社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ARC－177 (34)
2008－HPC－114 (34)
2008／3／6

線形計算ライブラリを用いた共役勾配法系固有値解法の実装と評価

西 田 晃 †

大規模疎行列の固有値を数値的に求める場合, 共役勾配法による求解が効率的であることが知られている. 共役勾配法を適切な前処理手法と組み合わせることによって, より高速に固有値を計算することができるが, 評価に際してはこれらのライブラリを用意する必要がある. 本研究では, 疎行列向けの代表的な固有値解法について整理するとともに, 線形計算ライブラリ Lis を用いた共役勾配法系固有値解法の実装結果について報告する.

Implementation and Evaluation of Conjugate Gradient based Eigensolvers using Linear Algebra Library

AKIRA NISHIDA †

When we need to compute some eigenvalues of a large sparse matrix numerically, the conjugate gradient method is known to be the optimal choice. Combining appropriate preconditioners with the conjugate gradient method, you can compute the eigensolutions more efficiently, although the evaluation of preconditioners needs pre-built linear algebra libraries. This study summarizes the existing major eigensolvers for sparse matrices and reports the results of the implementation of the conjugate gradient based eigensolvers using parallel linear algebra library Lis.

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, 冪乗法, Lanczos 法, (Jacobi-) Davidson 法等の Krylov 部分空間法が提案されている. 多くの場合, 最大または最小のものから数個の固有値 (及び固有ベクトル) を求めればよいため, 疎行列性を保存するこれらの解法は, 実アプリケーションにおいて広く用いられている.

一方, これとは独立に, 共役勾配法による求解を考えることができ, 高並列な計算機環境に適していることから, 広く知られるようになってきた[1),2)]. 共役勾配法は連立一次方程式解法との共通項が多く, 適切な前処理手法と組み合わせることにより, 高速に固有値を計算することができる. 本研究ではこの点を踏まえ, 既存の解法について整理するとともに, 線形計算ライブラリ Lis を利用した共役勾配法系固有値解法の実装とその評価結果について報告する.

冪乗法・部分空間反復法

これらの方法では, サイズ n の大規模行列 A の固有値を絶対値最大のものから☆ 求めるため, これを次元 $\nu \ll n$ のベクトル部分空間 G_l 上への直交射影に関する固有値問題に近似する. 射影を表す行列を π_l とすれば, 問題は G_l において

$$\pi_l(Ax_l - \lambda_l x_l) = 0 \qquad (1)$$

を満たす近似固有対

$$\lambda_l \in C, \quad 0 \neq x_l \in G_l \qquad (2)$$

の計算に帰着される. 部分空間 S を生成する r 個の独立なベクトルを

$$U = [u_1, ..., u_r], \quad 1 \leq r < n \qquad (3)$$

とすると, G_l 内の正規直交基底の選び方によって,

$$S_l = A^l S, \quad l = 1, 2, ... \qquad (4)$$

とすれば $r = 1$ の場合は冪乗法, また $r > 1$ ならば部分空間反復法となる.

Lanczos/Arnoldi 法

$\{u_1, ..., u_r\}$ によって生成される Krylov 部分空間を

$$\mathcal{K}_l = \mathrm{lin}(S, AS, ..., A^{l-1}S) \qquad (5)$$

とすると, A が Hermite 行列ならば Lanczos 法 ($r = 1$), ブロック Lanczos 法 ($r > 1$) が, 同様に A が非 Hermite 行列ならば Arnoldi 法, ブロック

† 中央大学／東京大学／科学技術振興機構
Chuo University / The University of Tokyo / Japan Science and Technology Agency

☆ 以下単に最大固有値と書く.

Arnoldi 法が得られる．これらの方法では，写像が三重対角行列または Hessenberg 行列 H_l で表される $\mathcal{K}_l$ の正規直交基底 $\{v_i\}_1^l$ を求め，H_l の固有対を元の問題の近似解として計算する．

(Jacobi-) Davidson 法

Davidson 法では，正規直交基底 $\{v_i\}_1^k$ で張られる部分空間 $\mathcal{K} = \mathrm{span}\{v_1, ..., v_k\}$ 上で行列 A の Ritz 値 θ_k 及び Ritz ベクトル u_k を計算する際に，残差 $r = Au_k - \theta_k u_k$ に関する修正方程式

$$M_k t = r, \quad M_k = D_A - \theta_k I \tag{6}$$

を解いて u_k を更新し，$\mathcal{K}$ の次元を拡張する．D_A は A の対角成分である．具体的には，t を $\mathcal{K}$ と直交化して v_{k+1} を得る．$V_{k+1} = [v_1, ..., v_{k+1}]$ と置けば，新しい Ritz 対 (θ_{k+1}, u_{k+1}) は行列

$$H_{k+1} = V_{k+1}^* A V_{k+1} \tag{7}$$

の固有対として計算されることになる．$M_k = I$ の場合には Lanczos/Arnoldi 法と同一になるので，Davidson 法は加速付 Lanczos/Arnoldi 法の一種と考えることができる．なお，

$$M_k^{-1} \approx (A - \theta_k I)^{-1} \tag{8}$$

を残差ベクトル r に対する前処理行列と考えると，この方法では θ_k に対応する近似固有ベクトル u_k の方向の成分を増幅させることになり，対角優位な行列の最大固有値を求める場合にしか顕著な効果が得られないことが分かっている[3]．このため，Jacobi-Davidson 法では u_k の直交補空間から更新のための成分を取り出すことによってこの問題点を解消している[4]．これらの解法は，固有ベクトルが必要な場合には得られた固有値をもとに逆反復法等により計算することになるため，余分な計算量を要する．

2. 共役勾配法系固有値解法

これに対して，実対称行列 A, B に関する一般化固有値問題

$$Ax = \lambda Bx \tag{9}$$

の最小固有値，またはこれと同値な問題

$$Bx = \mu Ax, \quad \mu = 1/\lambda \tag{10}$$

の最大固有値の計算を

Rayleigh 商

$$\mu(x) = \frac{x^T Bx}{x^T Ax} \tag{11}$$

の極値問題に帰着し，最急勾配方向が

$$\nabla\mu(x) \equiv g(x) = \frac{2(Bx - \mu Ax)}{x^T Ax} \tag{12}$$

であることから，適当な係数 α_i を用いて

$$x_{i+1} = x_i + \alpha_i p_i, \tag{13}$$

から最急上昇法を，また，修正方向 p として

$$p_i = -g_i + \beta_{i-1}p_{i-1}, \ \beta_{i-1} = \frac{g_i^T g_i}{g_{i-1}^T g_{i-1}} \tag{14}$$

などを用いることにより，共役勾配法を導くことができる[5]．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[6),7]，1980 年代以降の Knyazev らの研究[8)~10]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明らかになった．

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

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

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，

$$\mu^{(i)} = (x^{(i)}, Bx^{(i)})/(x^{(i)}, Ax^{(i)}) \tag{15}$$

$$r = Bx^{(i)} - \mu^{(i)} Ax^{(i)} \tag{16}$$

$$w^{(i)} = Tr \tag{17}$$

$$x^{(i+1)} = w^{(i)} + \tau^{(i)} x^{(i)} + \gamma^{(i)} p^{(i)} \tag{18}$$

$$p^{(i+1)} = w^{(i)} + \gamma^{(i)} p^{(i)} \tag{19}$$

と書くことができる．ここでは行列束 $Bx^{(i)} - \mu^{(i)} Ax^{(i)}$ に関する $\mathrm{span}\{w, x^{(i)}, p^{(i)}\}$ 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，$\mathrm{span}\{w, x^{(i)}, p^{(i)}\}$ 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，少ない計算量で更新値を求めることができる．

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

$$\mu_j^{(i)} = (x_j^{(i)}, Bx_j^{(i)})/(x_j^{(i)}, Ax_j^{(i)}) \tag{20}$$

$$r_j = Bx_j^{(i)} - \mu_j^{(i)} Ax_j^{(i)} \tag{21}$$

$$w_j^{(i)} = Tr_j \tag{22}$$

より，適当な係数 $\alpha^{(i)}$ を用いて

$$\mathrm{span}\{w_1^{(i)}, ..., w_m^{(i)}, x_1^{(i)}, ..., x_m^{(i)}, p_1^{(i)}, ..., p_m^{(i)}\} \tag{23}$$

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

$$x_j^{(i+1)} = \sum_{k=1,\dots,m} \alpha_k^{(i)} w_k^{(i)} + \tau_k^{(i)} x_k^{(i)} + \gamma_k^{(i)} p_k^{(i)} \tag{24}$$

$$p_j^{(i+1)} = \sum_{k=1,\dots,m} \alpha_k^{(i)} w_k^{(i)} + \gamma_k^{(i)} p_k^{(i)} \tag{25}$$

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 $j=1,...,m$ について得る．

なお，最急上昇法及び共役勾配法の収束率については，冪乗法との関係から条件

$$\delta_0(T^{-1}x,x) \le (Ax,x) \le \delta_1(T^{-1}x,x) \tag{26}$$

のもとで

$$\frac{\mu_1-\mu^{(n)}}{\mu^{(n)}-\mu_2} \le (1-\xi)^n \frac{\mu_1-\mu^{(0)}}{\mu^{(0)}-\mu_2},$$
$$\xi = \delta_0/\delta_1 \cdot (\mu_1-\mu_2)/(\mu_1-\mu_{\min}) \tag{27}$$

の関係が成り立つ[8]．

3. 非対称問題への拡張

非対称問題への拡張に関しては，2 つの手法が提案されている．今回これらの評価は行わなかったが，今後 Lis 上に実装する予定である．

3.1 共役残差法による方法

行列 A, B が非対称な場合には，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方法が提案されている[2),11)]．ここでは，固有値問題の残差を

$$r = \lambda Bx - Ax, \tag{28}$$
$$\lambda = (Ax,Bx)/(Bx,Bx) \tag{29}$$

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

$$F(r) = (r,r) \tag{30}$$

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

$$\lambda^{(0)} = (Ax^{(0)},Bx^{(0)})/(Bx^{(0)},Bx^{(0)}), \tag{31}$$
$$r^{(0)} = \lambda^{(0)}Bx^{(0)} - Ax^{(0)}, \tag{32}$$
$$p^{(0)} = r^{(0)} \tag{33}$$

を求め，以下の反復

$$\begin{aligned}\alpha^{(i)} &= [(r^{(i)},Ap^{(i)}) - \lambda^{(i)}(r^{(i)},Bp^{(i)})]\\ &\quad / \ [(Ap^{(i)},Ap^{(i)}) - 2\lambda^{(i)}(Ap^{(i)},Bp^{(i)})\\ &\quad +(\lambda^{(i)})^2(Bp^{(i)},Bp^{(i)})],\end{aligned} \tag{34}$$
$$x^{(i+1)} = x^{(i)} + \alpha^{(i)}p^{(i)}, \tag{35}$$
$$\lambda^{(i+1)} = (Ax^{(i+1)},Bx^{(i+1)})/(Bx^{(i+1)},Bx^{(i+1)}), \tag{36}$$
$$r^{(i+1)} = \lambda^{(i+1)}Bx^{(i+1)} - Ax^{(i+1)}, \tag{37}$$
$$\begin{aligned}\beta^{(i)} &= -[(Ar^{(i+1)},Ap^{(i)})\\ &\quad -\lambda^{(i+1)}\{(Ar^{(i+1)},Bp^{(i)})\\ &\quad +(Ap^{(i)},Br^{(i+1)})\}\\ &\quad +(\lambda^{(i+1)})^2(Br^{(i+1)},Bp^{(i)})]\\ &\quad / \ [(Ap^{(i)},Ap^{(i)}) - 2\lambda^{(i+1)}(Ap^{(i)},Bp^{(i)})\\ &\quad +(\lambda^{(i+1)})^2(Bp^{(i)},Bp^{(i)})],\end{aligned} \tag{38}$$
$$p^{(i+1)} = r^{(i+1)} + \beta^{(i)}p^{(i)} \tag{39}$$

を相対残差

$$\begin{aligned}\epsilon^{(i)} &= \| \lambda^{(i)}Bx^{(i)} - Ax^{(i)} \|_2\\ &\quad / \ \| \lambda^{(i)}Bx^{(i)} \|_2\end{aligned} \tag{40}$$

が十分小さくなるまで繰り返す．

3.2 共役勾配法による方法

共役勾配法系固有値解法において，行列 A, B の対称性に関する制約は使われていないため，Arnoldi 法と同様な原理により，絶対値に関する最大固有値を共役勾配法により求めることができ，また実装も共役残差法による方法と比較して容易である．この場合，対称問題で求めた最大 Ritz 値の代わりに，絶対値最大の Ritz 値を求めればよい[12)]．

4. 前 処 理

一般化固有値問題

$$Ax = \lambda Bx \tag{41}$$

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

$$(A-\lambda B)x = 0, \quad x \ne 0 \tag{42}$$

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．一般にこのように前処理行列を取ると不定値となるため，T が対称正定値でなければならない場合には

$$T \approx A^{-1} \tag{43}$$

と取ればよい．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，

$$w^{(i)} = Tr \tag{44}$$

すなわち

$$T^{-1}w^{(i)} = r \tag{45}$$

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．

共役残差法のアルゴリズムにおいては，行列 A を

$$A = LDU + E \tag{46}$$

の形に不完全 LU 分解した結果から，左からの前処理

$$(LDU)^{-1}Ax = \lambda(LDU)^{-1}Bx \tag{47}$$

と右からの前処理

$$\begin{aligned} &A(LDU)^{-1}(LDUx) \\ &= \lambda B(LDU)^{-1}(LDUx) \end{aligned} \tag{48}$$

が提案されている[11]．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

$$\begin{aligned} \beta^{(i)} = &-[(A(LDU)^{-1}r^{(i+1)}, Ap^{(i)}) \\ &-\lambda^{(i+1)}\{(A(LDU)^{-1}r^{(i+1)}, Bp^{(i)}) \\ &+(Ap^{(i)}, B(LDU)^{-1}r^{(i+1)})\} \\ &+(\lambda^{(i+1)})^2(B(LDU)^{-1}r^{(i+1)}, Bp^{(i)})] \\ &/\ [(Ap^{(i)}, Ap^{(i)}) - 2\lambda^{(i+1)}(Ap^{(i)}, Bp^{(i)}) \\ &+(\lambda^{(i+1)})^2(Bp^{(i)}, Bp^{(i)})], \end{aligned} \tag{49}$$

$$p^{(i+1)} = (LDU)^{-1}r^{(i+1)} + \beta^{(i)}p^{(i)}. \tag{50}$$

で置き換えたものに相当する．数値実験[11] 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．このような場合には，代数的マルチグリッド前処理[13] 等の前処理が適していると考えられる．

5. Lis を用いた実装

既存の線形計算ライブラリはいくつかあるが，本研究では，科学技術振興機構 CREST 事業の一環として開発を進めている Lis ☆を利用し，この上に固有値解法を実装した．Lis は複数の行列格納形式に対応した並列反復解法ライブラリであり，反復解法向けの基本演算ルーチンや，先述した代数的マルチグリッド前処理を含む前処理法を実装している．

海外においても，米 Lawrence Berkeley 国立研究所により開発された疎行列向け並列前処理付線形解法のためのアルゴリズムライブラリ Hypre に，対称問題向けの共役勾配法系固有値解法 LOBPCG が実装されている．Lis, Hypre ではオブジェクト指向に基づいた設計を行っており，すべての処理は関数を用いて定義されている．すなわち，行列，ベクトルデータ等の生成・廃棄，及びこれらのオブジェクトに対する操作は，それぞれの操作を記述する関数を呼び出すことにより処理される．各前処理手法はそれぞれ線形解法として実装され，必要に応じて前処理として利用することができるようになっており，これらを組み合わせて評価することができる．

Knyazev らによる共役勾配法の実装であるブロック版アルゴリズム LOBPCG は，Hypre の固有値計算アルゴリズムとして収録されているものの，基本的に独立したプログラムであり，前処理に関する Hypre のライブラリのみを必要に応じて使用している．現時点では対称正定値な疎行列のみに対応している．反復解法部は行列等のデータに関する基本的な処理を記述した関数を用いて記述されており，MPI を用いた並列化はこのレベルで行われている．

新たに開発が必要となるのは Rayleigh 商の計算に関する部分であるが，固有値と共に固有ベクトルを同時に求める必要があり，解くべき固有値問題のサイズが非常に小さいことから，本研究ではこの部分を冪乗法で実装している．なお，Hypre ではこの部分に QR 法の Fortran ライブラリを利用している．

本研究では，Lis を用いて共役勾配法系固有値解法を実装し，評価を行った．Lis ではオブジェクト指向に基づいた設計を行っており，すべての処理は関数を用いて定義されている．すなわち，行列，ベクトルデータ等の生成・廃棄，及びこれらのオブジェクトに対する操作は，それぞれの操作を記述する関数を呼び出すことにより逐次または並列に処理される．また，表 1 の各前処理手法はそれぞれ近似連立一次方程式解法として実装され，これらと組み合わせて固有値解法を実装することができる．

表 1 Lis で利用可能な前処理

Jacobi
SSOR
ILU(k)
ILUT
Crout ILU
I+S
SA-AMG
Hybrid
SAINV
Additive Schwarz
ユーザ定義前処理

実装した共役勾配法は，Lis の固有値計算アルゴリズムとして収録され，要素演算と前処理に関するライブラリを必要に応じて使用している．現時点では対称問題に対応しており，対称正定値な疎行列の固有値問題を解くことができる．図 2 にその構成を示す．反復解法部は行列等のデータに関する要素演算として定義された Lis の関数を用いて記述されており，MPI を用いた並列化はこ

☆ http://www.ssisc.org/lis/

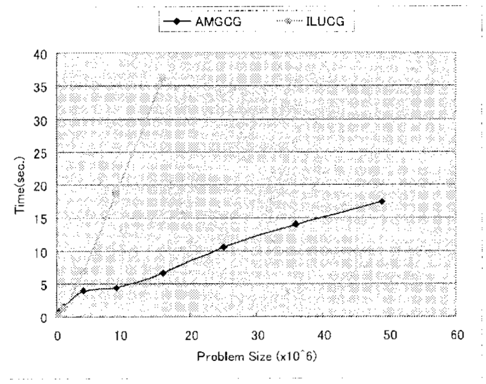

図1 AMGCG の Blue Gene/L 1024 ノード上での評価結果

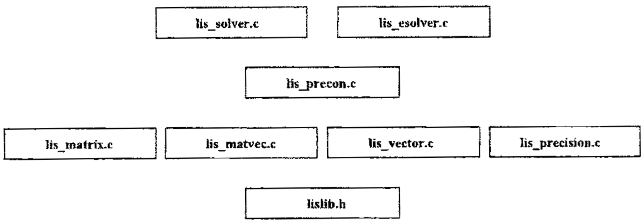

図2 固有値解法ライブラリ lis_esolver.c を含めた Lis の階層構造

のレベルで行われている.

Lis に実装されている前処理手法について, 主なものを簡単に述べる.

- 対角スケーリング・Jacobi 前処理
  これらは前処理としては単純な方法であるが, スケーラビリティに関して良好な性能を示す.
- SSOR・Hybrid 前処理
- 不完全 LU 分解前処理
  ILU 前処理のスケーラブルな実装. 問題サイズがプロセッサ数に比例する場合に, ほぼ一定の計算時間で処理することができる. ILU(k), ILUT, Crout ILU が実装されている.
- SA-AMG[13] 前処理
  代数的マルチグリッド法の逐次・並列実装. Lis では前処理の一つとして実装され, クリロフ部分空間法の前処理として用いることができる. 図1に SA-AMG 前処理を用いた連立一次方程式向け共役勾配法の Blue Gene/L 1024 ノード上での評価結果を示す[14].
- 近似逆行列前処理
- Additive Schwarz 前処理

6. 性能評価

ここでは, Lis の性能について調べるため, SGI Altix 3700 (1.3GHz Intel Itanium 2 processor × 32) を用いて評価を行った. コンパイラには Intel Compiler 9.1 for Linux を使用した.

一般にマルチグリッド前処理は反復回数が行列サイズに依存しないため, 大規模な問題ほど他の手法と比較して有利となる. Lis では代数的マルチグリッド法に関して, 対称, 非対称問題 (構造は対称) の双方を選択することができる. ここでは, 1次元 Poisson 問題の係数行列について, 最大固有値を AMG 前処理付き共役勾配法により計算した. 相対残差ノルムの閾値を 10^{-5} とし, 非零要素数がプロセッサ数に比例するよう問題サイズを取って反復解法部の実行時間を測定した結果を表2に示す.

これらの結果から, スケーラビリティに関しても充分な性能が得られていることが分かる.

7. むすび

本稿では, 共役勾配法系固有値解法及びその前処理法について, 本研究でこれまでに得られている知見をまとめるとともに, 線形計算ライブラリを用いた実装結果に

表2 AMG 前処理付共役勾配法の計算時間

Problem Size	10,000	20,000	40,000	80,000	160,000
# CPUs	2	4	8	16	32
time (s)	1.71	1.56	1.09	0.88	0.83

ついて報告した.

本手法の収束条件などに関しては, 未解明な部分も残っている. 今後大規模固有値解法に対する有力な解法の一つとして, 実問題を用いて特性を明らかにしていく必要がある.

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」によるものである.

参考文献

1) 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, Vol. 2004, No. 20, pp. 25–30 (2004).

2) 西田晃: 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, 情処研報, Vol. 2004, No. 80, pp. 85–90 (2004).

3) Sleijpen, G. L. G. and van der Vorst, H. A.: A Jacobi-Davidson iteration method for linear eigenvalue problems, *SIAM J. Matrix Anal. Appl.*, Vol. 17, No. 2, pp. 401–425 (1996).

4) 西田晃, 小柳義夫: 大規模固有値問題のための Jacobi-Davidson 法とその特性について, 情報処理学会論文誌, Vol. 41, No. SIG8, pp. 101–106 (2000).

5) Fletcher, R. and Reeves, C. M.: Function minimization by conjugate gradients, *Comp. J.*, Vol. 7, pp. 149–154 (1964).

6) Bradbury, W. W. and Fletcher, R.: New Iterative Method for Solution of the Eigenproblem, *Numer. Math.*, Vol. 9, pp. 259–267 (1966).

7) Hestenes, M. R. and Karush, W.: A method of gradients for the caluculation of the characteristic roots and vectors of a real symmetric matrix, *J. Res. Natl. Inst. Stand. Technol.*, Vol. 47, pp. 45–61 (1951).

8) Knyazev, A. V.: Preconditioned eigensolvers—an oxymoron?, *Electron. Trans. Numer. Anal.*, Vol. 7, pp. 104–123 (electronic) (1998). Large scale eigenvalue problems (Argonne, IL, 1997).

9) Knyazev, A. V.: Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method, *SIAM J. Sci. Comput.*, Vol. 23, No. 2, pp. 517–541 (2001).

10) Arbenz, P. and Lehoucq, R.: A comparison of algorithms for modal analysis in the absense of a sparse direct method, Technical Report SAND2003-1028J, Sandia National Laboratories (2003).

11) 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, Vol. 30, No. 5, pp. 661–667 (1989).

12) Nishida, A.: AMG Preconditioned Conjugate Gradient Type Methods for Nonsymmetric Eigenproblems, *Proceedings of 12th Copper Mountain Conference on Iterative Methods*, Copper Mountain, USA (2005).

13) Fujii, A., Nishida, A. and Oyanagi, Y.: Evaluation of Parallel Aggregate Creation Orders : Smoothed Aggregation Algebraic Multigrid Method, *High Performance Computational Science A*, pp. 99–122 (2005).

14) Nishida, A., Kotakemori, H., Fujii, A. and Nukada, A.: Development of Scalable Software Infrastructure on Blue Gene Systems (Report on the Juelich Blue Gene/L Scaling Workshop 2006), Technical Report FZJ-ZAM-IB-2007-02, John von Neumann Institute for Computing, Research Centre Juelich (2007).