraic multigrid solver and preconditioner*, *Applied Numerical Mathematics: Transactions of IMACS*, 41 (2002), pp.155–177.
- 6) M. R. HESTENES AND W. KARUSH, *A method of gradients for the calculation of the characteristic roots and vectors of a real symmetric matrix*, *J. Res. Natl. Inst. Stand. Technol.*, 47 (1951), pp. 45–61.
- 7) A.V. KNYAZEV, *Preconditioned eigensolvers—an oxymoron?*, *Electron. Trans. Numer. Anal.*, 7 (1998), pp. 104–123 (electronic). Large scale eigenvalue problems (Argonne, IL, 1997).
- 8) ———, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, *SIAM J. Sci. Comput.*, 23 (2001), pp. 517–541.
- 9) M. LUBY, *A simple parallel algorithm for the maximal independent set problem*, *SIAM J. Comput.*, 15 (1986), pp. 1036–1053.
- 10) 西田晃, 大規模固有値問題への並列 AMG 前処理付 共役勾配法の適用と評価, *情処研報*, 2004 (2004), pp. 25–30.
- 11) 藤井昭宏, 西田晃, 小柳義夫, *Smoothed Aggregation MG 法の異方性問題への対応と評価*, 2003 年先進的計算基盤システムシンポジウム論文集, pp. 137–144.
- 12) 末富英一, 関本博, 多群中性子拡散方程式に現れる非対称行列系の一般固有値問題に対する ORTHOMIN(1) 法の適用, *情報処理学会論文誌*, 30 (1989), pp. 661–667.