

CG法の最近の前処理のロバスト性と効率化について

— 閾値によるドロッピングと対角緩和処理 —

九州大学 情報基盤センター 藤野 清次 (Seiji Fujino)

Computing and Communications Center, Kyushu University

梗概: 近年, 共役勾配法 (Conjugate Gradient method, 以下 CG 法と略す) 系統の反復法の新しい前処理 (pre-conditioning) 法が次々と提案されている。代表的なものとして, 例えば, M. Benzi らによる SAINV (Stabilized AINV) や RIF 前処理などがあげられる。それらの前処理法は, 実際の問題に対して, 従来の不完全 (Incomplete) Cholesky (以下, IC と略す) 分解が持ち得なかった収束のロバスト性を持ち, かつ計算効率も非常に高い前処理法である。そこで, 本研究では, 様々な新しい前処理法の原版を実装して評価し, さらにその収束性を改良し収束性能向上させた高性能前処理つき共役勾配法を開発した。

1 はじめに

大規模な疎行列を係数行列 A に持つ連立一次方程式 $Ax = b$ は, 前処理つき反復法によって解かれることが多い。特に A が正定値対称行列のとき, 共役勾配法 (CG 法) がよく用いられる [6]。現在, 前処理行列の構成方法は, 大きく二つの考え方に分けることができる。一つは係数行列 A を近似する不完全分解, もう一つは逆行列 A^{-1} を近似する近似逆行列分解である。前者ではコレスキー分解がよく知られている [16]。後者では, A -直交化過程に基づいて逆行列 A^{-1} を近似分解する近似逆行列 (Approximate INVerse, 以下 AINV と略す) が最近知られてきた [2]。AINV では, 前処理行列のスパース性を保つために, dropping 処理と呼ばれる操作が行われる。dropping 処理とは, あらかじめある閾値を設定し, 分解過程で生まれた非零要素に対してその値が閾値よりも大きいときのみそれを残し, 小さいときは零とみなして以降の分解を行うという処理を指す。dropping 処理は, 計算量やメモリ量の削減という観点から AINV では不可欠な操作である。しかし, この操作によって, 前処理行列の正定値性が失われ, CG 法の収束性に悪影響を及ぼすことがある [16]。そこで, 近似分解の中の計算方法を工夫し, CG 法の収束の安定化を図ったものが安定化近似逆行列 (Stabilized-AINV, 以下 SAINV と略す) 前処理である [3]。SAINV では計算量を増加させることなく分解が安定に行なうことができる。

近年, SAINV の分解過程において得られた不完全逆行列因子 Z が行列 A の通常の不完全分解因子 L と数学的に等しい, という事実を利用した新しい前処理が提案された [4] [5]。この前処理でも安定化近似逆行列の優れた特徴を受け継いで安定した不完全分解ができる。そのため, RIF (Robust Incomplete Factorization) と呼ばれる。この前処理では, 余計な手間を掛けずに不完全分解因子 L が得られ, 元の近似逆行列を用いた前処理に比べてより一層有効な前処理になることが知られている。

すでに既報において, SAINV 前処理や RIF 前処理の性能向上のために, 近似分解中に dropping 処理を二重に行う double dropping 技法を提案し, その有効性を数値実験で実証した [7][8] [9][10][11][12]。本研究では, 疎行列データベースに収納された行列への適用だけでなく, 実際のコンクリート橋の梁 (BEAM) やケーブル定着部における応力解析で生じた行列に対しても double dropping つきの改良版の SAINV 前処理や同改良版の RIF 前処理を適用した結果とその有効性の検証を中心にその概要を述べる。

本論文の構成は以下のとおりである。まず, 2 節で前処理つき共役勾配法について記述する。3 節では, A -直交化に基づく近似逆行列分解の概略と同分解つき CG 法について述べる。4 節では, Double dropping つき安定化近似逆行列 (SAINV) について記述する。5 節では, 近似逆行列分解 (SAINV) からロバスト不完全分解 (RIF) が導かれる過程を示す。5.1 節では, 対称正定値行列用の前処理のまとめを簡単に行なう。そして, 6 節で数値実験結果を示す。最後に, 7 節でまとめを述べる。

2 前処理つき共役勾配法

正定値対称行列を係数行列 A に持つ連立一次方程式

$$(2.1) \quad Ax = b$$

を前処理つき CG 法で解くことを考える。ここで, A は $n \times n$ の正方行列, x , b は次元数 n の解および右辺ベクトルとする。不完全分解を用いた前処理では, 係数行列 A を

$$(2.2) \quad A \simeq LDL^t$$

のように分解する。ここで、 L は下三角行列、 D は対角行列、上付き添字 t は転置を表す。

3 A-直交化による近似逆行列分解つき CG 法と近似逆行列分解

一方、A-直交化による近似逆行列分解前処理では、逆行列 A^{-1} そのものを、

$$(3.1) \quad A^{-1} \simeq ZD^{-1}Z^t$$

のように近似する。ここで、 Z は上三角行列、 D は対角行列とする。この近似逆行列分解を用いた前処理つき CG 法の算法は以下のように表される。

$$\begin{aligned} r_0 &= b - Ax_0, \quad p_0 = ZD^{-1}Z^tr_0, \\ \text{for } m &= 1, 2, \dots \\ \alpha_m &= \frac{(r_{m-1}, ZD^{-1}Z^tr_{m-1})}{(p_{m-1}, Ap_{m-1})}, \\ x_m &= x_{m-1} + \alpha_m p_{m-1}, \\ r_m &= r_{m-1} - \alpha_m Ap_{m-1}, \\ \text{if } \|r_m\|_2 / \|r_0\|_2 &\leq \varepsilon \text{ stop} \\ \beta_m &= \frac{(r_m, ZD^{-1}Z^tr_m)}{(r_{m-1}, ZD^{-1}Z^tr_{m-1})}, \\ p_m &= ZD^{-1}Z^tr_m + \beta_m p_{m-1}, \\ \text{end for.} \end{aligned}$$

3.1 A-直交化による近似逆行列分解

A-直交化過程に基づく逆行列分解 (AINV) では、以下に示す分解によって Z および D を構成する [2].

$$\begin{aligned} \text{for } j &= 1, 2, \dots, n \\ z_j^{(0)} &= e_j \\ \text{end for} \\ \text{for } i &= 1, 2, \dots, n \\ \text{for } j &= i, i+1, \dots, n \\ d_j &= a_i^t z_j^{(i-1)} \\ \text{end for} \\ \text{for } j &= i+1, j+2, \dots, n \\ z_j^{(i)} &= \underline{z_j^{(i-1)} - \frac{d_j}{d_i} z_i^{(i-1)}} \\ \text{end for} \\ \text{end for} \end{aligned}$$

上記の近似逆行列分解において、 $z_j^{(i)}$ は反復 i 回目における行列 Z の第 j 番目の列ベクトル、 e_j は第 j 番目の要素が 1 である単位ベクトル、 d_j は対角行列 D の第 j 番目の対角要素、 a_i^t は行列 A の第 i 番目の行ベクトルをそれぞれ表す。また、下線をつけた部分は $z_j^{(i-1)}$ の更新を行う箇所であり、最終的に逆行列の近似分解の因子 Z が得られる。このとき、行列のスパース性を保つために、あらかじめ設定した閾値の値よりも小さい要素は捨てられる。この処理を **dropping**、閾値のことを **tolerance value** (以下、 tol と略す) と呼ぶ。上記の分解中で下線をつけた部分は、dropping 処理が行われると以下ようになる。ただし、dropping 処理は要素ごとの操作と

なるため、列ベクトル $z_j^{(i-1)}$ の第 k 番目の要素を $z_{kj}^{(i-1)}$ と表記することにする。

$$(3.2) \quad \begin{aligned} & \text{for } k = 1, \dots, i \\ & \quad \text{if } |z_{kj}^{(i-1)} - \frac{d_j}{d_i} z_{ki}^{(i-1)}| > \text{tol} \\ & \quad \quad z_{kj}^{(i)} = z_{kj}^{(i-1)} - \frac{d_j}{d_i} z_{ki}^{(i-1)} \\ & \quad \text{else} \\ & \quad \quad z_{kj}^{(i)} = 0 \\ & \quad \text{end if} \\ & \text{end for} \end{aligned}$$

AINV では、このようにして近似逆行列因子 Z および対角行列 D を構成し、前節で示した近似逆行列分解前処理つき CG 法に適用する。

4 Double Dropping つき SAINV

既報において SAINV の分解過程で二重の dropping を行い、CG 法の収束性を大幅に改善させる前処理を提案した [7][8]。この二重の dropping 処理 (以下、double dropping と呼ぶ) では、小さな値の要素を棄却する従来の dropping 処理の他に、ベクトル $z_j^{(i-1)}$ を更新するかどうかを判定するための dropping 処理を追加する。この新たに追加した dropping 処理のために閾値 tol_dd が導入される。そして、その更新の可否判定には $z_i^{(i-1)}$ に掛ける比率 $\frac{d_j}{d_i}$ の絶対値が使用される。したがって、 $z_j^{(i-1)}$ の更新を表す (3.2) 式は (4.1) 式に置き換わる。ここで、下線部分が double dropping にあたる部分である。

$$(4.1) \quad \begin{aligned} & \text{if } \frac{|d_j|}{d_i} > \text{tol_dd} \\ & \quad \text{for } k = 1, \dots, i \\ & \quad \quad \text{if } |z_{kj}^{(i-1)} - \frac{d_j}{d_i} z_{ki}^{(i-1)}| > \text{tol} \\ & \quad \quad \quad z_{kj}^{(i)} = z_{kj}^{(i-1)} - \frac{d_j}{d_i} z_{ki}^{(i-1)} \\ & \quad \quad \quad \text{else} \\ & \quad \quad \quad \quad z_{kj}^{(i)} = 0 \\ & \quad \quad \quad \text{end if} \\ & \quad \quad \text{end for} \\ & \quad \text{end if} \end{aligned}$$

この double dropping は、前述した SAINV における近似分解の安定性に影響を与えない。それゆえ、SAINV と同様に安定して分解を行うことができる。

5 近似逆行列からロバスト不完全分解へ

5.1 完全分解について

ここでは、係数行列 A とその逆行列 A^{-1} を完全分解することを考える。完全分解のとき、(2.2) 式と (3.1) 式は各々次のように表すことができる。

$$(5.1) \quad A = \bar{L}\bar{D}\bar{L}^t,$$

$$(5.2) \quad A^{-1} = \bar{Z}\bar{D}^{-1}\bar{Z}^t.$$

ここで、 \bar{L} 、 \bar{D} 、 \bar{Z} は完全分解因子を各々表す。特に、(5.1) 式は完全コレスキー分解に相当する。そしてその逆行列は、

$$(5.3) \quad A^{-1} = \bar{L}^{-t}\bar{D}^{-1}\bar{L}^{-1}$$

表 1: 対称正定値行列用の前処理 (原版と改良版)

疎行列	前処理	原版	改良 (Improved) 版
対称正定値	なし IC 分解 その他	IC 完全コレスキー	重み調節 IC 分解
	A-直交化 同 RIC 分解 同	SAINV RIF RIC 同	ISAINV IRIF pf-RIC 対角緩和 RIC

となる。ここで、(5.2) 式および (5.3) 式が同じ構成をしていることがわかる。したがって、一般に完全コレスキー分解には一意性があることから、完全分解因子 \bar{L} と逆行行列因子 \bar{Z} の間に次の関係が成り立つ。

$$(5.4) \quad \bar{Z}^t = \bar{L}^{-1}.$$

この等式から、(5.1) 式と (5.2) 式は両方とも次のように変形でき、これは $A\bar{Z}\bar{D}^{-1}$ が完全分解因子 \bar{L} と同じになることを意味する。

$$(5.5) \quad \bar{L} = A\bar{Z}\bar{D}^{-1} \quad (\text{または } \bar{L}D = A\bar{Z}).$$

したがって、不完全分解因子 L と近似逆行行列因子 Z の間にも、

$$(5.6) \quad L \simeq AZD^{-1} \quad (\text{または } LD \simeq AZ)$$

という関係が成り立つ。(5.6) 式は、近似逆行行列因子 Z から不完全分解因子 L が計算可能であることを意味しており、A-直交化に基づく不完全分解において重要な式である [4][5].

5.2 対称正定値行列用の前処理のまとめ

本論文において収束性能を調べた疎 (スパース) な対称正定値行列用の前処理 (原版と改良版) の一覧を表 1 に示す。ただし、今回開発した改良版の前処理を太字による表記で示す。

6 数値実験

6.1 計算機環境と計算条件

数値実験は九州大学情報基盤センターに設置された以下の計算機環境で行った。

- 計算機：富士通 Prime Power 850 (PE 数：32)
- CPU(クロック周波数)：SPARC64 (1.3GHz)
- 1PE 当たりのメモリ容量：1.5 Gbyte
- 最適化オプション：-Kfast

上記の計算機の 1 PE を使用した。計算はすべて倍精度演算で行い、最大反復回数は各行列の次元数とし、そこで計算を中断させた。収束判定値 ε は相対残差 L_2 ノルム $\|r_m\|_2 / \|r_0\|_2$ の値が 10^{-9} 以下になったときとした。初期近似解 x_0 はすべて 0 とした。また、行列は対角スケールングを用いて対角項を 1 に正規化した。

表 2: テスト行列の特徴.

項目	行列	
	BEAM	CABLE
次元数	10626	59002
総非零要素数	233268	1986094
1行当たり非零要素数	21.95	33.66
平均バンド幅	576	1741
総ノード数	1977	20194
総要素数	2832	16084

表 3: 行列 BEAM に対する従来の前処理つき CG 法の収束性 (最大反復回数=次元数)

行列	前処理	メモリ [Mbyte]	反復 回数	CPU 時間 [秒]		
				前処理	CG	合計
BEAM	なし		最大	-	-	-
	IC	3.5	7620	-	102.	102.
	重み調節 IC	3.5	7322	-	79.9	79.9
	完全 Cholesky	100	1	53.7	0.12	53.8

6.2 テスト行列

コンクリート橋の設計に実際に使用された応力解析の問題で得られた行列をテスト行列に採用した。解くべき方程式の右辺項は設定された加重条件を元に離散化による値を用いた。行列名を **BEAM**(梁, はり) と **CABLE** と各々呼び、その特徴を以下に示す。

行列 BEAM は橋梁の間にトラック荷重を課したときの応力解析で生じた問題で、シェル要素のみで離散化された。一般に、シェル要素による離散化行列の解析は難しくことが多く、収束までに多くの反復回数が必要になる [17]。一方、行列 CABLE はコンクリート橋におけるケーブル定着部の応力解析で生じた問題で、ソリッド要素だけで離散化が行なわれた。ソリッド要素による離散化の場合、シェル要素の場合と比較して、問題は比較的解き易く、反復法の収束も速いとされる。しかし、行列の次元数は一般に大きくなることが多い。図 1 に行列 BEAM のときの応力 (x - y 方向成分) 分布の様子を示す。

6.3 実験結果

さらに、図 2 と図 3 に、各種前処理つき CG 法の残差の履歴を示す。図 2 の横軸は反復回数、図 3 の横軸は CPU 時間、縦軸はいずれも相対残差 L_2 ノルム (常用対数目盛) を表す。図からわかるように、**IRIF** および対角緩和 **RIC** つき CG 法の収束性が他の前処理法に比べて非常に優れている。行列 BEAM に対して、表 3 に従来の前処理つき CG 法の収束性を示す。同様に、表 4 に A-直交化に基づく前処理つき CG 法の収束性 (閾値は最適値探索) およびをロバスト不完全分解型前処理つき CG 法の収束性 (閾値と対角緩和 ω の値は数回試行後の結果) を示す。これらの結果から、従来の前処理に比べて新しい前処理が非常に効率的であることがわかる。

なお、ロバスト不完全分解型の前処理 (RIC 分解, ポストフィルタリングつき RIC 分解, 対角緩和 RIC 分解) つき CG 法の理論および実験結果については、参考文献 [1] [13] [14] [15] を参照。

図 4 に行列 CABLE に対する応力 (x - y 方向成分) 分布を示す。さらに、行列 CABLE に対して、表 5 に従来の前処理つき CG 法の収束性を示す。同様に、表 6 に A-直交化に基づく前処理つき CG 法の収束性 (閾値は最適値探索) およびをロバスト不完全分解型前処理つき CG 法の収束性 (閾値と対角緩和 ω の値は数回試行後の結果) を示す。行列 BEAM と同様に、これらの結果からも、従来の前処理に比べて新しい前処理が効率的であることがわかる。

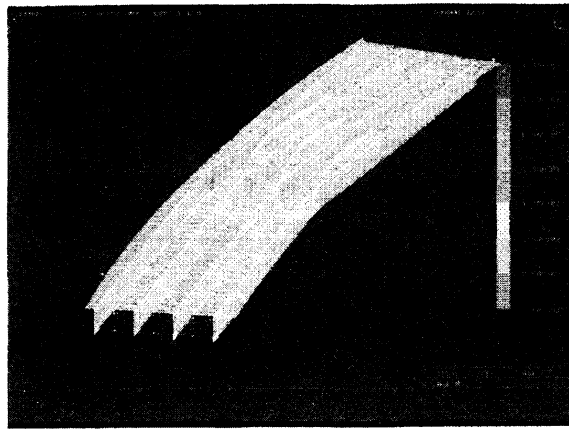


図 1: 橋梁の間の応力 (x - y 方向成分) 分布.

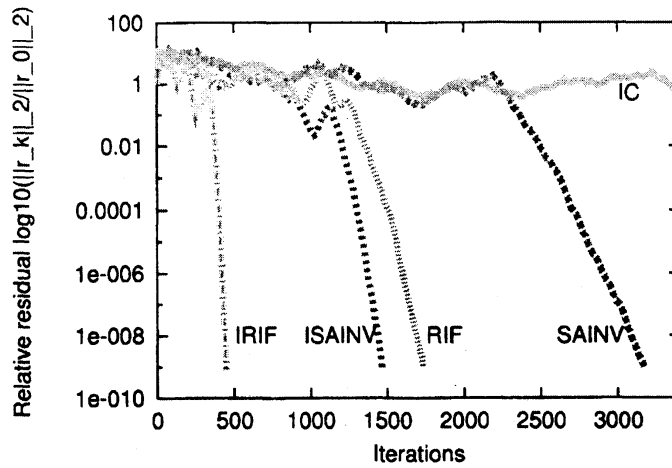


図 2: 行列 BEAM に対する IC 分解, RIF および IRIF 前処理つき CG 法の収束履歴.

表 4: 様々な前処理つき CG 法の収束性の比較 (行列 BEAM)

行列	前処理	閾値		メモリ (比) [Mbyte]	反復 回数	CPU 時間 [秒]		
		tol	tol_dd			前処理	CG	合計 (比)
BEAM	SAINV	0.11	-	11.8 (1.00)	3151	2.15	27.18	29.33 (1.0)
	ISAINV	0.08	0.32	6.72 (0.57)	1377	0.54	8.99	9.53 (0.32)
	RIF	0.11	-	11.5 (0.97)	1735	2.70	13.1	15.8 (0.54)
	IRIF	0.02	0.07	9.33 (0.79)	447	1.67	4.49	6.16 (.21)
	前処理	閾値	対角緩和	メモリ (比)	反復	CPU 時間 [秒]		
		tol	ω (倍率)	[Mbyte]	回数	前処理	CG	合計 (比)
	RIC	0.001	-	9.46 (1.00)	607	0.88	8.02	8.90 (1.00)
	pf-RIC	0.001	$\times 10$	5.86 (0.62)	670	0.87	5.04	5.91 (0.66)
対角緩和 RIC	0.001	$\times (1/100)$	10.4 (1.10)	210	1.19	3.17	4.36 (.49)	

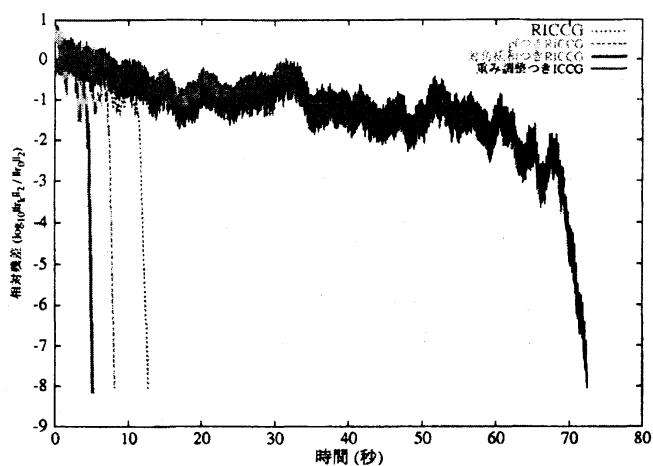


図 3: 行列 BEAM に対する重み調節 IC, RIC, pf-RIC および対角緩和 RIC 前処理つき CG 法の収束履歴.

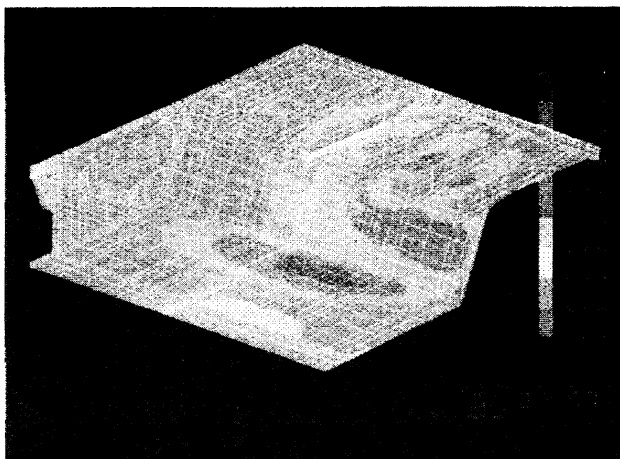


図 4: 応力 (x - y 方向成分) 分布 (行列 CABLE).

表 5: 行列 CABLE に対する従来の前処理つき CG 法の収束性

行列	前処理	メモリ [Mbyte]	反復 回数	CPU 時間 [秒]		
				前処理	CG	合計
CABLE	なし		7330	0.0	314.	314.
	IC	27.8	3024	2.24	376.	378.
	重み調節 IC	27.8	2736	0.30	310.	310.
	完全 Cholesky	1670	1	5182.	1.86	5184.

表 6: 様々な前処理つき CG 法の収束性の比較 (行列 CABLE)

行列	前処理	閾値		メモリ (比) [Mbyte]	反復 回数	CPU 時間 [秒]		
		tol	tol_dd			前処理	CG	合計 (比)
CABLE	SAINV	0.07	-	75.3 (1.00)	1469	20.1	94.2	114.3 (1.00)
	ISAINV	0.04	0.06	61.8 (0.82)	1183	16.5	88.8	105.3 (0.92)
	RIF	0.04	-	88.1 (1.17)	786	49.2	61.9	111.1 (0.97)
	IRIF	0.04	0.08	61.2 (0.81)	785	28.9	61.3	90.2 (.79)
	前処理	閾値 tol	対角緩和 ω (倍率)	メモリ (比) [Mbyte]	反復 回数	CPU 時間 [秒]		
	RIC	0.001	-	105 (1.00)	512	16.8	74.6	91.5 (1.00)
	pf-RIC	0.001	$\times 10$	55.4 (0.53)	573	16.7	47.2	63.9 (0.70)
	対角緩和 RIC	0.001	$\times (1/100)$	118. (1.12)	159	22.9	25.5	48.5 (.53)

7 まとめ

実際問題に対して、従来の前処理 (対角スケーリングや不完全コレスキー分解) などに比べて新しい前処理 (近似逆行列分解やロバスト不完全分解) 前処理が CG 法の前処理として非常に効率的でロバストであることがわかった。

謝辞

近似逆行列およびロバスト不完全分解前処理の各々について新しいアイデアを出し収束性を大幅に改良してくれた九州大学大学院システム情報科学府 池田優介君 柿原正伸君に心より敬意を表する。

参考文献

- [1] Ajiz, M. A., Jennings, A.: A robust incomplete Choleski-conjugate gradient algorithm, *International Journal for Numerical Methods in Engineering*, Vol. 20, pp. 949-966 (1984).
- [2] Benzi, M., Meyer, C. D., Tuma, M.: A sparse approximate inverse preconditioner for the conjugate gradient method, *SIAM J. on Scientific Computing*, Vol. 17, pp. 1135-1149 (1996).
- [3] Benzi, M., Cullum, J. K., Tuma, M.: Robust approximate inverse preconditioning for the conjugate gradient method, *SIAM J. on Scientific Computing*, Vol. 22, pp. 1318-1332 (2000).
- [4] Benzi, M.: Preconditioning techniques for large linear systems : A survey, *J. of Comput. Physics*, Vol. 189, pp. 418-477 (2002).
- [5] Benzi, M., Tuma, M.: A robust incomplete factorization preconditioner for positive definite matrices, *Numer. Lin. Alg. Appl.*, Vol. 10, pp. 385-400 (2003).
- [6] Hestenes, M. R., Stiefel, E.: Methods of conjugate gradients for solving linear systems, *J. of Research of the National Bureau of Standards*, Vol. 49, pp. 409-436 (1952).
- [7] 池田優介, 藤野清次: 二重ドロッピングによる安定化近似逆行列前処理の改良, 情報処理学会 コンピューティングシステム ACS 論文誌 第 4 号, 2004. (印刷中)
- [8] Ikeda, Y., Fujino, S.: An improvement of stabilized AINV preconditioning by double dropping, Proc. of the International Symposium on Information Science and Electrical Engineering 2003(ISEE 2003), pp.393-396, Fukuoka, Nov. 13-14, 2003.
- [9] Ikeda, Y., Fujino, S.: An effective use of Improved Stabilized Approximate INVerse preconditioning according to characteristics of computers, Trans. of Journal *INFORMATION*. (To appear)

- [10] 池田優介, 藤野清次, 柿原 正伸, 井上 明彦: A-直交過程に基づく RIF 前処理の効率化について, 情報処理学会 「ハイパフォーマンスコンピューティングと計算科学シンポジウム: HPCS2004」 講演集, pp.1-8, 2004.
- [11] 池田優介, 藤野清次, 柿原 正伸, 井上 明彦: 情報処理学会 コンピューティングシステム ACS 論文誌, 2004. (投稿中)
- [12] 藤野清次: CG 法の最近の前処理のロバスト性と効率化について-閾値によるドロップリングと対角緩和処理-, 京都大学数理解析研究所 研究集会「数値解析と新しい情報技術」講演予稿集, pp.7-9, 12月 2003.
- [13] Kakahara, M., Fujino, S.: An Improvement of Ajiz-Jennings type of Incomplete Factorization Preconditioning by means of Post Filtering, Trans. of Journal *INFORMATION*. (To appear)
- [14] 柿原 正伸, 池田 優介, 藤野 清次, Moethuthu: 対角緩和つき Ajiz-Jennings 型不完全分解前処理の提案, 情報処理学会 「ハイパフォーマンスコンピューティングと計算科学シンポジウム: HPCS2004」 講演集, pp.149-156, 2004.
- [15] Kaporin, I. E., High quality preconditioning of a general symmetric positive definite matrix based on its $U^T U + U^T R + R^T U$ -decomposition, *Numer. Lin. Alg. Appl.*, Vol. 5, pp. 483-509 (1998).
- [16] Meijerink, J.A., van der Vorst, H.A.: An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix, *Math. Comput.*, Vol. 31, pp. 148-162 (1977).
- [17] 有限要素法による構造解析システム: FEMLEEG ユーザガイド, ホクトシステム (2003).