

COCG 法の積型解法について

東京大学大学院工学系研究科 曾我部知広 (Tomohiro Sogabe)

School of Engineering, The University of Tokyo

九州大学情報基盤センター 藤野清次 (Seiji Fujino)

Computing and Communications Center, Kyushu University

東京大学大学院工学系研究科 張紹良 (Shao-Liang Zhang)

School of Engineering, The University of Tokyo

1 はじめに

A を $N \times N$ 複素対称行列 ($A = A^T \neq A^H$), b を N 次元ベクトルとした大規模な複素対称線形方程式

$$Ax = b \quad (1)$$

を解く数値解法として Krylov 部分空間法がある. Krylov 部分空間法の代表として共役勾配法 (CG 法 [4]) が挙げられるが, これは正値エルミート行列用の解法であり上記のような問題には適していない. 一方, 非エルミート行列用の解法は, ランチョス・プロセスに基づく Bi-CG 法 [2] や, その改良型である積型解法族 (CGS 法 [7], Bi-CGSTAB 法 [9], GPBi-CG 法 [10]), そしてアーノルディ・プロセスに基づく GMRES 法 [6] など数多く提案されている. これらの解法は, 式 (1) に適用可能であるが, それらの多くは 1 反復当たりの行列-ベクトル積を 2 回以上必要とし, また係数行列の複素対称性を利用していない. 係数行列の特性を活かした解法として, 1 反復当たりの行列-ベクトル積は 1 回であり, 演算量は CG 法と同程度ある COCG 法 (Conjugate Orthogonal Conjugate Gradient method) が有効であることが知られている [8]. そこで本研究では, Bi-CG 法の一般化積型解法 [3] を参考にすることにより, COCG 法に対する一般化積型解法を導入し, COCG 法の収束速度の加速及びその収束過程において振動する残差ノルムの安定化を試みる.

2 COCG 法

係数行列が複素対称である線形方程式 (1) に対する COCG 法は, 次に述べる条件のもとで導くことができる. まず, n 反復目の近似解 x_n に対応する COCG 法の残差ベクトル r_n は, n 次のランチョス多項式 R_n [5] と初期残差ベクトル $r_0^c = b - Ax_0^c$ によって次のように定義される.

$$r_n^c = R_n(A)r_0^c. \quad (2)$$

ここで、ランチョス多項式 $R_n(\lambda)$ は補助多項式 $P_n(\lambda)$ を用いて次の交代漸化式を満たす。

$$R_0(\lambda) = 1, \quad P_0(\lambda) = 1, \quad (3)$$

$$R_n(\lambda) = R_{n-1}(\lambda) - \alpha_{n-1}\lambda P_{n-1}(\lambda), \quad (4)$$

$$P_n(\lambda) = R_n(\lambda) + \beta_{n-1}P_{n-1}(\lambda), \quad n = 1, 2, \dots. \quad (5)$$

補助ベクトル $\mathbf{p}_n^c = P_n(A)\mathbf{r}_0^c$ を導入すると、式 (2) と交代漸化式 (3)~(5) より COCG 法の残差ベクトル $\mathbf{r}_n^c$ が、次の 2 つの漸化式によって計算される。

$$\mathbf{r}_n^c = \mathbf{r}_{n-1}^c - \alpha_{n-1}A\mathbf{p}_{n-1}^c, \quad (6)$$

$$\mathbf{p}_n^c = \mathbf{r}_n^c + \beta_{n-1}\mathbf{p}_{n-1}^c. \quad (7)$$

残差ベクトルに対し、次の双直交条件

$$\mathbf{r}_n^c \perp K_n(\bar{A}; \bar{\mathbf{r}}_0^c) \quad (8)$$

を課すことにより α_n と β_n が決定され COCG 法が得られることを示す。ここで $K_n(\bar{A}; \bar{\mathbf{r}}_0^c) = \text{Span}\{\bar{\mathbf{r}}_0^c, \bar{A}\bar{\mathbf{r}}_0^c, \dots, (\bar{A})^{n-1}\bar{\mathbf{r}}_0^c\}$ である。まず、双直交条件 (8) より、

$$(\bar{A}^{n-1}\bar{\mathbf{r}}_0^c)^H \mathbf{r}_n^c = (A^{n-1}\mathbf{r}_0^c)^T \mathbf{r}_n^c = 0. \quad (9)$$

従って、式 (6) を式 (9) に代入すると α_{n-1} の計算式が得られる。

$$\alpha_{n-1} = \frac{(A^{n-1}\mathbf{r}_0^c)^T \mathbf{r}_{n-1}^c}{(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_{n-1}^c}. \quad (10)$$

また、双直交条件 (8) と式 (6) より次の性質

$$A\mathbf{p}_{n-1}^c \perp K_{n-1}(\bar{A}; \bar{\mathbf{r}}_0^c) \quad (11)$$

が成り立つので、 $(\bar{A}^{n-1}\bar{\mathbf{r}}_0^c, A\mathbf{p}_n^c) = (A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_n^c = 0$ となる。故に式 (7) より

$$(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_n^c = (A^{n-1}\mathbf{r}_0^c)^T A\mathbf{r}_n^c + \beta_{n-1}(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_{n-1}^c = 0$$

が成り立つ。従って、 β_{n-1} の計算式は次のようになる。

$$\beta_{n-1} = -\frac{(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{r}_n^c}{(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_{n-1}^c} = -\frac{(A^n\mathbf{r}_0^c)^T \mathbf{r}_n^c}{(A^{n-1}\mathbf{r}_0^c)^T A\mathbf{p}_{n-1}^c}. \quad (12)$$

また、式 (10) を用いると式 (12) は以下のように表すことができる。

$$\beta_{n-1} = -\alpha_{n-1} \frac{(A^n\mathbf{r}_0^c)^T \mathbf{r}_n^c}{(A^{n-1}\mathbf{r}_0^c)^T \mathbf{r}_{n-1}^c}. \quad (13)$$

COCG 法のアルゴリズムでは, α_n と β_n の計算式 (10), (13) を直接用いずに $\bar{\mathbf{r}}_n^c$ と $\bar{\mathbf{p}}_n^c$ を利用する. すなわち $\bar{\mathbf{r}}_n^c$ と $\bar{\mathbf{p}}_n^c$ を次のように展開する.

$$\begin{aligned}\bar{\mathbf{r}}_{n-1}^c &= \overline{R_n(A)\mathbf{r}_0^c} = \left((-1)^{n-1} \prod_{i=0}^{n-2} \bar{\alpha}_i \right) \bar{A}^{n-1} \bar{\mathbf{r}}_0^c + \bar{\mathbf{z}}_1, \\ \bar{\mathbf{p}}_{n-1}^c &= \overline{R_n(A)\mathbf{r}_0^c} = \left((-1)^{n-1} \prod_{i=0}^{n-2} \bar{\alpha}_i \right) \bar{A}^{n-1} \bar{\mathbf{r}}_0^c + \bar{\mathbf{z}}_2.\end{aligned}$$

ただし $\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2 \in K_n(\bar{A}; \bar{\mathbf{r}}_0^c)$. 従って, 双直交条件 (8) と性質 (11) より α_{n-1} に関する計算式 (10) と β_{n-1} に関する計算式 (11) は次の式と数学的に同値となる.

$$\alpha_n = \frac{(\mathbf{r}_n^c)^T \mathbf{r}_n^c}{(\mathbf{p}_n^c)^T A \mathbf{p}_n^c}, \quad \beta_n = \frac{(\mathbf{r}_{n+1}^c)^T \mathbf{r}_{n+1}^c}{(\mathbf{r}_n^c)^T \mathbf{r}_n^c}. \quad (14)$$

式 (6)~(7) と式 (14) より複素対称行列用の数値解法である COCG 法が得られる.

アルゴリズム 1: COCG 法

```

 $\mathbf{x}_0^c$  is an initial guess,  $\mathbf{r}_0^c = \mathbf{b} - A\mathbf{x}_0^c$ ,  $\beta_{-1} = 0$ ,
for  $n = 0, 1, 2 \dots$  until  $\|\mathbf{r}_n^c\| \leq \epsilon \|\mathbf{r}_0^c\|$  do:
begin
   $\mathbf{p}_n^c = \mathbf{r}_n^c + \beta_{n-1} \mathbf{p}_{n-1}^c$ ,
   $\alpha_n = \frac{(\mathbf{r}_n^c)^T \mathbf{r}_n^c}{(\mathbf{p}_n^c)^T A \mathbf{p}_n^c}$ ,
   $\mathbf{x}_{n+1}^c = \mathbf{x}_n^c + \alpha_n \mathbf{p}_n^c$ ,
   $\mathbf{r}_{n+1}^c = \mathbf{r}_n^c - \alpha_n A \mathbf{p}_n^c$ ,
   $\beta_n = \frac{(\mathbf{r}_{n+1}^c)^T \mathbf{r}_{n+1}^c}{(\mathbf{r}_n^c)^T \mathbf{r}_n^c}$ ,
end

```

COCG 法は, 生成する残差列 $\{\mathbf{r}_0^c, \dots, \mathbf{r}_n^c\}$ に対し次の直交性を満たす [2].

$$\bar{\mathbf{r}}_i^c \perp \mathbf{r}_j^c \quad (i \neq j).$$

これにより, 破綻がなければ COCG 法は高々 N 回の反復で収束することがわかる.

3 COCG 法の積型解法

Bi-CG 法を改良した積型解法族 (CGS 法, Bi-CGSTAB 法, GPBi-CG 法) に慣らい, 本研究では COCG 法に対する積型解法族を提案する.

3.1 定義

COCG法の積型解法を設計するにあたり, 新しい残差ベクトル $\mathbf{r}_n$ を COCG法の残差ベクトル $\mathbf{r}_n^c$ と最高次項の係数がゼロでない n 次多項式 H_n との積で定義する.

$$\mathbf{r}_n = H_n(A)\mathbf{r}_n^c = \mathbf{b} - A\mathbf{x}_n. \quad (15)$$

ここで, $H_n(\lambda)$ は次の交代漸化式を満たすように設計する.

$$H_0(\lambda) = 1, \quad G_0(\lambda) = \zeta_0, \quad (16)$$

$$H_n(\lambda) = H_{n-1}(\lambda) - \lambda G_{n-1}(\lambda), \quad (17)$$

$$G_n(\lambda) = \zeta_n H_n(\lambda) + \eta_n G_{n-1}(\lambda), \quad n = 1, 2, \dots. \quad (18)$$

ここで, 特に $\zeta_n = \alpha_n$, $\eta_n = \frac{\beta_{n-1}}{\alpha_{n-1}}\alpha_n$ とおくと H_n がランチョス多項式 R_n に帰着する.

3.2 残差多項式の計算

交代漸化式 (3)~(5) および (16)~(18) から, 多項式族 $H_n R_n$, $H_n R_{n+1}$, $\lambda G_{n-1} R_{n+1}$, $H_n P_n$, $\lambda H_n P_{n+1}$, $\lambda G_n P_n$, $G_n R_{n+1}$ に関する漸化式が次のように作れる.

$$H_{n+1} R_{n+1} = H_n R_{n+1} - \eta_n \lambda G_{n-1} R_{n+1} - \zeta_n \lambda H_n R_{n+1} \quad (19)$$

$$= H_n R_n - \alpha_n \lambda H_n P_n - \lambda G_n R_{n+1}, \quad (20)$$

$$H_n R_{n+1} = H_n R_n - \alpha_n \lambda H_n P_n, \quad (21)$$

$$\begin{aligned} \lambda G_n R_{n+2} &= H_n R_{n+1} - H_{n+1} R_{n+1} \\ &\quad - \alpha_{n+1} \lambda H_n P_{n+1} + \alpha_{n+1} \lambda H_{n+1} P_{n+1}, \end{aligned} \quad (22)$$

$$H_{n+1} P_{n+1} = H_{n+1} R_{n+1} + \beta_n H_n P_n - \beta_n \lambda G_n P_n, \quad (23)$$

$$\lambda H_n P_{n+1} = \lambda H_n R_{n+1} + \beta_n \lambda H_n P_n, \quad (24)$$

$$\begin{aligned} \lambda G_n P_n &= \zeta_n \lambda H_n P_n \\ &\quad + \eta_n (H_{n-1} R_n - H_n R_n + \beta_{n-1} \lambda G_{n-1} P_{n-1}), \end{aligned} \quad (25)$$

$$G_n R_{n+1} = \zeta_n H_n R_n + \eta_n G_{n-1} R_n - \alpha_n \lambda G_n P_n. \quad (26)$$

次に, 6つの補助ベクトル

$$\mathbf{t}_n = H_n(A)\mathbf{r}_{n+1}^c, \quad \mathbf{y}_n = A G_{n-1}(A)\mathbf{r}_{n+1}^c,$$

$$\mathbf{p}_n = H_n(A)\mathbf{p}_n^c, \quad \mathbf{w}_n = A H_n(A)\mathbf{p}_{n+1}^c,$$

$$\mathbf{u}_n = A G_n(A)\mathbf{p}_n^c, \quad \mathbf{z}_n = G_n(A)\mathbf{r}_{n+1}^c$$

を導入すると, 式 (15) 及び式 (19)~(26) から COCG法の積型解法の残差 $\mathbf{r}_{n+1}$ は次の漸化式で計算することができる.

$$\mathbf{r}_{n+1} = \mathbf{t}_n - \eta_n \mathbf{y}_n - \zeta_n A \mathbf{t}_n \quad (27)$$

$$= \mathbf{r}_n - \alpha_n A \mathbf{p}_n - A \mathbf{z}_n, \quad (28)$$

$$\mathbf{t}_n = \mathbf{r}_n - \alpha_n A \mathbf{p}_n, \quad (29)$$

$$\mathbf{y}_{n+1} = \mathbf{t}_n - \mathbf{r}_{n+1} - \alpha_{n+1} \mathbf{w}_n + \alpha_{n+1} A \mathbf{p}_{n+1}, \quad (30)$$

$$\mathbf{p}_{n+1} = \mathbf{r}_{n+1} + \beta_n (\mathbf{p}_n - \mathbf{u}_n), \quad (31)$$

$$\mathbf{w}_n = A \mathbf{t}_n + \beta_n A \mathbf{p}_n, \quad (32)$$

$$\mathbf{u}_n = \zeta_n A \mathbf{p}_n + \eta_n (\mathbf{t}_{n-1} - \mathbf{r}_n + \beta_{n-1} \mathbf{u}_{n-1}), \quad (33)$$

$$\mathbf{z}_n = \zeta_n \mathbf{r}_n + \eta_n \mathbf{z}_{n-1} - \alpha_n \mathbf{u}_n. \quad (34)$$

近似解 $\mathbf{x}_{n+1}$ の計算は, $\mathbf{r}_{n+1} = \mathbf{b} - A \mathbf{x}_{n+1}$ の関係と式 (28) より

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n + \mathbf{z}_n \quad (35)$$

となる.

3.3 α_n と β_n の計算式

COCG 法の積型解法における α_n と β_n の計算は, COCG 法で生成される α_n と β_n と等しくなるように設計し, その収束性を裏付ける. 多項式 H_n の最高次項の係数は $(-1)^n \prod_{i=0}^{n-1} \zeta_i$ であるので双直交条件 (8) より次の式が成立する.

$$(\mathbf{r}_0^c)^T \mathbf{r}_n = \left((-1)^n \prod_{i=0}^{n-1} \zeta_i \right) ((\bar{A})^n \bar{\mathbf{r}}_0^c, \mathbf{r}_n^c),$$

$$(\mathbf{r}_0^c)^T A \mathbf{p}_n = \left((-1)^n \prod_{i=0}^{n-1} \zeta_i \right) ((\bar{A})^n \bar{\mathbf{r}}_0^c, A \mathbf{p}_n^c).$$

従って, $\mathbf{r}_0 = \mathbf{r}_0^c$ を考慮に入れると式 (10) 及び (13) より次の計算式が得られる.

$$\alpha_n = \frac{\mathbf{r}_0^T \mathbf{r}_n}{\mathbf{r}_0^T A \mathbf{p}_n}, \quad \beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{\mathbf{r}_0^T \mathbf{r}_{n+1}}{\mathbf{r}_0^T \mathbf{r}_n}. \quad (36)$$

3.4 COCG 法の積型解法

漸化式 (27)~(34) および近似解の計算式 (35), そして α_n と β_n の計算式 (36) より以下の COCG 法の積型反復解法のアルゴリズムが得られる.

アルゴリズム 2: COCG 法の積型解法

$\mathbf{x}_0$ is an initial guess, $\mathbf{r}_0 = \mathbf{b} - A \mathbf{x}_0$,

$\mathbf{r}_0^* = \mathbf{r}_0$, $\mathbf{t}_{-1} = \mathbf{w}_{-1} = 0$, $\beta_{-1} = 0$,

for $n = 0, 1, 2 \dots$ until $\|\mathbf{r}_n\| \leq \epsilon \|\mathbf{r}_0\|$ do:

begin

$$\mathbf{p}_n = \mathbf{r}_n + \beta_{n-1}(\mathbf{p}_{n-1} - \mathbf{u}_{n-1}),$$

$$\alpha_n = \frac{\mathbf{r}_0^T \mathbf{r}_n}{\mathbf{r}_0^T A \mathbf{p}_n},$$

$$\mathbf{y}_n = \mathbf{t}_{n-1} - \mathbf{r}_n - \alpha_n \mathbf{w}_{n-1} + \alpha_n A \mathbf{p}_n,$$

$$\mathbf{t}_n = \mathbf{r}_n - \alpha_n A \mathbf{p}_n,$$

compute ζ_n, η_n ,

$$\mathbf{u}_n = \zeta_n A \mathbf{p}_n + \eta_n(\mathbf{t}_{n-1} - \mathbf{r}_n + \beta_{n-1} \mathbf{u}_{n-1}),$$

$$\mathbf{z}_n = \zeta_n \mathbf{r}_n + \eta_n \mathbf{z}_{n-1} - \alpha_n \mathbf{u}_n,$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{p}_n + \mathbf{z}_n,$$

$$\mathbf{r}_{n+1} = \mathbf{t}_n - \eta_n \mathbf{y}_n - \zeta_n A \mathbf{t}_n,$$

$$\beta_n = \frac{\alpha_n}{\zeta_n} \cdot \frac{\mathbf{r}_0^T \mathbf{r}_{n+1}}{\mathbf{r}_0^T \mathbf{r}_n},$$

$$\mathbf{w}_n = A \mathbf{t}_n + \beta_n A \mathbf{p}_n,$$

end

3.5 ζ_n, η_n の選び方

ζ_n, η_n の選び方により様々な積型解法が得られる。 $H_n = R_n$ となるように、パラメータ ζ_n, η_n を決めることにより得られる解法を COCGS 法, $\zeta_n = \arg \min_{\omega} \|\mathbf{r}_{n+1}\|$, $\eta_n = 0$ により得られる解法を COCGSTAB 法, $\zeta_n, \eta_n = \arg \min_{\zeta, \eta} \|\mathbf{r}_{n+1}\|$ により得られる解法を GPCOCG 法と名付ける。これらのパラメータの選び方は、非エルミート行列用の解法である Bi-CG 法の積型解法族と同様であり、特に係数行列の複素対称性を考慮に入れていない。Table 1 に ζ_n, η_n と積型解法との関係を示す。

Table 1. The choice for the product-type methods based on the COCG method.

COCGS	$\zeta_n = \alpha_n, \quad \eta_n = \frac{\beta_{n-1}}{\alpha_{n-1}} \alpha_n,$
COCGSTAB	$\zeta_n = \frac{(A \mathbf{t}_n)^H \mathbf{t}_n}{(A \mathbf{t}_n)^H A \mathbf{t}_n}, \quad \eta_n = 0,$
GPCOCG	$\zeta_n = \frac{\mathbf{y}_n^H \mathbf{y}_n \cdot (A \mathbf{t}_n)^H \mathbf{t}_n - \mathbf{y}_n^H \mathbf{t}_n \cdot (A \mathbf{t}_n)^H \mathbf{y}_n}{(A \mathbf{t}_n)^H A \mathbf{t}_n \cdot \mathbf{y}_n^H \mathbf{y}_n - \mathbf{y}_n^H A \mathbf{t}_n \cdot (A \mathbf{t}_n)^H \mathbf{y}_n},$ $\eta_n = \frac{(A \mathbf{t}_n)^H A \mathbf{t}_n \cdot \mathbf{y}_n^H \mathbf{t}_n - \mathbf{y}_n^H A \mathbf{t}_n \cdot (A \mathbf{t}_n)^H \mathbf{t}_n}{(A \mathbf{t}_n)^H A \mathbf{t}_n \cdot \mathbf{y}_n^H \mathbf{y}_n - \mathbf{y}_n^H A \mathbf{t}_n \cdot (A \mathbf{t}_n)^H \mathbf{y}_n},$ If $n = 0$, then $\zeta_n = \frac{(A \mathbf{t}_n)^H \mathbf{t}_n}{(A \mathbf{t}_n)^H A \mathbf{t}_n}, \quad \eta_n = 0.$

COCG法の積型解法族(COCCGS法, COCCGSTAB法, GPCOCG法)は, 1反復当たりの行列-ベクトル積は2回となり COCG法に比べ1回多くなる。また, 内積の演算量など他の演算量も全体的に多くなる。COCG法とその積型解法族の1反復当たりの演算量の詳細を Table 2 に示す。

Table 2. Summary of operations for iteration.

Method	Inner Product	AXPY	Matrix-Vector Product	Precond Solve
COCG	2	3	1	1
COCCGS	2	6	2	2
COCCGSTAB	4	6	2	2
GPCOCG	7	12	2	2

ここで AXPY は, ベクトルのスカラー倍とベクトルとの和 ($\alpha x + y$) の回数を意味する。

4 数値実験

本節では, 数値実験として NEP Collection [1] の問題で, 複素対称行列かつ正則である DWG961B と QC2534 を取り上げる。数値解法として, COCG法とその積型解法族(COCCGS法, COCCGSTAB法, GPCOCG法)及び Bi-CG法とその積型解法族(CGS法, Bi-CGSTAB法, GPBi-CG法)を採用した。数値実験は, ALPHA ワークステーション 750MHz 上で, 倍精度演算で行った。初期近似解は $x_0 = [0, 0, \dots, 0]^T$ とし, 右辺項 b は乱数, 収束判定は $\|r_n\|/\|b\| \leq 10^{-12}$ として打ち切り回数は 8000 回とした。†は, 打ち切り回数で収束しなかったことを意味する。数値実験は, 前処理として対角スケーリング及び ILU(0) を用いた。DWG961B に関して, ILU(0) を用いた実験では全ての解法が収束しなかったことと, QC2534 に関して, 対角スケーリングを用いた実験においても同様に収束しなかったため実験結果は割愛する。

両問題に対する各解法の収束履歴を Fig. 1～Fig. 4 に示す。図中の縦軸は, 相対残差の常用対数を表し, 横軸は反復回数を表す。Fig. 1 では, COCG法と COCCGS法のみが収束し, COCCGSTAB法及び GPCOCG法は, 打ち切り回数まで収束せずに停滞した。COCCGS法は COCG法と比べ反復回数の上で COCG法より速く収束した。しかし, COCCGS法の残差ノルムの振る舞いは, COCG法よりさらに激しく振動しながら収束した。対応する Bi-CG法系統の解法の収束履歴を Fig. 2 に示す。Fig. 2 より Bi-CG法系統の解法は, COCG法系統の解法とほぼ同様の収束性を示していることがわかる。

Fig. 3 では, COCG法の積型解法族は COCG法より速く収束しているのがわかる。特に COCCGSTAB法と GPCOCG法は, COCG法に比べ約 4 割程少ない反復回数で収束し, それらの収束過程における残差の振動も, COCG法に比べ緩やかになった。対応する Bi-CG法系統の収束履歴を Fig. 4 に示す。この数値例では, COCG法系統の解法が Bi-CG法系

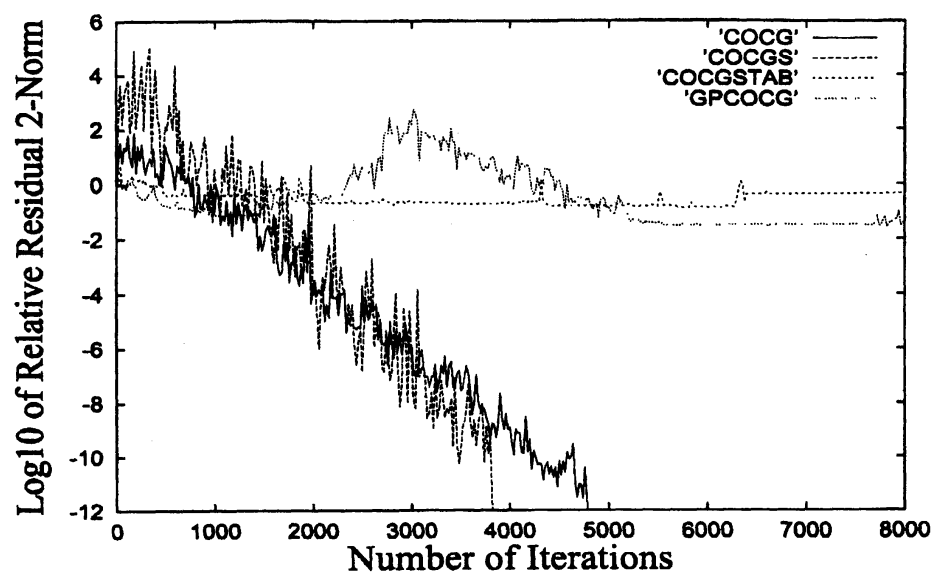


Fig. 1. DWG961B: Convergence history of COCG and the product-type methods.

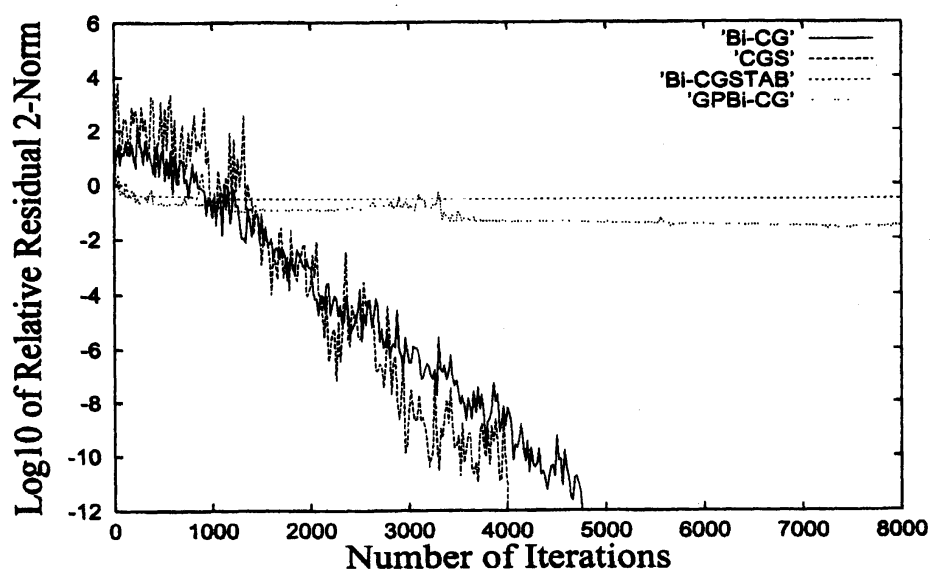


Fig. 2. DWG961B: Convergence history of Bi-CG and the product-type methods.

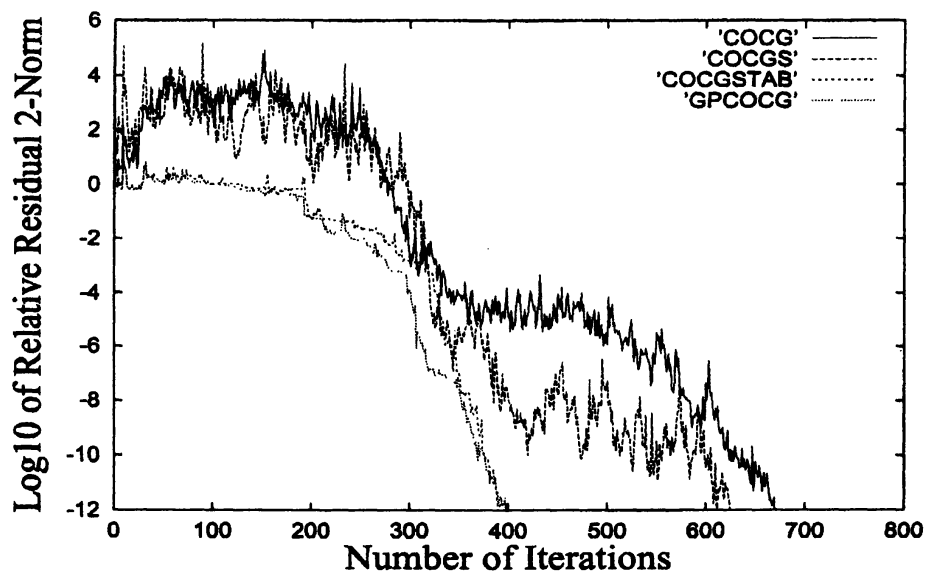


Fig. 3. QC2534: Convergence history of COCG and the product-type methods.

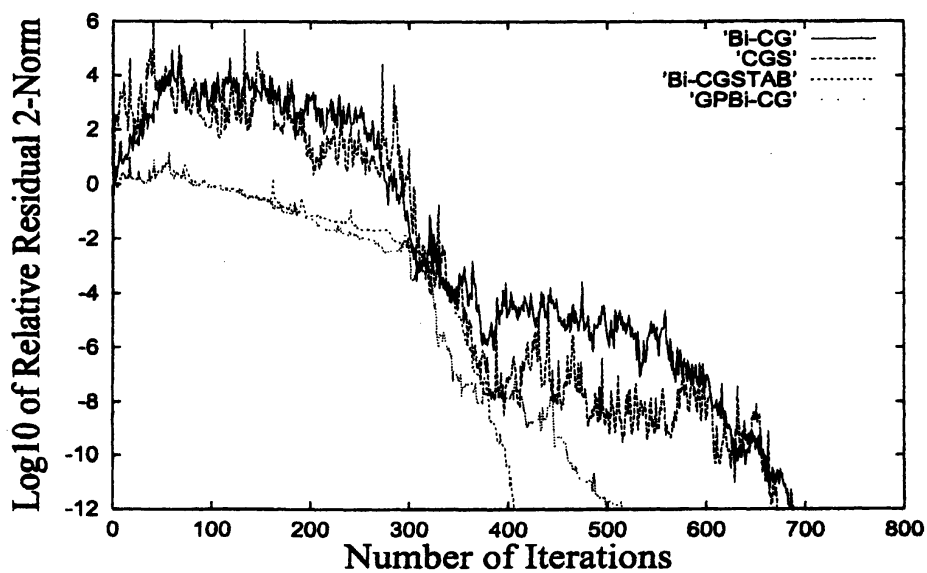


Fig. 4. QC2534: Convergence history of Bi-CG and the product-type methods.

統の解法に比べ全体的に良い収束性を示した。

Table 3. Convergence results for the iterative methods. The diagonal preconditioner and the ILU(0) preconditioner are applied to DWG961B and QC2534 respectively.

Method	DWG961B			QC2534		
	Its	Time	TRR	Its	Time	TRR
COCG	4788	1.29	-11.09	670	12.14	-9.99
COCGS	3827	1.97	-9.34	625	22.72	-9.65
COCGSTAB	†	†	†	399	14.85	-11.58
GPCOCG	†	†	†	398	15.54	-11.42
Bi-CG	4759	2.41	-11.06	689	25.12	-9.94
CGS	4003	2.08	-9.96	672	24.61	-8.77
Bi-CGSTAB	†	†	†	407	15.34	-11.71
GPBi-CG	†	†	†	515	20.09	-11.21

反復回数 (Its), 計算時間 (Time [秒]), そして真の相対残差 (TRR) の詳細を Table 3 に示す。Table 3 において, DWG961B で用いた前処理は対角スケーリングであり, QC2534 で用いた前処理は ILU(0) である。計算時間の観点からすると COCG 法が最も速く収束しているのが Table 3 からわかる。これは, 1 反復当たりの演算量が他の解法に比べ少ないことに起因する。しかし QC2534 の問題に関して, COCG 法は桁落ちのため要求された精度で打ち切られた時点での真の相対残差の精度が COCGSTAB 法に比べ 2 桁ほど低くなっていた。これは, COCG 法の収束過程においてその残差ノルムが激しく振動することが原因と考えられる。本実験で, COCG 法と COCGS 法が同様の収束の振る舞いを示し, COCGSTAB 法と GPCOCG 法が同様の収束の振る舞いを示すことがわかった。一方, Bi-CG 法系統の解法と比較してみると, 計算時間, 反復回数ともに COCG 法系統の積型解法が有効であった。

5 まとめ

本研究では, 複素対称行列用の反復解法である COCG 法に対してその積型解法を提案しアルゴリズムを導いた。非対称行列用の反復解法である Bi-CG 法の積型解法族は, 偏微分方程式から得られた線形方程式に対して極めて有効であることが確認されているため, そのアナロジーとして積型の考え方を COCG 法に適用し収束の加速を試みた。その中で COCGSTAB 法は, 反復回数の上では, COCG 法に比べ最大約 4 割反復回数が減少し, かつ精度の良い解が得られた。複素対称線形方程式 (1) は, Bi-CG 法系統の解法も適用可能であるので, COCG 法系統の解法と比較実験を行い, 結果としては COCG 法系統の解法が有効であった。今後の課題は, COCG 法の一般化積型解法に現れる ζ_n と η_n を選択する際, 複素対称性を考慮し, さらに効率の良いアルゴリズムを模索することである。

参考文献

- [1] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, Technical Report CS-97-355, Department of Computer Science, University of Tennessee, Knoxville, TN, March, 1997.
- [2] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems*, Lecture Notes in Mathematics 506, 1976, pp. 73-89.
- [3] 藤野清次, 張紹良: 反復法の数理, 朝倉書店, 1996.
- [4] M. R. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Standards, 49(1952), pp. 409-436.
- [5] C. Lanczos, *Solution of Systems of Linear Equations by Minimized Iterations*, J. Res. Nat. Bur. Standards, 49(1952), pp. 33-53.
- [6] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., 7(1986), pp. 856-869.
- [7] P. Sonneveld, *CGS, A Fast Lanczos-type Solver for Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., 10(1989), pp. 36-52.
- [8] H. A. van der Vorst, J. B. M. Melissen, *A Petrov-Galerkin type Method for Solving $A\mathbf{x} = \mathbf{b}$, Where A is Symmetric Complex*, IEEE Transaction on Magnetics, VOL. 26, NO. 2, (1990), pp. 706-708.
- [9] H. A. van der Vorst, *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., 13(1992), pp. 631-644.
- [10] S.-L. Zhang, *GPBi-CG: Generalized Product-type Methods Based on Bi-CG for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Comput., 18(1997), pp. 537-551.